der Bundeswehr
Universität München

Das Terrain Covering Problem
unter Unsicherheiten
-
Ein onlinebasierter Ansatz für autonome Agenten
(Titel der Dissertation)

Michael Preuß
(Vorname und Name)

Vollständiger Abdruck der von der Fakultät für Informatik der Universität der Bundeswehr München zur Erlangung des akademischen Grades eines

Doktors rer. nat.

genehmigten Dissertation.

Gutachter:
Professor Dr. Pickl
Professor Dr. Siegle

Die Dissertation wurde am 06.10.2016 bei der Universität der Bundeswehr München eingereicht und durch die Fakultät für Informatik am 13.10.2016 angenommen. Die Mündliche Prüfung fand am 01.03.2017 statt.

Universität der Bundeswehr München

Das Terrain Covering Problem unter Unsicherheiten

Ein onlinebasierter Ansatz für autonome Agenten

Dissertation
zur Erlangung des Grades eines
Doktors rer. nat.
der Universität der Bundeswehr München
an der Fakultät für Informatik

vorgelegt von
Michael Preuß

Dekan:	Professor Dr. Buchenrieder
Gutachter:	Professor Dr. Pickl
	Professor Dr. Siegle

Tag der mündlichen Prüfung:	01.03.2017
Dekan & Vorsitzender des Prüfungsausschusses:	Professor Dr. Buchenrieder
Gutachter:	Professor Dr. Pickl
	Professor Dr. Siegle
Prüfer:	Professor Dr. Borghoff
	Professor Dr. Lechner
	Professor Dr. Reinhardt

Danksagung

Ich danke

- Herrn Professor Dr. Stefan Pickl für seine kontinuierliche, allumfassende Unterstützung und den wertvollen Diskussionen bezüglich dieser wissenschaftlichen Arbeit im Rahmen des Doktorvaters,
- Herrn Professor Dr. Markus Siegle für seine wissenschaftlich relevanten sowie wichtigen Anmerkungen und seiner Tätigkeit als Zweitgutachter für mein Promotionsverfahren,
- Herrn Professor Dr. Alexander Bordetsky für den Erfahrungsaustausch und die Aufnahme in das CENETIX Team,
- Herrn Professor Dr. Matthias Dehmer und Herrn Professor Dr. Andreas Holzinger für interdisziplinäre Diskurse über meine Dissertation hinaus,
- Frau Dr. Silja Meyer-Nieberg für ihren fachlichen Diskurs und den Austausch neuer Optimierungsansätze,
- Herrn Dr. Martin Zsifkovits, Herrn Armin Leopold, Frau Manon Raap, Herrn Sorin Nistor und Herrn Maximilian Moll für die Unterstützung in einem kollegialen Umfeld,
- Herrn Tino Krug für seinen professionellen IT-Support einschließlich der Hardware-Bereitstellung für die Experimentierung
- Frau Annemarie Fischaleck und Frau Gabriela Karasz für ihre administrative Unterstützung und die herzliche Stimmung im Institut

und im besonderen Maße

- meiner gesamten Familie für ihre Unterstützung und das Verständnis für meine Hoch- und Tiefphasen.

Was gibt uns wohl den schönsten Frieden,
Als frei am eigenen Glück zu schmieden.
Johann Wolfgang von Goethe (1749–1832)

Jede Verknüpfung schafft Wissen...
und jeder Schritt in der Zukunft,
muss mit dieser endlosen Kette verbunden sein.
King Camp Gillette (1855–1932)

Nähe zu wagen ist ein schwieriger Balanceakt.
Wer zu weit geht oder nicht weit genug,
entfernt sich vom anderen.
Ernst Ferstl (*1955)

Für meine Großeltern
Christa Materne,
Renate Preuß und
Otto Preuß

Kurzfassung

In dieser Dissertation wird ein neues Terrain Covering Model vorgestellt, das autonomen Agenten eine vollständige Befahrung einer zuvor unbekannten Systemumgebung ermöglicht. Das zugrundeliegende Online Terrain Covering Problem kommt in verschiedenen Anwendungsbereichen wie beispielsweise der Landwirtschaft, dem Militär und Katastrophenschutz sowie der Reinigungs- und Weltraumindustrie zum Tragen. Hierbei können mit Hilfe von mobilen Robotern unter anderem eine Fernerkundung, Aufklärung, Such- und Rettungsaktion oder Überwachung unterstützt werden, die für Menschen zeitaufwendig, mühsam oder gefährlich sind. Eine unbekannte Umgebung stellt hohe Anforderungen an einen robusten und effizienten Einsatz. Es müssen kontinuierlich zielgerichtete Entscheidungen unter Untersicherheiten getroffen werden. Neben einer geeigneten Koordination im Sinne des Task Allocation Problem wird in dieser Dissertation ein neues Lösungsverfahren für das Shortest Path Problem unter Unsicherheiten beschrieben und analysiert. Abgesehen von den Kosten, die aus einer Befahrung einer Route resultieren müssen im Sinne einer robusten Lösung vor allem die Unsicherheiten betrachtet werden. Die Transformation in ein Mehrkriterielles Optimierungsproblem verbessert unter anderem die Vergleichbarkeit von Routen und bietet dem Entscheidungsträger eine Vielzahl von Pareto optimalen Routen dar.

Abstract

This thesis describes a new terrain covering model which enables autonomous agents to traverse unknown system environments. The referring online terrain covering problem can be applied to many domains like agriculture, military, disaster management as well as the cleaning industry and aerospace. Mobile robots can be used to support people in time consuming, tedious and dangerous tasks like reconnaissance, search and rescue or surveillance. To cover an unknown environment high demands are placed on a robust and efficient application of robots. Decisions have to be made under uncertainty in a continuous effort. In addition to a proper coordination in terms of the task allocation problem, this thesis describes a new solution and analysis of the shortest path problem under uncertainties. The solution considers the minimization of costs as well as the minimization of uncertainties within a multiobjective optimization problem. The transformation results in an improved comparability of routes and offers a wide variety of pareto optimal routes.

Inhaltsverzeichnis

Abbildungsverzeichnis IV

Algorithmenverzeichnis VI

Tabellenverzeichnis VII

Abkürzungsverzeichnis VIII

Symbolverzeichnis IX

1 Einleitung 2
- 1.1 Kurzbeschreibung des Ansatzes . . . 4
- 1.2 Beitrag zur Forschung . . . 5
- 1.3 Aufbau der Dissertation . . . 6
- 1.4 Publikationen . . . 8

2 Das Terrain Covering Problem 9
- 2.1 Anwendungen des Terrain Covering Problems für mobile Agenten . . . 10
- 2.2 Eine formale Beschreibung des Terrain Covering Problems . . . 19
  - 2.2.1 Das Multiple Traveling Salesman Problem . . . 20
  - 2.2.2 Das Multiple Covering Salesman Problem . . . 21
- 2.3 Eine Abgrenzung des Terrain Covering Problems . . . 23
- 2.4 Eine Taxonomie für Multi Agent Terrain Covering Modelle . . . 28
- 2.5 Modelle für das Terrain Covering Problem . . . 37
  - 2.5.1 Multi Agent Terrain Covering Modelle . . . 37
  - 2.5.2 Multi Agent Terrain Mapping Modelle . . . 42
  - 2.5.3 Eine Entscheidungsunterstützung zur Modellierung von Terrain Covering Modellen . . . 45
- 2.6 Forschungsfragen und Untersuchungsgegenstände . . . 48
- 2.7 Zusammenfassung . . . 51

3 Das Multi Agent Robotic System 57
- 3.1 Der Agent als Bestandteil eines Systems . . . 59
- 3.2 Das Multi Agent System . . . 61
  - 3.2.1 Das Multi Agent Robotic System . . . 64
  - 3.2.2 Eigenschaften eines Multiagent (Robotic) Systems . . . 66
- 3.3 Die Robustheit bezogen auf ein Multiagent (Robotic) System . . . 70
  - 3.3.1 Vier Typen der Unsicherheit . . . 74
  - 3.3.2 Maßnahmen zur Steigerung der Robustheit . . . 77

4 Das Shortest Path Problem **82**
4.1 Das Single Objective Shortest Path Problem unter Unsicherheiten 83
4.2 Eine Transformation in das Multiobjective Shortest Path Problem (unter Unsicherheiten) 86
4.2.1 Eine Einführung in das Mehrkriterielle Optimierungsproblem 87
4.2.2 Die Ant Colony Optimization Metaheuristik – Eine Metaheuristik zur Lösung des Multiobjective Shortest Path Problem unter Unsicherheiten 92
4.2.2.1 Das Grundmodell 93
4.2.2.2 Erweiterungen für das Multiobjective Optimization Problem . . . 98
4.2.2.3 Die BiCriterionAnt Heuristik 102
4.2.3 Zusammenfassung 105
4.3 Eine experimentelle Untersuchung des transformierten Single Objective Shortest Path Problem unter Unsicherheiten 106
4.3.1 Annahmen 106
4.3.2 Funktionsweise 109
4.3.3 Der IRace Algorithmus - Eine offline Parameterkonfiguration 112
4.3.3.1 Untersuchungsgegenstand und Metriken 116
4.3.3.2 Experimentaufbau 119
4.3.3.3 Analyse und Auswertung 122
4.4 Zusammenfassung 128

5 Ein Terrain Covering Modell unter Unsicherheiten **131**
5.1 Das Stochastic Multiobjective-Terrain Covering Modell - Ein onlinebasiertes Multi-Robot Terrain Covering Modell unter Unsicherheiten 131
5.1.1 Annahmen 132
5.1.2 Maßnahmen zur Steigerung der Robustheit 136
5.1.3 Funktionsweise 138
5.2 Eine experimentelle Untersuchung des Stochastic Multiobjective-Terrain Covering Modells 142
5.2.1 Vergleichsmodelle 142
5.2.1.1 A*-Terrain Covering Modell 143
5.2.1.2 Node Counting-Terrain Covering Modell 144
5.2.2 Metriken 146
5.2.3 Untersuchungsgegenstände und Hypothesen 148
5.2.4 Experimentaufbau 149
5.2.5 Analyse und Auswertung 150
5.2.5.1 Anzahl eingesetzter Agenten 161
5.2.5.2 Kommunikationsreichweite 163
5.2.5.3 Systemumgebung 166
5.2.5.4 CENETIX Experimentserie 167
5.3 Zusammenfassung 182

6 Zusammenfassung & Ausblick **185**

Literaturverzeichnis **VII**

Anhang **XXV**

A Mehrkriterielle Optimierung XXV
B Programmiertechnische Umsetzung XXXII
C IRace Ergebnisse XXXVIII

Abbildungsverzeichnis

Abbildung 1.1: Aufbau der Dissertation und Forschungsplan 7

Abbildung 2.1: Anwendungsfelder des Terrain Covering 16
Abbildung 2.2: Unbemannte und autonome Agenten für das Terrain Covering Problem 17
Abbildung 2.3: Teilprobleme des Terrain Covering 23
Abbildung 2.4: Ausführungsschritte für das Terrain Covering Problem 27
Abbildung 2.5: Taxonomie der Terrain Covering Modelle 28

Abbildung 3.1: Taxonomie von Fehlerquellen im Terrain Covering 73
Abbildung 3.2: Systemunsicherheiten Typ I bis III 77

Abbildung 4.1: Dominanz und Pareto-Optimalität 90
Abbildung 4.2: Grundidee der ACO Metaheuristik 93
Abbildung 4.3: Pheromonaktualisierung für MACO Heuristiken 101
Abbildung 4.4: Strömungsinformation in einer Systemumgebung 108
Abbildung 4.5: Modell der Strömungskosten für eine approximierte Umgebungszerlegung 109
Abbildung 4.6: Racing Ansatz zur Ermittlung einer Parameterkonfiguration 116
Abbildung 4.7: Testumgebung zur experimentellen Untersuchung des Shortest Path Problem 120
Abbildung 4.8: IRace - Entwicklung der Parameterausprägungen für das SSPP . . . 123
Abbildung 4.9: IRace - Paarweise Gegenüberstellung der Parameterausprägungen für das SSPP 125
Abbildung 4.10: BiCriterionAnt - Lösungsgüte auf verschiedenen Systemumgebungen für das SSPP 127
Abbildung 4.11: BiCriterionAnt - Lösungsraum für eine SSPP-Instanz 129

Abbildung 5.1: Quadtree Zerlegung 134
Abbildung 5.2: Fahrkosten für das Node Counting-Terrain Covering Modell 145
Abbildung 5.3: Experimentinitialisierung - Terrain Covering Modelle 151
Abbildung 5.4: Terrain Covering Modelle - Exploration [frei|32|5] 153
Abbildung 5.5: Terrain Covering Modelle - Mehrfachabdeckung [frei|32|5] 154
Abbildung 5.6: Terrain Covering Modelle - Redundanzen [frei|32|5] 155
Abbildung 5.7: A* Terrain Covering Modell - Abdeckung [frei|32|5] 156
Abbildung 5.8: SMO Terrain Covering Modell - Abdeckung [frei|32|5] 157
Abbildung 5.9: NC Terrain Covering Modell - Abdeckung [frei|32|5] 158
Abbildung 5.10: NCs Terrain Covering Modell - Abdeckung [frei|32|5] 159
Abbildung 5.11: Möglichkeiten einer Abdeckung unter Unsicherheiten 160

Abbildung 5.12: Terrain Covering Modelle - Energieverbrauch [frei|32|5] 160
Abbildung 5.13: Terrain Covering Modelle - Kosteneffizienz [frei|32|5] 162
Abbildung 5.14: SMO-TCM - Zellenklassifikation [frei|32|5] 163
Abbildung 5.15: Terrain Covering Modelle - Exploration [frei|32|1/3/5] 164
Abbildung 5.16: Terrain Covering Modelle - Energieverbrauch [frei|4/8/32|5] 165
Abbildung 5.17: SMO-TCM - Energieverbrauch [frei|4/8/32|5] 165
Abbildung 5.18: Terrain Covering Modelle - Energieverbrauch [frei/outdoor/indoor|32|5] 166
Abbildung 5.19: SMO-TCM - Energieverbrauch [frei/outdoor/indoor|32|5] 167
Abbildung 5.20: A* Terrain Covering Modell - Abdeckung [outdoor|32|5] 168
Abbildung 5.21: SMO Terrain Covering Modell - Abdeckung [outdoor|32|5] 169
Abbildung 5.22: NC Terrain Covering Modell - Abdeckung [outdoor|32|5] 170
Abbildung 5.23: NCs Terrain Covering Modell - Abdeckung [outdoor|32|5] 171
Abbildung 5.24: A* Terrain Covering Modell - Abdeckung [indoor|32|5] 172
Abbildung 5.25: SMO Terrain Covering Modell - Abdeckung [indoor|32|5] 173
Abbildung 5.26: NC Terrain Covering Modell - Abdeckung [indoor|32|5] 174
Abbildung 5.27: NCs Terrain Covering Modell - Abdeckung [indoor|32|5] 175
Abbildung 5.28: Live-Experiment in der Bucht von San Francisco 176
Abbildung 5.29: Experimentinitialisierung - CENETIX Experimentserie 177
Abbildung 5.30: Terrain Covering Modelle - Exploration [San Francisco|32|5] 178
Abbildung 5.31: Terrain Covering Modelle - Mehrfachabdeckung [San Francisco|32|5] 179
Abbildung 5.32: Terrain Covering Modelle - Abdeckung [San Francisco|32|5] Teil 1 . 180
Abbildung 5.33: Terrain Covering Modelle - Abdeckung [San Francisco|32|5] Teil 2 . 181
Abbildung 5.34: Terrain Covering Modelle - Energieverbrauch [San Francisco|32|5] . 182
Abbildung 5.35: Terrain Covering Modelle - Energieverbrauch [frei/outdoor/indoor/-San Francisco|32|5 183

Abbildung A.1: Überlegenheitsrelationen XXVII
Abbildung A.2: Konvergenz- und Diversitätsmetrik XXX
Abbildung B.3: Implementierung des SMO-TCM XXXII
Abbildung B.4: Implementierung des BiCriterionAnt Moduls XXXIV
Abbildung B.5: Implementierung des MRQSimUncertainty Moduls XXXVII

Algorithmenverzeichnis

Algorithmus 4.1: Ant Colony Optimization – Grundalgorithmus 102
Algorithmus 4.2: Multiple Ant Colony Optimization – Grundalgorithmus 105
Algorithmus 4.3: BiCriterionAnt – Routenplanung 116
Algorithmus 4.4: Grundalgorithmus für IRace 118

Algorithmus 5.1: Grundalgorithmus für SMO-TCM 144
Algorithmus 5.2: Auktionsalgorithmus für SMO-TCM 146
Algorithmus 5.3: A*-Algorithmus zur Bestimmung der kürzesten Route 148
Algorithmus 5.4: Modifiziertes Node Counting-Terrain Covering Modell 150

Tabellenverzeichnis

Tabelle 1.1: Barrieren bezüglich des Einsatzes von Robotern 4

Tabelle 2.1: Vergleich von Terrain Covering Modellen hinsichtlich der Eigenschaften für ein MARS 45

Tabelle 2.2: Auswirkungen verschiedener Modellierungsansätze auf die Eigenschaften eines TCM (Teil 1) 47

Tabelle 2.3: Auswirkungen verschiedener Modellierungsansätze auf die Eigenschaften eines TCM (Teil 2) 48

Tabelle 2.4: Entscheidungsunterstützung zur Modellierung eines MA(R)S 51

Tabelle 2.5: Vergleich von TCMs und TMMs hinsichtlich der Modellierung . . . 58

Tabelle 4.1: IRace - Experimentaufbau für das SSPP 123

Tabelle 4.2: IRace - Ermittelte Parameterkonfigurationen für das SSPP 127

Tabelle 5.1: Experimentaufbau zur Überprüfung der Terrain Covering Modelle . 154

Tabelle 6.1: Zusammenfassung der Metriken für eine Pareto-Front XXXII

Abkürzungsverzeichnis

ACO	Ant Colony Optimization
ANN	Artificial Neural Network
BFS	Breadth First Search
CENETIX	Center for Network Innovation and Experimentation
COMTESSA	Core Competence Center For Operations Research Tenacity - Management - Excellence Safety & Security ALLIANCE
CSP	Covering Salesman Problem
DFS	Depth First Search
DM	Decision Maker
GA	Genetic Algorithm
MACO	Multiple Ant Colony Optimization
MANET	Mobile Adhoc Network
MARS	Multiagent Robotic System
MAS	Multiagent System
MCSP	Multiple Covering Salesman Problem
MOO	Multiobjective Optimization
MOOP	Multiobjective Optimization Problem
MOSPP	Multiobjective Shortest Path Problem
MTSP	Multiple Traveling Salesman Problem
NC	Node Counting
NCs	Node Counting Stochastic
PF	Pareto Frontier
PM	Pareto-Menge
SMO	Stochastic Multiobjective
SOO	Singleobjective Optimization
SOOP	Singleobjective Optimization Problem
SOSPP	Singleobjective Shortest Path Problem
SR	Swarm Robotic
SSPP	Stochastic Shortest Path Problem
TC	Terrain Covering
TCM	Terrain Covering Model
TCP	Terrain Covering Problem
TMM	Terrain Mapping Model
TSP	Travelling Salesman Problem

Symbolverzeichnis

Allgemein

$\sim$	Proportionalität
$\bullet_R$	Großer tiefgestellter Index R fungiert als zusätzliche Kennung einer Variable
$\bullet_i$	Kleiner tiefgestellter Index i bezeichnet ein Element einer Menge
$\bullet_{i,j}$	Tiefgestellte Indizes i, j bezeichnen den Eintrag einer Matrix an der Stelle i-te Zeile, j-te Spalte
A	Großer, kursiv gedruckter römischer Buchstabe bezeichnet eine Menge
a	Kleiner kursiv gedruckter römischer Buchstabe bezeichnet ein Element einer Menge
$\mathbf{b}$	Kleiner, fett gedruckter römischer Buchstabe bezeichnet einen Vektor
$\lvert A \rvert$	Kardinalität einer Menge
$\lvert\lvert \bullet \rvert\rvert$	Distanzmetrik

Terrain Covering

Ξ	Systemumwelt
Ω	Systemumgebung
$\Omega_{x,y}$	Maximale Ausdehnung der Systemumgebung
$z_i \in Z$	Zelle
$z_S \in Z$	Startzelle
$z_G \in Z$	Zielzelle
$a_k \in A$	Agent
$\Phi_{a_k}(z_i, t)$	Nachbarschaft einer konkreten Zelle z_i für einen Agenten a_k zum Zeitpunkt t
$r_i \in R$	Route (Lösung der Definitionsmenge)
$\mathbf{d}_R(z_i)$	Strömungsrichtungsvektor für eine Zelle z_i
$t_i \in \mathcal{T}$	Aufgabe
$\mathcal{M}_O$	$(0, 1)$-Hindernismatrix
$\mathcal{M}_C$	$(0, 1)$-Abdeckungsmatrix
$b_i \in \mathcal{B}$	Gebote

Shortest Path Problem

G	Graph
$v_i \in V$	Knoten
$e_i \in E$	Kante
$\mathfrak{a}_i \in \mathfrak{A}$	Aktion eines Agenten
$\mathfrak{z}_i \in \mathfrak{Z}$	Zustand eines Agenten

$\mathfrak{s}_i \in \mathfrak{S}$	Szenario
$p_i \in P$	Pfade
$c(p_i)$	Kosten für einen Pfad

Ameisenalgorithmen

$a_k \in A$	Ameise
$\mathcal{P}$	Pheromonmatrix
$\rho(z_i)$	Pheromoninformationen für die Zelle z_i
$\rho_{\min}, \rho_{\max}$	Obere und untere Grenze für die Pheromonwerte
$\mathcal{H}$	Heuristikmatrix
$\eta(z_i)$	Heuristische Information für die Zelle z_i
λ_{a_k}	Individueller Gewichtungsparameter für jede Ameise a_k
α, β	Übergeordnete Gewichtungsparameter
$p(z_i)$	Auswahlwahrscheinlichkeit einer z_i
$r_i \in R$	Route (Lösung der Definitionsmenge)
$s_i \in S$	Kodierte Route (Lösung der Zielmenge)
$S_{ND} \subseteq S$	Nicht dominierte Lösungen
ϱ_E	Evaporationsrate
ϱ_{ANZ}	Gesamtpheromonmenge, die in jeder Generation auf Lösungen $s_i \in S_{ND}$ abgelegt werden können
$\Delta\rho(z_i)$	Konkrete Pheromonmenge, die auf eine Lösungskomponente abgelegt wird
$\max_{\text{Generation}}$	Maximale Anzahl an Generationen

Mehrkriterielle Optimierung

$\prec$	Starke Dominanz
$\preceq$	Schwache Dominanz
$\|\|$	Unvergleichbarkeit

1. Einleitung

> Geschrieben steht: "Im Anfang war das Wort!"
> Mir hilft der Geist! Auf einmal seh ich Rat
> und schreib getrost: Im Anfang war die Tat!
> Johann Wolfgang von Goethe (1749–1832)

Die Weiterentwicklung von Robotersystemen basiert auf dem Potenzial, den Menschen in gefährlichen, zeitaufwändigen und monotonen Aufgaben unterstützen zu können. In einfach strukturierten Anwendungen in denen gleichförmige Aktionen ausgeführt werden müssen und die Systemumwelt vollständig bekannt ist, unterstützen schon heute eine Vielzahl von verschiedenen Robotersystemen den Menschen. Vor allem in der Industrie werden Robotersysteme erfolgreich in der Montage, der Herstellung von Produkten, bei denen ein hoher Grad von Genauigkeit benötigt wird, sowie zum Transport von Gütern in Warenhäusern eingesetzt [PwC14]. Es wird geschätzt, dass der weltweite Markt zur Herstellung und Nutzung von industriellen Robotern im Jahr 2020 ein jährliches Volumen von 41 Milliarden US Dollar erreicht [Cha14]. Des Weiteren ist im Bereich der Pflege und Therapie mit einem vermehrten Einsatz von Robotersystemen zu rechnen. Robotersysteme können mit Menschen interagieren und das Stressniveau von Patienten reduzieren sowie die Kommunikation erhöhen [WS07]. Weitere wichtige Anwendungsgebiete liegen in den Bereichen von Servicedienstleistungen, der Raumfahrt und dem Militär [ARo13].

Trotz der weitreichenden Anwendungsgebiete von Robotersystemen bestehen Barrieren, die derzeit einem Einsatz von (autonomen) Robotersystemen entgegenstehen. Vor allem mit steigender Komplexität der Szenarien, aufgrund unvollständiger oder falscher Informationen bezüglich der Systemumwelt, nehmen die Herausforderungen hinsichtlich einer autonomen Steuerung von Robotersystemen in einer Systemumwelt zu. Im gleichen Maße muss die Ausprägung von einem koordinierten und autonomen Verhalten zwischen mehreren Agenten organisiert werden. Um Robotersysteme ebenfalls in einer unstrukturierten und unbekannten Systemumwelt erfolgreich einsetzen zu können, müssen geeignete Ansätze entwickelt werden, welche beispielsweise die Robustheit, Effizienz und Flexibilität sicherstellen. Die in dieser Dissertation durchgeführte Untersuchung des neu entwickelten Terrain Covering Ansatzes zur robusten Abdeckung einer unbekannten Systemumgebung tragen dazu bei, Barrieren weiter abzubauen. Vor allem wird ein positiver Einfluss auf die Weiterentwicklung von Methoden zur Ausprägung von robusten Verhaltensweisen, einschließlich einer robusten Entscheidungsfindung unter Unsicherheiten genommen. Hierbei wird ein mehrkriterieller Ansatz verfolgt, der generisch unterschiedliche Ziele in der Optimierung berücksichtigt und somit die Flexibilität der Robotersysteme sowie die Anpassung an verschiedene Anwendungen verbessert. Ferner nimmt die optimierte Entscheidungsfindung einen positiven Einfluss auf den Energieverbrauch. Aufgrund des reduzierten Energieverbrauchs kann die Einsatzzeit der Robotersysteme verlängert werden. Darüber hinaus ist anzumerken, dass eine optimierte Entscheidungsfindung mit einer zunehmenden Systemkomplexität einhergeht.

Barriere	Potenzial
(Wahrnehmung des) Fähigkeitsspektrums von Robotern	+
Entwicklungs- und Nutzungskosten	∘
Gesetzgebung zur Nutzung von Robotern	∘
Zertifizierung und Validierung von Robotern (in sicherheitskritischer Systemumgebung)	∘
Zuverlässigkeitsüberprüfung von Robotern in Anwendungen	∘
Inbetriebnahme und Bedienbarkeit von Robotern	∘
Überwachungsmöglichkeit von Robotern	∘
Systemkomplexität von Robotern	-
Robustes Verhalten von Robotern in nicht deterministischer Systemumgebung	+++
Entscheidungsfindung unter Unsicherheiten	+++
Flexibilität und Anpassung von Robotern an Anwendungen	++
Energieniveau hinsichtlich Einsatzzeit in der Anwendung	+
Fortbewegungsmöglichkeiten von Robotern	∘

Tabelle 1.1.: Einfluss dieser Dissertation auf den Abbau der Barrieren bezüglich des Einsatzes von Robotern. Skala: erheblich negativer Einfluss (- - -); neutraler/kein Einfluss (∘); erheblich positiver Einfluss (+++).

Grundsätzlich überwiegen die positiven Potenziale und tragen dazu bei, dass das Fähigkeitsspektrum von Robotern verbessert wird und somit im gleichen Maße zu einer stärkeren Wahrnehmung führt. Die wesentlichen Barrieren bezüglich des Einsatzes von (autonomen) Robotersystemen sind in der Tabelle 1.1 zusammengefasst [ARo13, SPA15].

Die Untersuchungen in dieser Dissertation werden anhand des Terrain Covering Problems (TCP) vorgestellt und überprüft. Das TCP repräsentiert ein aktives Forschungsfeld, welches durch Ereignisse und demnach resultierenden Anwendungen auch zukünftig weiter an Bedeutung zunehmen wird. Hierbei werden autonome Robotersysteme in eine unbekannte Systemumgebung platziert, die anschließend koordiniert und vollständig befahren wird. Es wird das Ziel verfolgt, die bereits zur Verfügung stehenden Informationen aus der Systemumgebung geeignet in die Entscheidungsfindung, insbesondere für die Routenplanung und Aufgabenkoordination, einzubeziehen. Aus diesem Grund wird ein neuer Ansatz zur Lösung des Shortest Path Problem unter Unsicherheiten vorgestellt. Der in dieser Dissertation neu entwickelte Terrain Covering Ansatz bietet die Möglichkeit mit Hilfe von autonomen Robotersystemen verschiedene Anwendungen geeignet zu unterstützen. Nachfolgend sind zwei prägnante Beispiele aufgeführt, für die sich der Einsatz von Robotersystemen positiv auf die Bearbeitung der übergeordneten Aufgabe auswirkt.

- **Aufklärung nach den Erdbeben in Italien:**
  Am 20. Mai 2012 und an den darauffolgenden Tagen erschütterte eine Vielzahl von Erdbeben den Norden von Italien, zwischen Modena und Bologna mit einem Radius von 50 km. Mehr als 900.000 Menschen waren unmittelbar betroffen. Mindestens 16 Menschen verloren ihr Leben und die Erdbeben verursachten massive Schäden an Gebäuden. Aufgrund von Nachbeben mussten 3000 Menschen evakuiert werden. Es war eine nationale Aufgabe, die Sicherheit und Evakuierung der Bevölkerung zu organisieren. Des Weiteren haben die regionalen Feuerwehren und Sicherheitskräfte einen großen Beitrag zum Katastrophenmanagement geleistet [CNN12]. Bereits Ende Mai ist das EU-geförderte Projekt Natural Human-Robot Interaction in Dynamic Environments (NIFTi) zur Unterstützung angefordert worden. Es sind Teams von Wissenschaftlern und bis zu vier Land- und Luftfahrzeugen eingesetzt worden, um eine Aufklärung der Gesamtlage zu unterstützen und eingestürzte Gebäude auf deren Statik und sichere Wege zu überprüfen [KTL+12, Fra16].

- **Suche nach dem Air Malaysia Flugzeug MH370:**
  Am 8. März 2014 endete der Air Malaysia Flug MH370 von Kuala Lumpur nach Peking in einer Katastrophe. Das Flugzeug stürzte in den Indischen Ozean. An Bord des Flugzeuges waren neben den zwölf Besatzungsmitgliedern weitere 227 Fluggäste. An der Suche nach dem Flugzeug beteiligten sich bisher 14 Nationen mit mehr als 43 Schiffen und 58 Flugzeugen. Bis heute wurde das Flugzeug nicht gefunden [Neu14]. Die Suchmission ist die bisher größte multinationale Luft- und Wassersuche. Experten schätzen die Kosten auf mehrere Hundert Millionen Dollar [Pea14]. Aufgrund verschiedener Faktoren stellt sich die Suche nach dem Flugzeug besonders schwierig dar. Einerseits ist das Suchgebiet mit maximal 7.68 Millionen Quadratkilometern oder 11 % des Indischen Ozeans sehr groß. Die Konzentration auf ein kleineres Suchgebiet hat sich fortlaufend geändert [Mal14]. Zudem liegt das Suchgebiet weit entfernt vom Festland. Resultierend ist die Versorgung der Einsatzkräfte aufwendig und zeitintensiv. Andererseits sendet die Blackbox nur 30 Tage ein Signal [Har14].

Resultierend ist für die beschriebenen oder ähnliche Anwendungen ein effizienter und robuster Einsatz von mehreren Robotersystemen notwendig, um beispielsweise Kosten im Rahmen einer Untersuchung der Umgebung zu reduzieren. Für eine effiziente Organisation der Robotersysteme muss das zugrundeliegende TCP betrachtet werden. Aufgrund von unvollständigen Informationen bezüglich der Systemumwelt können Ineffizienzen in der Koordination sowie Routenplanung, negative Auswirkungen auf die Robustheit und ferner auf die Zielerreichung resultieren. Es können verschiedene Anwendungen des TCP identifiziert werden. Die ersten Anwendungen sind in der Landwirtschaft [LLH11, YAB+13] zu finden. Hierbei befährt ein autonomes Fahrzeug einen Acker und verfügt über vollständige Informationen hinsichtlich der Systemumgebung. Ebenfalls erfahren Reinigungsroboter eine frühe Anwendung. Eine (unbekannte) Systemumgebung wird mit Hilfe von mehreren Robotern koordiniert und vollständig befahren [UMN97, LY02, HAS12]. Aktuelle Anwendungen kommen vor allem im Militär [JRBM02, Yam04, SBE08] und Katastrophenschutz [Rod05, MKP+08, LTB+10, KTL+12] zum Tragen. Die Anforderungen an die Robustheit und Effizienz nehmen insbesondere bei Unsicherheiten bezüglich der Systemumwelt einen besonderen Stellenwert ein. Des Weiteren existieren herausfordernde Anwendungen für Weltraummissionen [MLB07]. Hierbei kann nur in einem sehr begrenzten Ausmaß ein Einfluss auf die Roboter genommen werden. Unter Umständen muss die Möglichkeit bestehen, zusätzliche Verhaltensweisen selbstständig erlernen zu können.

1.1. Kurzbeschreibung des Ansatzes

In dieser Dissertation wird ein neues Terrain Covering Modell (TCM) eingeführt, um eine unbekannte Systemumgebung mit Hilfe eines autonomen Multiagenten (Roboter-)Systems (M(A)RS[1]) befahren zu lassen. Das TCP untergliedert sich für den Einsatz von mehreren Agenten in das Task Allocation Problem und das Shortest Path Problem (SPP). Zu Beginn der vollständigen Abdeckung wird die Systemumgebung in gleichförmige Zellen zerlegt. Für jede dieser Zellen wird eine Aufgabe definiert, die die Abdeckung der Zelle repräsentiert. Anschließend wird jeder Agent mit einer gleichen Aufgabenreihenfolge initialisiert und in die Systemumgebung platziert. Mit Hilfe einer hybriden Organisationsform werden die Agenten über einen auktionsbasierten Algorithmus

[1] Engl. Multiagent (Robotic) System

koordiniert. Die hybride Organisationsform implementiert dezentrale und zugleich zentrale Elemente einer Koordination der Agenten. Resultierend können Entscheidungen über alle verfügbaren Agenten hinweg getroffen werden. Zugleich kann jeder Agent selbstständig Entscheidungen herbeiführen.

Für jede Aufgabe wird eine Auktion erstellt und ein Informationsaustausch bezüglich der Systemumgebung vorgenommen. Alle Agenten in Kommunikationsreichweite nehmen an der Auktion mit einem Gebot teil. Das Gebot repräsentiert eine geeignete Route zur Aufgabenzelle. Aufgrund der unvollständigen Informationen und Unsicherheiten bezüglich der Systemumgebung wird das Stochastic Shortest Path Problem betrachtet. Ferner wird eine robuste Route gesucht, die unter Betrachtung aller möglichen Szenare eine gute Kostenstruktur aufweist. Die Kosten einer Route repräsentieren beispielsweise den notwendigen Energieverbrauch, der durch einen Agenten für die Befahrung aufgebracht werden muss. Die Suche nach einer Route basiert auf der Mehrkriteriellen Optimierung. Mittels der Ant Colony Optimization (ACO) Metaheuristik [Dor92] werden Pareto-optimale Routen bestimmt, die jeweils die Kosten und deren Eintrittswahrscheinlichkeit betrachten. Abschließend wird aus der Menge der Pareto-optimalen Routen eine geeignete ausgewählt und befahren. Es werden von den Agenten fortlaufend Auktionen erstellt, bis alle Aufgaben abgearbeitet sind. Das TCM gewährleistet stets eine vollständige Abdeckung aller befahrbaren Zellen. Es werden weitere Ansätze integriert, um die Robustheit und Effizienz der Abdeckung zu steigern.

1.2. Beitrag zur Forschung

Die nachfolgende Dissertation erfüllt im wesentlichen drei Beiträge zur aktuellen Forschung auf den Gebieten des Einsatzes von einem MA(R)S zur Lösung des TCP. Hierzu wird die Forschung zum Shortest Path Problem unter Unsicherheiten um einen neuen Lösungsansatz weiterentwickelt.

- **Erweiterte Taxonomie von Multiagent Terrain Covering Modellen:**
  Es wird eine erweiterte Taxonomie von Multiagent TCMs eingeführt (Kapitel 2). Die Taxonomie unterscheidet Gestaltungsdimensionen und umfasst verschiedene Modellierungsansätze bezüglich der Optimierung, den Agenten und Sensorsystemen sowie der Systemumgebung. Ferner werden bestehende Multiagent TCMs in die Taxonomie eingeordnet, um eine bessere Vergleichbarkeit herbeiführen zu können. Des Weiteren wird anhand von Beispielen der Gebrauch der Taxonomie zur Entscheidungsunterstützung aufgezeigt, um einen zielgerichteten Einfluss auf die Ausprägung der Eigenschaften eines MA(R)S nehmen zu können. Resultierend ermöglicht die Entscheidungsunterstützung für verschiedene Problemstellungen und Anwendungsgebiete des TCP eine geeignete Modellierung des zugrundeliegenden TCM vornehmen zu können.

- **Entwicklung und Analyse eines neuen Lösungsansatzes für das Shortest Path Problem unter Unsicherheiten:**
  In einer unbekannten Systemumgebung liegen Unsicherheiten bezüglich der Befahrung von Teilregionen sowie den zugehörigen Kosten vor. Diesbezüglich wird ein neuer Lösungsansatz für das Shortest Path Problem unter Unsicherheiten beschrieben (Kapitel 4). Hierbei werden die Unsicherheiten und Kosten jeweils in eine Zielfunktion überführt. Resultierend wird das

Mehrkriterielle Optimierungsproblem (MOOP[2]) gelöst, mit der Möglichkeit einer Vergleichbarkeit zwischen Routen vornehmen zu können. Ferner besteht das Ziel darin, verschiedene Bereiche des gesamten Lösungsraumes zu untersuchen. Aus dem untersuchten Lösungsraum wird anschließend eine geeignete Teilmenge von Pareto-optimalen Lösungen ermittelt, um eine Entscheidungsunterstützung bereitstellen zu können. Abschließend wird der neue Lösungsansatz aufbauend auf den konzeptionellen Überlegungen anhand eines konkreten Beispiels der Routenplanung für Agenten simulationsgestützt überprüft. Neben der Definition einer geeigneten Heuristik-Parameterkonfiguration wird die Effektivität des neuen Lösungsansatzes gezeigt.

- **Entwicklung und Analyse eines neuen robusten Multiagent Terrain Covering Modells:**
  Es werden verschiedene Ansätze zur Steigerung der Robustheit aus der Systemtheorie zusammengefasst und auf den robusten sowie effizienten Einsatz eines MA(R)S bezüglich dem TCP übertragen (Kapitel 3). Ferner werden die Ansätze in einem neuen Multiagent TCM zusammengefasst (Kapitel 5). Es wird das Ziel verfolgt, eine unbekannte Systemumgebung möglichst robust und effizient mit Hilfe von autonomen Agenten zu befahren. Unter anderem wird der neue Lösungsansatz für das Stochastic Shortest Path Problem (SSPP) integriert. Neben einem konzeptionellen Entwurf wird eine Implementierung und Analyse des neuen TCM anhand von simulationsgestützten Experimenten durchgeführt, um die positiven Auswirkungen auf die Robustheit und Effizienz zu überprüfen.

1.3. Aufbau der Dissertation

Die Dissertation gliedert sich im Wesentlichen in drei Teile und führt dabei die notwendigen Grundlagen ein, um darauf aufbauend die zuvor beschriebenen Beiträge zur aktuellen Forschung einordnen zu können. Der Aufbau der Dissertation und daraus resultierende Forschungsplan ist in der Abbildung 1.1 veranschaulicht.

- **Teil I:**
  Im ersten Teil werden mit den Kapiteln 2 und 3 das TCP und das MARS eingeführt. Für das TCP werden aktuelle Anwendungsgebiete beschrieben sowie eine Abgrenzung von Teilproblemen vorgenommen. Darauf aufbauend werden TCMs für das Multiagent Online TCP aus der Literatur vorgestellt. Hierbei werden die TCMs anhand der Eigenschaften für ein MARS hinsichtlich des Erfüllungsgrades überprüft. Die Gestaltungsdimensionen werden anschließend in eine Taxonomie überführt, welche eine bessere Vergleichbarkeit von verschiedenen Ansätzen ermöglicht. Des Weiteren ermöglicht die Taxonomie eine Identifikation der Forschungslücke und möglicher Untersuchungsgegenstände, welche als Grundlage dieser Dissertation dienen. Ausgehend von der Forschungslücke werden zuerst die Eigenschaften eines MARS eingeführt. Nachfolgend werden verschiedene Ansätze zur Steigerung der Robustheit und Effizienz aus der Systemtheorie auf das TCP übertragen.

[2]Engl. Multiobjective Optimization Problem

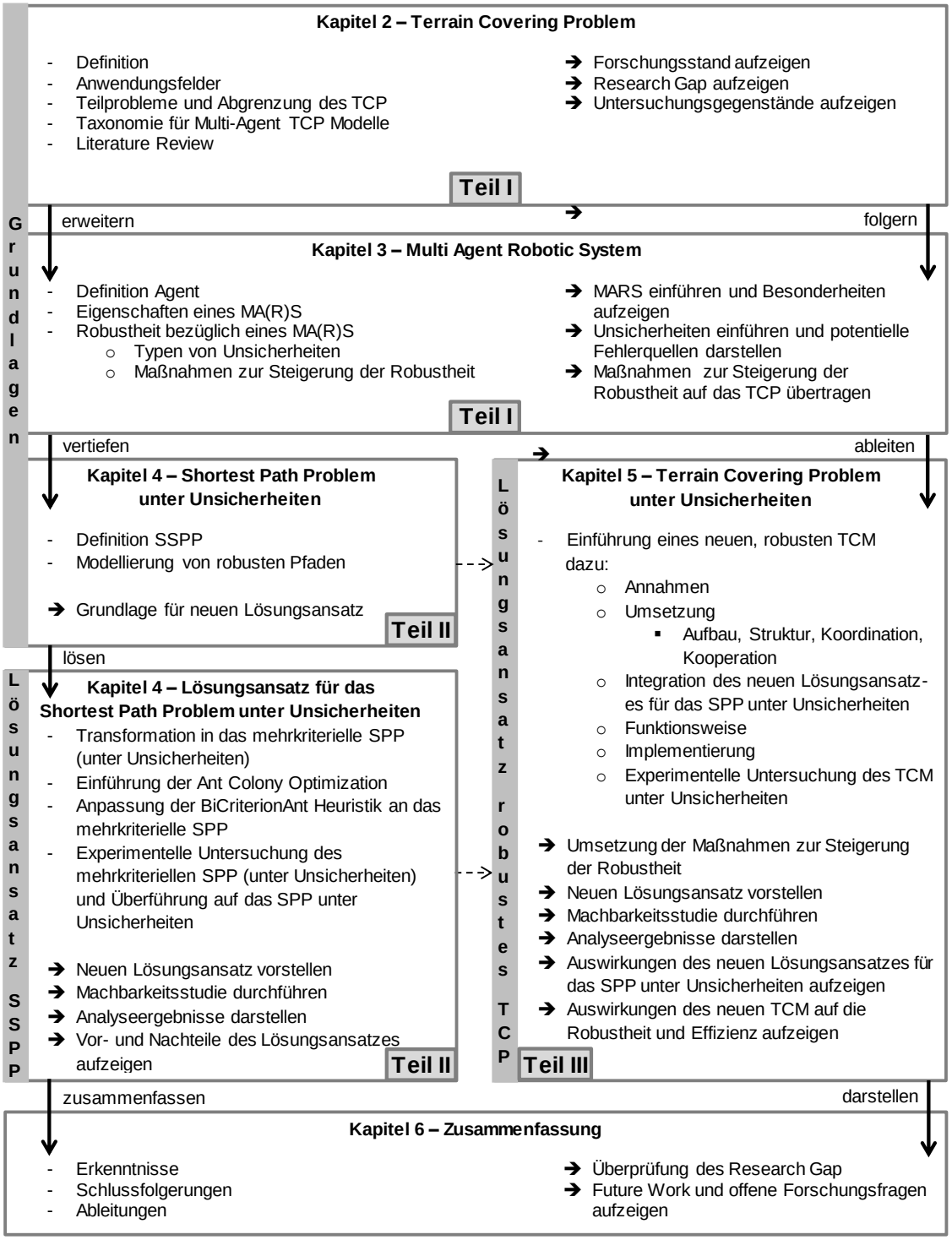


Abbildung 1.1.: Aufbau der Dissertation und Forschungsplan.

- **Teil II:**
  Im zweiten Teil wird allgemein das SPP und ein neuer Lösungsansatz für das Stochastic Shortest Path Problem (SSPP) in dem Kapitel 4 beschrieben. Der Lösungsansatz transformiert das Singleobjective Shortest Path Problem (SOSPP) unter Unsicherheiten in das Multiobjective Shortest Path Problem (MOSPP). Entsprechend werden die Unsicherheiten mit einer zusätzlichen Zielfunktion beschrieben. Hierbei werden zuerst das Mehrkriterielle Optimierungsproblem und die zugehörigen Grundlagen eingeführt. Darauf aufbauend wird die ACO Metaheuristik vorgestellt, die als Lösungsverfahren herangezogen wird. Neben dem Grundmodell und den notwendigen Erweiterungen hinsichtlich Mehrkriterieller Optimierungen, werden die Vorteile der ACO Metaheuristik zusammengefasst. Ferner wird die BiCriterionAnt Heuristik an die spezifischen Annahmen des Routenplanungsproblems im Rahmen des TCP angepasst. Abschließend wird eine simulationsgestützte Überprüfung anhand eines konkreten Beispiels der Routenplanung durchgeführt, um die Auswirkungen des neuen Lösungsverfahrens auf die Entscheidungsfindung darzustellen.

- **Teil III:**
  Im dritten Teil wird mit dem Kapitel 5 ein neues TCM für den Einsatz eines autonomen MA(R)S vorgestellt. Hierbei werden zuvor beschriebene Ansätze zur Steigerung der Robustheit und Effizienz integriert. Insbesondere werden die Auswirkungen des neuen Lösungsansatzes zum SSPP bezugnehmend auf das TCP analysiert. Zuerst werden geeignete Annahmen eingeführt, die sich an realen Anwendungen orientieren. Des Weiteren wird die Funktionsweise hinsichtlich der Selbstkoordination und Routenplanung beschrieben. Daran anschließend werden Hypothesen aufgestellt und der simulationsgestützte Experimentaufbau dargelegt. Abschließend werden aufbauend auf den durchgeführten Experimenten die Auswirkungen auf das TCP diskutiert.

1.4. Publikationen

Aus der Forschung zu dieser Dissertation sind nachfolgende Publikationen mit einem eigenen Anteil von mindestens 50 % entstanden:

Preuß, M.: *A Multi-Objective Online Terrain Coverage Approach.* In: Huisman, D., I. Louwerse und A. P. M. Wagelmans (Herausgeber): Operations Research Proceedings 2013, Seiten 347–353. Springer International Publishing, 2014.

Preuß, M., M. Dehmer und S. Pickl: *An Improved Graph-Based Decision Support Approach by employing Graph Measures.* In: Proceedings of the Networking and Electronic Commerce Research Conference, Seiten 81–84, 2014.

Preuß, M., M. Dehmer, S. Pickl und A. Holzinger: *On Terrain Coverage Optimization by Using a Network Approach for Universal Graph-Based Data Mining and Knowledge Discovery.* In: Ślęzak, D., A.-H. Tan, J. F. Peters und L. Schwabe (Herausgeber): Brain Informatics and Health, Band 8609 der Reihe Lecture Notes in Computer Science, Seiten 564–573. Springer International Publishing, 2014.

2. Das Terrain Covering Problem

Meistens gelangen die Menschen
nur durch die Folgen der Unordnung
zur Einführung der Ordnung, und Gesetzlosigkeit
führt gewöhnlich erst zu Gesetzen.
Johann Christoph Friedrich von Schiller (1759–1805)

In diesem Kapitel werden zuerst grundlegende Definitionen eingeführt und Anwendungen für das Terrain Covering Problem und dem TCP ähnliche Probleme vorgestellt, für die mobile Roboter eingesetzt werden. Hierbei werden die Gemeinsamkeiten der Anwendungen und die verschiedenen Anforderungen an die Roboter dargestellt. Ferner werden die Grenzen des Einsatzes von mobilen Robotern zusammengefasst. Daran anschließend wird eine formale Betrachtung des TCP vorgenommen sowie eine Einordnung und Abgrenzung von Teilproblemen des TCP. Des Weiteren wird eine Übersicht bezüglich bestehender Ansätze und Modelle des TCP wiedergegeben. Abschließend wird in diesem Kapitel, basierend auf den bestehenden Terrain Covering Modellen, eine Taxonomie abgeleitet, um eine Einordnung neuer Ansätze vornehmen zu können. Zudem wird eine Einordnung verschiedener Modelle des TCP entlang der definierten Gestaltungsdimensionen erleichtert. Darüber hinaus wird eine Entscheidungsunterstützung für eine anwendungsbezogene Modellierung eines TCM vorgestellt.

Das TCP besteht bereits seit den Anfängen der Menschheit. Die ersten Menschen haben beispielsweise eine Umgebung nach Nahrung, Werkzeugen, Baumaterialien und anderen Stämmen abgesucht/abgedeckt. Mit der Weiterentwicklung der Menschheit, hin zu einer intelligenteren und leistungsfähigeren Gesellschaft, sind technologische Errungenschaften entsprungen, welche dazu beitragen können, das TCP effizienter lösen zu können. Die Motivation der Automatisierung und Einführung von Robotern ist nicht allein in den resultierenden Vorteilen für die Anwendungsgebiete zu sehen, sondern vielmehr im Versuch, die Menschen von mühsamen, zeitaufwendigen und gefährlichen Aufgaben zu entbinden. Treibende Faktoren für die Weiterentwicklung der Robotertechnik sind vor allem die Sicherung der Wettbewerbsfähigkeit in einem internationalen Umfeld, verbesserte Lebensqualität einer alternden Gesellschaft sowie die Unterstützung von Einsatzkräften in lebensbedrohlichen Situationen. Es wird angenommen, dass die Robotertechnik einen hohen Stellenwert in der Gesellschaft einnehmen wird [ARo13]. Trotz der innovativen und hochtechnologischen Weiterentwicklungen in den vorangegangenen Jahrzehnten, hat sich ein allgemeiner Einsatz von autonomen Robotern für das TCP noch nicht durchgesetzt. Der Umstand ist vor allem dem komplexen und keineswegs trivialen Gesamtsystem geschuldet. Das Gesamtsystem umfasst die Systemumwelt und die einzusetzende Robotertechnik [VR95]. Aufgrund von Unsicherheiten im Gesamtsystem kann ein autonomer, effizienter, robuster und flexibler Einsatz von Robotern erschwert werden. In dieser Dissertation werden verschiedene Ansätze vorgestellt, um eine Optimierung des TCP unter Unsicherheiten erzielen zu können.

Hierbei wird ein MARS eingesetzt, welches aus eigenständig agierenden Agenten zusammengesetzt ist.

Zusammenfassend kann das TCP wie folgt definiert werden:

Definition 2.1 (Terrain Covering Problem)**.** : *Eine kontinuierliche, begrenzte Systemumgebung Ω wird durch mindestens einen Agenten $a \in A$ vollständig, durch den Besuch einer jeden frei befahrbaren Teilregion, abgedeckt, um eine übergeordnete Aufgabe erfüllen zu können. Eine Teilregion gilt als besucht, genau dann wenn sie durch die Sensoren von mindestens einem Agenten untersucht worden ist.*

Des Weiteren wird in dieser Dissertation die nachfolgende Terminologie, im Sinne des TCP, für ein MARS verwendet. Fahrzeuge und Roboter, die in realen Anwendungen zum Einsatz kommen, werden als Agenten bezeichnet.

Definition 2.2 (Agent)**.** : *Ein Agent $a \in A$ ist ein hardware- und softwarebasiertes Robotersystem, das in einer Systemumwelt agieren kann. Ein Agent ist mit einem Sensorsystem ausgestattet, um Informationen von anderen Agenten empfangen zu können. Des Weiteren kann ein Agent eine Teilmenge $B_a(Pos)$ der Systemumwelt wahrnehmen, welche abhängig von der Position Pos des Agenten in der Systemumwelt ist. Hingegen dienen Effektoren dazu, Manipulationen an der Systemumwelt vorzunehmen.*

Im Nachfolgenden werden Anwendungen des TCP erläutert. Insbesondere werden die Anforderungen an die Agenten beschrieben und auf die Gemeinsamkeiten der Anwendungen eingegangen. Ferner werden die Grenzen des Einsatzes von mobilen Agenten zusammengefasst.

2.1. Anwendungen des Terrain Covering Problems für mobile Agenten

Autonome und unbemannte Agenten werden in einer Vielzahl von verschiedenen Anwendungsfeldern des TCP genutzt. Die Anwendungsfelder reichen von der Landwirtschaft [LLH11, YAB+13] über die Raumfahrt [MLB07] bis hin zum Militär [JRBM02, Yam04, SBE08] und können in allen Dimensionen zu Wasser [HTL96, SBE08], Land [KRS04, EHL+06] und in der Luft [OM04, SKSM08] realisiert werden. In erster Linie soll der Einsatz von Agenten die Menschen in der Aufgabenerfüllung und Zielerreichung unterstützen. Vor allem gefährliche, zeitaufwändige, monotone, aber auch sehr komplexe Aufgaben können durch Agenten geeignet übernommen werden.

Zusammenfassend wird ein unbemannter und autonomer Agent wie folgt definiert:

Definition 2.3 (Unbemannter Agent)**.** : *Ein Agent a besitzt die Fähigkeit, von Menschen geplante Entscheidungen ferngesteuert auszuführen.*

Definition 2.4 (Autonomer Agent)**.** : *Ein Agent a besitzt die Fähigkeit, selbstständig, sinnvolle Entscheidungen zu treffen und Aktionen ohne jeglichen Einfluss durch einen Menschen auszuführen.*

Um die gestellten Aufgaben (teilweise) autonom erfüllen zu können, ist der Agent mit verschiedenen Sensoren und Effektoren ausgestattet. In [Yuh00] werden die Sensoren in drei verschiedene Klassen unterteilt. Mit Hilfe der Navigationssensoren kann der Agent beispielsweise eine Positionsbestimmung durchführen oder den Kurs, die Geschwindigkeit und die Systemumwelt messen. Die Systemumwelt kann unter anderem mittels taktiler oder optischer Sensoren wahrgenommen werden. Hierbei können Hindernisse, Bodenbeschaffenheit oder Entfernungen bestimmt werden. Dem Agenten ist es somit möglich, Bewegungen im Raum zu messen und eine sichere Navigation vorzunehmen. Die Klasse der Missionssensoren unterstützen den Agenten, in Abhängigkeit des Anwendungsfeldes, die gestellten Aufgaben bewältigen zu können. Zum Beispiel können den Agenten durch verschiedene Sensoren zusätzliche Fähigkeiten hinzugefügt werden, wie die Wahrnehmung und Messung von Akustik, Luftfeuchtigkeit, Rauch, Druck, Temperatur, Gasen oder Materialien. Kommunikationssensoren zur Koordination und zum Informationsaustausch werden ebenfalls den Missionssensoren zu geordnet. Sowohl die Navigationssensoren als auch die Missionssensoren gehören der Klasse der exterozeptiven[1] Sensoren an. Hingegen gehören die Systemsensoren den propriozeptiven[2] Sensoren an und dienen somit der Eigenwahrnehmung. Demzufolge ermöglichen die Systemsensoren eine Überwachung der Hardware- und Softwareanteile des Agenten. Beispielsweise können Systemparameter wie die Prozessorenauslastung oder der aktuelle Systemzustand kontrolliert werden. Ferner können allgemeine Parameter wie Beschädigungen und Energieniveau überwacht werden [SSVO09].

Neben den Sensoren verfügt der Agent über Effektoren, um sich zum einen in der Systemumwelt fortbewegen zu können. Zum anderen werden Effektoren eingesetzt, um Sensoren ausrichten oder Veränderungen an der Systemumwelt vornehmen zu können. Abhängig von der Anwendung können Effektoren unter anderem zum Greifen, Schneiden oder Bohren eingesetzt werden.

- **Klassisches Terrain Covering:**

  Für klassische Terrain Covering Aufgaben ist die Systemumwelt vollständig bekannt und statisch. Dementsprechend können die Routen bereits vor der Abdeckung optimiert werden. Hierbei können sowohl die verschiedenen Optimierungsziele als auch Eigenschaften wie die Nähe der Agenten zu einander oder Hindernisse betrachtet werden, die einen Einfluss auf die Qualität des Nachrichtenaustausches nehmen können. Die Agenten bewegen sich bei der Abdeckung auf den zuvor definierten Routen und benötigen vor allem Sensoren der Positionsbestimmung. Generell können die Sensoren und Effektoren der Agenten sehr spezifisch auf die Systemumwelt und zusätzlichen Aufgaben angepasst werden.

  Die ersten Anwendungen des Terrain Covering sind in den Bereichen der Landwirtschaft und Reinigung wieder zu finden. Diese Anwendungen zeichnen sich vor allem dadurch aus, dass die Systemumwelt bereits vor der Abdeckung bekannt und meist einfach strukturiert ist. Häufig bewegen sich die Roboter auf geplanten Routen und üben dabei verschiedene Tätigkeiten aus. Die ersten Ideen bezüglich Service- und Reinigungsroboter sind in den 1970er Jahre aufgekommen und wurden in den 1990er Jahren zu prototypischen Realisierungen umgesetzt [UMN97, FP00]. Hierbei bewegen sich die Roboter auf zufallsbedingten Routen durch die Systemumwelt und führen nur Richtungsänderungen beim Auftreten von Hindernissen aus. Der erste kommerzielle Reinigungsroboter Trilobite wurde 2001 eingeführt und basiert auf

[1] Lat. exter recipere, etwas Außen befindliches wahrnehmen
[2] Lat. proprius recipere, etwas Eigenes wahrnehmen

zufallsbedingten Routen [Ele10]. Eine optimierte Abdeckung wurde erst bei späteren Prototypen betrachtet [HAS12]. Zusammenfassend sind die Anforderungen an die Sensoren und Effektoren niedrig einzustufen. Für die meisten Szenarien sind Abstands- und Drucksensoren ausreichend.

In dem Anwendungsgebiet der Landwirtschaft hat der Japaner Kawamura 1982 den ersten autonomen Agenten entwickelt, dass zum Pflücken von Tomaten eingesetzt werden konnte [KNFU84]. Bis heute wurden zahlreiche prototypische Landwirtschaftsroboter entwickelt [LLH11, YAB+13]. Dennoch hat sich eine kommerzielle Verbreitung bisher nicht durchgesetzt. Vor allem die komplexen Teilaufgaben, wie beispielsweise das Greifen, Schneiden, Pflücken, Bestäuben und Zählen von Früchten umfassen, stellen hohe technische Anforderungen an die Sensoren und Effektoren. Aufgrund der sehr spezifischen Eigenschaften der verschiedenen Früchtesorten, können Teilaufgaben nur bedingt generalisiert werden. Resultierend sind nur anwendungsspezifische Landwirtschaftsroboter mit einem begrenzten Aufgabenspektrum möglich. Hierbei sind die Anforderungen der Robustheit und Effizienz in vielen Szenarien bisher nur unzureichend erfüllt und die Entwicklungskosten zu hoch [GZKT08].

Ein neues Anwendungsgebiet der klassischen Terrain Covering Aufgaben umfassen Agenten, die zu Ausbildungs- und Lernzwecken eingesetzt werden. Beispielsweise werden im australischen Nationalmuseum seit 2013 autonome Agenten eingesetzt, die sich selbstständig, auf definierten Routen, durch das Museum bewegen. Ausgestattet mit einer Kamera können räumlich entfernte Schüler das Museum erkunden und mit dem Museumspersonal kommunizieren [Rob13].

- **Geographie/Kartographie:**

  Das Anwendungsgebiet der Geographie und Kartographie grenzt sich zu den klassischen Anwendungsgebieten dahingehend ab, dass die abzudeckende Systemumwelt in der Regel unbekannt ist. Aus diesem Grund müssen geeignete Verfahren sowie Maßnahmen der Koordination, inklusive der Aufgabenplanung und Konfliktlösung, integriert werden, um eine vollständige Abdeckung, unter den Eigenschaften der Effizienz, Robustheit und Flexibilität, zu gewährleisten. Ferner werden hohe Anforderungen an die Sensorik gestellt, um die Positionsbestimmung in einem unbekannten Raum durchzuführen und ein Modell der Systemumwelt zu erstellen. Aufgrund der unbekannten Systemumwelt und möglicher Hindernisse müssen vor allem bei landgestützten Anwendungen die Agenten über geeignete Fortbewegungsmöglichkeiten verfügen, um unterschiedliche Geländetypen bewältigen zu können.

  Autonome Unterwasserfahrzeuge können unter anderem für das Kartographieren des Meeresbodens eingesetzt werden. Hierbei können die Unterwasserfahrzeuge nach fossilen Rohstoffen suchen. Im Jahr 2008 hat ein amerikanisches Forscherteam in dem Projekt Circum-Arctic Resource Appraisal mit Hilfe von autonomen Unterwasserfahrzeugen große Öl- und Gasreserven in der Arktis nachgewiesen [BCG+08, HR12].

  Bereits 1970 sind die ersten unbemannten Rover auf dem Mond gelandet, um eine Exploration der Mondoberfläche vorzunehmen. Sojourner war 1997 der erste Rover auf dem Mars und sammelte Daten über die Oberflächenbeschaffenheit und Umweltbedingungen. Der Rover legte eine Strecke von 0.1 Km zurück. Mit Curiosity ist die neueste Generation 2013 auf dem

Mars gelandet. Gemäß Angaben der NASA hat der Rover am 18.06.2014 bereits eine Strecke von 7.9 Km zurückgelegt [Jet14]. Aufgrund der zunehmenden Entfernung von der Erde nimmt die Dauer der Nachrichtenübermittlung zu und eine Fernsteuerung ist nur noch bedingt möglich. Für eine robuste, autonome Steuerung der Rover werden vor allem die Fähigkeiten der Datenfusionierung, Konfliktlösung und Optimierung benötigt. Neben einer robusten Route, die sicher befahren werden kann, müssen unter anderem der Energieverbrauch, die Anzahl der Richtungsänderungen und die Ausrichtung zur Sonne betrachtet werden. Zudem sind die Sensoren und Effektoren extremen Umwelteinflüssen wie Temperatur, Luftdruck, Korrosion oder Strahlung ausgesetzt. Bei Beschädigungen muss der Rover in der Lage sein selbstständig Anpassungen vorzunehmen, so dass die Bearbeitung der Aufgaben dennoch fortgeführt werden kann [MLB07].

- **Aufklärung:**

  Das Terrain Covering im Anwendungsgebiet der Aufklärung stellt hohe Anforderungen an die autonomen Agenten, und ist dem Anwendungsgebiet der Geographie/Kartographie ähnlich. In der Regel ist die Systemumwelt teilweise oder vollständig unbekannt. Es müssen ebenfalls geeignete Verfahren der Koordination vorhanden sein, um eine optimierte Abdeckung vorzunehmen. Oftmals agieren die Agenten in einer gefährlichen Systemumwelt. Aus diesem Grund müssen die Agenten zum einen die Systemumwelt möglichst schnell abdecken und zum anderen mögliche Bedrohungen reduzieren. Zusammenfassend liegt eine erhöhte Wahrscheinlichkeit vor, dass ein Agent durch Fremdeinwirkung ausfällt. In diesem Fall müssen sich die Agenten selbstständig reorganisieren können.

  Autonome Luft- und Landfahrzeuge können vor einer militärischen Operation ein unbekanntes Gebiet aufklären, um ein umfassendes Lagebild zu erstellen. Das Lagebild kann beispielsweise strategische Landmarken, Wege, Gebäude oder Personen enthalten. Hierbei sollten die Agenten möglichst klein und leise sein, damit sie nicht entdeckt werden. Mit Hilfe der Agenten können die Gefahren für die eigenen Soldaten reduziert und ein Informationsvorsprung erzielt werden [JRBM02].

  Nach einer nuklearen Katastrophe, wie beispielsweise 2011 in Fukushima können autonome Agenten zum Einsatz kommen. Die Agenten können flexibel und agil in Gebieten operieren, die für Menschen unzugänglich sind. Hierbei können unter anderem Beschädigungen an Reaktoren aufgeklärt und radioaktiver Müll beseitigt werden [Par94]. Ähnlich werden im US-Militär Landfahrzeuge eingesetzt, um chemische und radioaktive Kampfstoffe aufzuspüren [Yam04].

  Autonome Unterwasserfahrzeuge können hingegen zur Aufklärung und Beseitigung von Seeminen eingesetzt werden. Schätzungen der US-Navy ergeben, dass ein Entminen der Straße von Hormus mit einer Ausdehnung von zirka 200 mal 50 Km, mittels herkömmlicher Verfahren, Wochen bis Monate andauern könnte [Cap08]. Autonome Unterwasserfahrzeuge können hingegen die Systemumwelt ohne Gefahren für Menschen absuchen. Aufgrund der geringeren Kosten können mehrere autonome Unterwasserfahrzeuge verschiedene Regionen parallel abdecken. Resultierend kann ein größeres Gebiet in gleicher Zeit aufgeklärt werden [SBE08]. Des Weiteren können Unterwasserfahrzeuge nach einer Überschwemmung oder einem Erdbeben zur Aufklärung von Schäden an Dämmen und Brücken eingesetzt werden [Gui11].

- **Suchen und Retten:**

  Die Terrain Covering Aufgaben im Anwendungsgebiet des Suchen und Retten sind die der Aufklärung ähnlich. Hierbei wird das Ziel verfolgt, dass zu einen die Systemumwelt vollständig abgedeckt wird und zum anderen bestimmte Zielobjekte möglichst schnell aufgefunden werden. Dementsprechend müssen Informationen aus der Systemumwelt, hinsichtlich der Zielobjekte, kontinuierlich ausgewertet und die Agenten entsprechend koordiniert werden. Beispielsweise sollten die Agenten disloziert die Systemumwelt abdecken, damit die Wahrscheinlichkeit, ein Zielobjekt zu finden, erhöht wird. Resultierend geht ein erhöhter Koordinationsaufwand einher. Des Weiteren ist die Systemumwelt für Landfahrzeuge oftmals schwierig zu befahren, weil Such- und Rettungsaktionen in der Regel nach Katastrophen eingeleitet werden und die Systemumwelt unstrukturiert ist. Hingegen können Unterwasser- und Luftfahrzeuge besonders gut für flächendeckende Such- und Rettungsaufgaben eingesetzt werden. Die Agenten können zudem Informationen über Infrastrukturen oder die materielle und medizinische Versorgung sammeln, um das Lagebild zu verbessern [Mur14, PDP14].

  Autonome und unbemannte Landfahrzeuge können bei der Suche von Menschen in eingestürzten Gebäuden unterstützend eingesetzt werden. Vor allem für Einsatzkräfte unzugängliche oder gefährliche Passagen können überwunden werden. Zusätzlich können die Sensoren zur Aufklärung von Verschütteten und Rettungswegen sowie der Gebäudestabilität genutzt werden. Nach dem Einsturz des World Trade Centers 2001 sind erstmals vier ferngesteuerte Landfahrzeuge im Rahmen eines urbanen Such- und Rettungseinsatzes eingesetzt worden. Von einem autonomen Einsatz ist aufgrund der komplexen Umwelt, eingeschränkten Sensoren und Akzeptanz durch Einsatzkräfte abgesehen worden. Es sind Wege in das Innere der Trümmer und Gebäude lokalisiert worden [CM03b, Mur04]. Des Weiteren sind beispielsweise 2007 nach dem Einsturz der Crandall Canyon Mine in den USA, 2009 nach dem Einsturz des Kölner Stadtarchives und 2012 nach dem Erdbeben in Italien unbemannte Landfahrzeuge unterstützend genutzt worden [MKP+08, LTB+10, KTL+12].

  Nach dem Flugzeugabsturz des Air France Fluges 447 von Rio de Janeiro nach Paris im Juni 2009 sind erstmals autonome Unterwasserzeuge eingesetzt worden, um das Wrack im Atlantik zu orten. Es sind während der erfolgreichen, zweijährigen Suche über 2000 Quadratkilometer vor der brasilianischen Küste abgedeckt und kartiert worden. Hierbei kamen drei Unterwasserfahrzeuge zum Einsatz. Es sind fortlaufend die Routen untereinander abgestimmt und Seegebiete zugewiesen worden [Sch11, Wil11, Wis11]. Um die Einsatzzeiten zu erhöhen, können Unterwassergleiter eingesetzt werden, die den Auftrieb nutzen, um sich fortzubewegen. Hierfür wird abwechselnd der Bug mit Wasser und Luft gefüllt [RDE+04]. Nach dem Absturz der Air Malaysia Maschine MH370 ist 2014 ein unbemanntes Unterwasserfahrzeug eingesetzt worden und hat mehr als 850 Quadratkilometer abgedeckt [Mal14, Har14].

  Erstmals sind unbemannte Luftfahrzeuge nach dem Hurrikan Katrina in Louisiana im Jahr 2005 eingesetzt worden, um nach Menschen in Not zu suchen. Mit Hilfe einer Kamera konnten Häuser abgesucht werden. Vor allem kann aufgrund der Agentengröße ein flexibler und mobiler Einsatz gewährleistet werden. Die Luftfahrzeuge können nahezu überall landen und auch enge Passagen abfliegen [Rod05, MY07]. In den Jahren 2010 und 2011 haben sich Erdbeben in Haiti und Japan ereignet, bei denen ebenfalls umbenannte Luftfahrzeuge im Rahmen von Such- und Rettungsaufgaben erfolgreich eingesetzt worden sind [Gui11].

- **Überwachung:**

  Anwendungen im Bereich der Überwachung sind denen der Aufklärung ähnlich. Ein wesentlicher Unterschied besteht darin, dass eine Systemumwelt mehrfach, wiederholt abgedeckt wird, um Veränderungen der Systemumwelt wahrzunehmen oder selbst Änderungen vorzunehmen. Die Systemumwelt kann zu Beginn der kontinuierlichen Abdeckung unbekannt sein. Nach der erstmaligen Abdeckung können optimierte Routen für die Agenten ermittelt werden. Ein wesentliches Optimierungsziel stellt die gleichmäßige Abdeckung dar. Hierbei werden die Teilregionen in möglichst gleichen Zeitintervallen oder einem anderen definierten Muster wiederholt abgedeckt. Die Agenten sollten über Navigationssensoren des Fernbereiches verfügen, um ein möglichst großes Gebiet abdecken zu können. Des Weiteren sollte der Energieverbrauch möglichst gering sein, um eine möglichst lange Einsatzdauer ohne Unterbrechungen gewährleisten zu können.

  Im Jahr 2010 explodierte die Ölbohrinsel Deepwater Horizon im Golf von Mexiko und zirka vier Millionen Barrel Öl sind ausgetreten. Die resultierende Ölpest hat Teile der Meeresfauna und -flora geschädigt. Die Fischerei- und Tourismuswirtschaft hat zum Teil immer noch unter den negativen Auswirkungen der Ölpest zu leiden [Goe14]. Zur Überwachung großer Seegebiete sind autonome Unterwassergleiter eingesetzt worden, um Messungen der Wasserqualität vorzunehmen. Die Unterwassergleiter sind im Sinne eines Frühwarnsystems eingesetzt worden, um Veränderungen vor der Küste Floridas zu dokumentieren. Bis zu vier Unterwassergleiter patrouillierten im Zeitraum von Mai bis Oktober 2010. Die gesammelten Informationen ermöglichen das Aufstellen von neuen Modellen, die eine Ausbreitung von Stoffen im Meer prognostizieren können [Mah12].

  In Europa gibt es zirka 200.000 Km Gaspipelines, welche durch Patrouillen mit Luft- und Landfahrzeugen abgedeckt und beobachtet werden. Das Mineralölunternehmen TOTAL S.A. entwickelt und nutzt autonome Unterwasserfahrzeuge für die Überwachung von Pipelines. Es können Beschädigungen und Druckveränderungen somit frühzeitig entdeckt werden. Die Unterwasserfahrzeuge sollen durch Unterwasserstationen mit Energie und Umweltinformationen versorgt werden. Folglich soll es den Unterwasserfahrzeugen möglich sein bis zu drei Monate agieren zu können [TOT10].

  Ein neues Anwendungsgebiet für autonome Landfahrzeuge stellt das Aufspüren und Beobachten von Tieren dar, um beispielsweise eine Populationsgröße zu schätzen. Diese Aufgabe kann mehrere Tage oder Monate in Anspruch nehmen, bis ein bestimmtes Tier oder Verhaltensweisen von Tieren aufgezeichnet werden können. Die Anwesenheit von Menschen kann, durch den verursachten Lärm oder Duft, das natürliche Verhalten der Tiere beeinflussen. Autonome Agenten hingegen können kontinuierlich und systematisch große Gebiete absuchen. Die Agenten selbst werden von den Tieren nicht als Lebewesen wahrgenommen [Tew13].

Die verschiedenen Anwendungsfelder weisen Gemeinsamkeiten auf und stellen hohe Anforderungen an die autonomen und unbemannten Agenten, welche in der Abbildung 2.1 veranschaulicht sind. In der Regel verfolgen die Agenten das übergeordnete Ziel, eine Systemumwelt möglichst schnell und vollständig abzudecken, um beispielsweise bestimmte Objekte aufzufinden oder Aktionen auszuführen. Zusätzlich wird das Ziel verfolgt, dass die Agenten eine möglichst lange Betriebsdauer aufweisen.

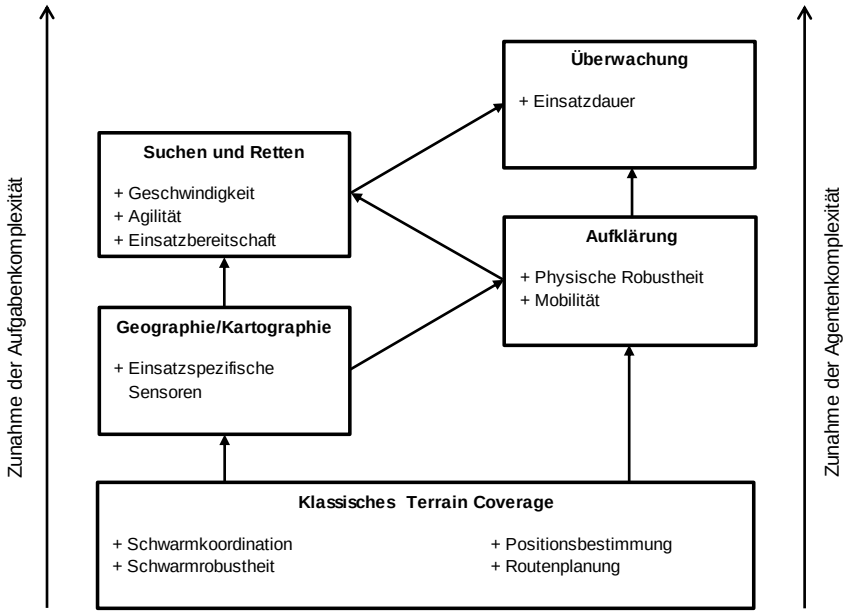


Abbildung 2.1.: Anwendungsfelder des Terrain Covering und die spezifischen Anforderungen an die Agenten.

Die Betriebsdauer kann beispielsweise durch die Auswahl geeigneter Routen oder alternativer Antriebstechnologien beeinflusst werden.

Bezüglich des Einsatzes von Agenten in realen Anwendungen zeigt sich, dass die zahlreichen Herausforderungen der Systemumwelt die derzeitigen Möglichkeiten und Fähigkeiten der Agenten einschränken [CM03b, CM05, MKP+08, Mur14]. Aufgrund von extremen, äußeren Einflüssen, wie Druck, Hitze oder Strahlung, können Störungen und Schäden an den Agenten auftreten [Sha12]. Vor allem bei Landfahrzeugen unterliegen die Effektoren für die Fortbewegung starken Beanspruchungen, infolge von Verunreinigungen oder Hindernissen [Pis00]. Des Weiteren kann die Systemumwelt aufgrund von Hindernissen stark unstrukturiert sein und Befahrung sowie Aufklärung erschweren. Hinsichtlich der Sensoren verursachen Hindernisse, Wetterbedingungen und eine unzureichende Ausleuchtung der Systemumwelt beispielsweise Störungen an den optischen Kameras. Resultierend kann eine eingeschränkte Navigation und Identifikation von Objekten vorliegen. Bei der Kommunikation treten insbesondere für die drahtlose Übermittlung eine Vielzahl von Störungen auf. Zum Beispiel können Unterbrechungen bei der Übermittlung auftreten. Im Falle einer gestörten Kommunikation müssen die Agenten in der Lage sein, selbständig geeignete Entscheidungen zu treffen. Störungen an dem internen Kontrollsystem der Agenten können unter anderem zu einer fehlerhaften Fahrtsteuerung und fehlerhaften Zustandsübergängen führen [CM03a]. Ferner kann eine fehlerhafte Grundkonfiguration, hinsichtlich der Auswahl von Sensoren und Effektoren sowie eine fehlerhafte Definition der Kontrollparameterkonfiguration, einen negativen Einfluss auf die Zielerreichung nehmen [Sha12]. Abschließend ist zu erwähnen, dass bereits beim Transport der Agenten verschiedene Schäden an der sensiblen Technik verursacht werden können.

Resultierend werden an die Agenten hohe Anforderungen gestellt. Die Agenten müssen sich robust gegenüber äußeren Einflüssen verhalten können. Hierbei können Sensoren dazu dienen, Gefahren und Änderungen der Systemumwelt frühzeitig wahrzunehmen, um ein sicheres sowie autonomes

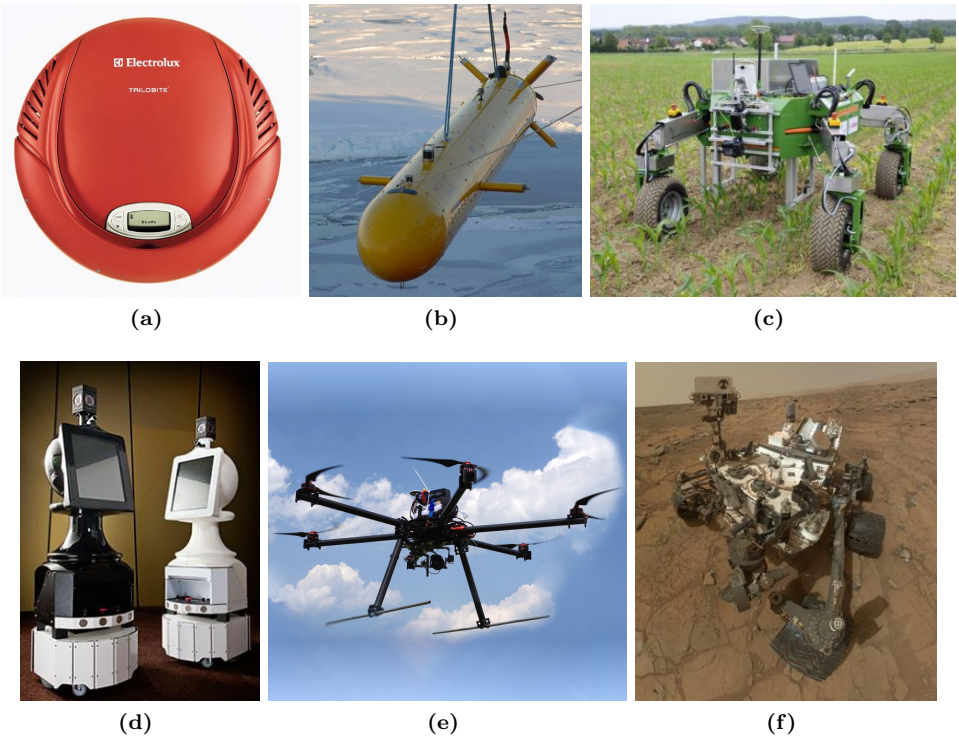


Abbildung 2.2.: Unbemannte und autonome Agenten für Anwendungsfelder des Terrain Covering. Unter (a) Reinigungsroboter Trilobite 2001 [Ele10], (b) Autonomes Unterwasserfahrzeug bei einer Arktisexpedition 2011 [HR12], (c) Landwirtschaftsroboter BoniRob 2012 [BWDA12], (d) Museumsroboter Chesster und Kasparov 2013 [The13], (e) Unbemanntes Luftfahrzeug HERO 2013 [Bri15], (f) Mars Rover Curiosity 2013 [Jet13].

und optimiertes Navigieren zu ermöglichen. Ferner hat sich beim Einsatz von Agenten gezeigt, dass ein flexibler Austausch von Sensoren möglich sein sollte, um Anpassungen gemäß der Systemumwelt vornehmen zu können [CM03b]. Ferner werden Sensoren benötigt, die eine kabellose Kommunikation zwischen den Agenten und Bediener ermöglichen. Im Sinne der Anwendungen, ist ein untereinander abgestimmtes und koordiniertes Verhalten der Agenten notwendig. Zusätzlich sollten die Effektoren für die Fortbewegung den Eigenschaften der Systemumwelt und Anwendung angepasst sein. In der Abbildung 2.2 sind verschiedene Agenten möglicher Anwendungsfelder des TCP abgebildet.

In dieser Dissertation werden Systemumgebungen betrachtet für die $\Omega \subset \mathbb{R}^2$ gilt. Grundsätzlich kann das online und offline TCP unterschieden werden. Für das online TCP liegen zu Beginn der Abdeckung keine Informationen bezüglich der Systemumgebung vor. Im Verlauf der vollständigen Abdeckung werden Informationen über die Systemumgebung gesammelt, welche für eine optimierte Koordination und Routenplanung von Agenten genutzt werden können. Hingegen verfügen die Agenten bei einem offline TCP über die Gesamtheit aller benötigten Umgebungsinformationen. Aus diesem Grund können bereits zu Beginn der Abdeckung optimale Routen ermittelt werden.

Für eine optimierte Abdeckung können Agenten miteinander in Interaktion treten und ein MAS bilden, welches wie folgt definiert wird [DL89, Syc98].

Definition 2.5 (Multi Agent System)**.** : *Ein MAS ist ein lose gekoppeltes Netzwerk von interagierenden Agenten, die eine Problemlösung herbeiführen, welche außerhalb der eigenen Fähigkeiten oder dem individuellen Wissensstand liegen kann.*

Ferner bedarf es einer Definition der Systemumwelt.

Definition 2.6 (Systemumwelt)**.** : *Die Systemumwelt* $\Xi = \{\Omega, \Psi\}$ *beschreibt die Systemumgebung* Ω*, einschließlich allgemeiner Eigenschaften* Ψ*. Die Gesamtheit aller Eigenschaften* Ψ *umfasst beispielsweise die Ausprägung der räumlichen Systemgrenzen sowie den Einfluss auf die Sensoren und Effektoren. Neben der realen Systemumwelt ist die Repräsentation beziehungsweise das Abbild der Systemumwelt* $\Xi' = \{\Omega', \Psi'\}$ *zu unterscheiden, welche von den Agenten erstellt und genutzt wird. In der Regel stellt die Repräsentation* Ξ' *ein vereinfachtes Abbild der realen Systemumwelt dar und wird mit Hilfe einer Zerlegung in Zellen vereinfacht.*

Basierend auf der Sensorreichweite der Agenten wird die Systemumwelt gewöhnlich in Zellen unterteilt, welche mindestens einmal besucht werden. Eine Zelle wird wie folgt definiert.

Definition 2.7 (Zelle)**.** : *Eine Zelle* $z_i \in \Xi'$ *ist eine durch den Agenten festgelegte Teilregion der Systemumwelt mit* $\Xi' \subseteq \cup z_i$*. Die räumliche Ausdehnung und Form einer Zelle bestimmt sich durch verschiedene Faktoren, wie die Sensorreichweite oder Abdeckungsstrategie.*

Allgemein kann eine Zelle mindestens zwei Zustände annehmen.

Definition 2.8 (Blockierte Zelle)**.** : *Eine Zelle* z_i *gilt als blockiert, wenn sie teilweise oder vollständig durch ein Hindernis belegt ist.*

Definition 2.9 (Freie Zelle)**.** : *Eine Zelle* z_i *ist genau dann frei, wenn sie vollständig abgedeckt werden kann.*

Eine freie Zelle gilt gemäß der nachfolgenden Definition als abgedeckt.

Definition 2.10 (Abdeckung)**.** : *Eine Zelle* z_i *ist abgedeckt, genau dann wenn ein Agent sie besucht und kartographiert hat, beziehungsweise es gilt:* $\exists z_i \subset \Xi, a_k \in A : z_i \subseteq B_{a_k}(Pos)$*.*

Die Qualität der Gesamtabdeckung ist stark abhängig von den geplanten Routen der Agenten. Eine Route wird mittels des Konzepts der Nachbarschaft auf der Menge der Zellen bestimmt.

Definition 2.11 (Nachbarschaft)**.** : *Eine Nachbarschaft* $\Phi_a(z_i)$ *für eine Zelle* z_i *definiert sich durch die Menge* $[z_{i+1}, \ldots, z_{i+n}]$ *aller mit* z_i *direkt verbundenen Zellen, zum Beispiel über eine gemeinsame Zellgrenze.*

Definition 2.12 (Route)**.** : *Eine Route enthält mindestens zwei benachbarte Zellen* z_i*,* z_{i+1}*. Alle Zellen* $[z_i, z_{i+1}, \ldots, z_{i+n}]$ *einer Route sind über die Nachbarschaftsbeziehungen direkt (aufeinanderfolgende Zellen* z_i*,* z_{i+1}*) oder indirekt über Zellen miteinander verbunden.*

Die Routenplanung verfolgt unterschiedliche Ziele wie beispielsweise die Routenlänge, die benötigte Zeit und den Energieverbrauch. In dieser Dissertation werden die Kosten entsprechend der nachfolgenden Definition verstanden.

Definition 2.13 (Kosten)**.** : *Die Kosten geben den Energieverbrauch eines Agenten an, der aufgebracht werden muss, um eine Route vollständig abzudecken.*

Im nachfolgenden Abschnitt wird eine formale Beschreibung des TCP vorgenommen. Hierzu werden ausgehend von den genannten Anwendungsbeispielen das Multiple Traveling Salesman Problem (MTSP) und das Multiple Covering Salesman Problem (MCSP) eingeführt, um geeignete Routen für die Agenten ermitteln zu können. Für beide Probleme wird das Ziel verfolgt die Kosten der Gesamtroute, bestehend aus den individuellen Routen der Agenten, für eine vollständige Abdeckung der Systemumwelt zu minimieren. Die Optimierung der Kosten wirkt sich positiv auf die Effizienz und dem Einsatz sowie folglich auch auf die Akzeptanz autonomer Agenten aus.

2.2. Eine formale Beschreibung des Terrain Covering Problems

Das Terrain Covering Problem kann mit der Problemklasse der TSP formal beschrieben werden. Das TSP wurde erstmals im 18. Jahrhundert von Hamilton und Kirkman untersucht. Demnach sei ein gerichteter Graph $G = (V, E)$ mit positiven Kosten c_{ij} gegeben. Die Kostenmatrix C kann bezüglich der Kanten $(i, j) \in E$ oder der Knoten V definiert werden. Gilt $(i, j) \notin E$, dann existiert keine Kante zwischen den Knoten i und j und die Kosten betragen $c_{ij} = \infty$. Für das TSP wird eine Tour mit der Reihenfolge aller zu besuchenden Knoten in V gesucht, so dass die Kosten über alle genutzten Kanten c'_{ij} minimiert werden. Die resultierende Tour kann zum Startknoten zurückführen [BN68]. Das zugehörige Minimierungsproblem generiert genau einen Pfad für einen Agenten. Dieser spezielle Pfad wird als Hamiltonkreis bezeichnet. Wird die Restriktion der Rückkehr zum Startknoten nicht verlangt, genügt die Lösung des Hamiltonpfades. Das TSP gehört zu der Komplexitätsklasse der $\mathcal{NP}$-vollständigen Entscheidungsproblemen[3]. Dazu wird in [Nar89] der Beweis geführt, dass das Hamiltonkreisproblem für einen gerichteten Graphen $\mathcal{NP}$-vollständig ist. Ferner kann das TSP auf das Hamiltonkreisproblem reduziert werden, genau dann wenn die Kantenkosten konstant sind [Pap77].

Das TSP beziehungsweise das generalisierte MTSP und MCSP beschreiben formal das Problem der Routenplanung für mehrere Agenten unter bestimmten Nebenbedingungen. Mit Hilfe dieser Probleme kann die Minimierung redundanter Besuche von Regionen in einer Systemumwelt untersucht werden. Des Weiteren können unterschiedliche Kostenfunktionen betrachtet werden, welche einer Minimierung unterliegen. Die Kosten können beispielsweise die Routenlänge, den Zeit- und Energiebedarf sowie Eigenschaften einer Bedrohungslage oder Befahrbarkeit repräsentieren. Im Folgendem wird formal das MTSP und das MCSP erläutert und wie diese für eine effiziente Routenplanung, im Sinne der Kostenminimierung, eingesetzt werden können.

[3] Ein Entscheidungsproblem $\mathcal{P}$ heißt $\mathcal{NP}$-vollständig, wenn $\mathcal{P}$ zur Klasse $\mathcal{NP}$ gehört und für jedes Problem $K \in \mathcal{NP}$ gilt, K ist polynomial reduzierbar auf $\mathcal{P}$.

2.2.1. Das Multiple Traveling Salesman Problem

Für das generalisierte MTSP besteht die Annahme, dass mehrere Agenten $a \in A$ für die Abdeckung aller Knoten zur Verfügung stehen. Die Agenten starten von einem gemeinsamen Startknoten aus V. In Anlehnung an das Vehicle Routing Problem wird dieser Startknoten als Single Depot bezeichnet. Die einzelnen Pfade enden ebenfalls wieder am Startknoten und alle Knoten werden mindestens durch einen Agenten aus A besucht. Der Problemstellung entsprechend können Variationen des MTSP vorgenommen werden. Zum einen können für die Agenten unterschiedliche Startknoten gegeben sein. Zum anderen kann die Anzahl der Agenten in A variabel nach oben begrenzt sein. Des Weiteren gibt es die Klasse der zeitabhängigen MTSP, bei denen Zeitfenster für Knoten definiert sind, in denen der Besuch erfolgen muss. Das MTSP wird als binäres Minimierungsproblem für $G' = (V, E')$ in [Bek06] wie folgt formuliert:

$$\min Z = \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \tag{2.1}$$

Unter den Nebenbindungen:

$$\sum_{j=2}^{|V|} x_{1j} = |A|, \tag{2.2}$$

$$\sum_{j=2}^{|V|} x_{j1} = |A|, \tag{2.3}$$

$$\sum_{i=1}^{|V|} x_{ij} > 0, \quad j = 2, ..., |V| \tag{2.4}$$

$$\sum_{j=1}^{|V|} x_{ij} > 0, \quad i = 2, ..., |V| \tag{2.5}$$

$$\text{Subtour-Eliminierungsbedingung} \tag{2.6}$$

$$x_{ij} \in \{0, 1\}, \forall (i, j) \in E, \tag{2.7}$$

mit

$$x_{ij} = \begin{cases} 1, & \text{wenn Kante (ij) auf der Multiple Traveling Salesman Tour liegt} \\ 0, & \text{sonst} \end{cases}$$

Die in (2.1) angegebene Zielfunktion beschreibt die Minimierung der Gesamtkosten aller Touren. Die Nebenbedingungen legen Beschränkungen hinsichtlich des Lösungsraumes fest. Diesbezüglich stellen die Nebenbedingungen (2.2) und (2.3) sicher, dass alle m Touren zum Startknoten zurückführen. Die Nebenbedingung (2.3) wird durch (2.2), (2.4) und (2.5) bereits impliziert und wird zur Vollständigkeit aufgeführt. Aus dem TSP können die Nebenbedingungen (2.4), (2.5) und (2.7) übernommen werden. Dass alle Knoten mindestens einmal besucht werden, wird durch die Nebenbedingungen (2.4) und (2.5) gewährleistet. Die letzte Nebenbedingung (2.7) erzwingt die binäre Entscheidung für die Hinzunahme oder das Weglassen einer Kante für die Tour. Mit der Nebenbedingung (2.6) wird verhindert, dass zwischen Knoten generierte Subtouren in die Lösung aufgenom-

men werden. Subtouren stellen Pfade dar, die nicht mit der Multiple Travelling Salesman Tour über Kanten verbunden sind. In der Literatur werden verschiedene Subtour-Eliminierungsbedingungen dargestellt, welche ebenfalls Anwendung für das TSP finden. In [DFJ54] wird eine mögliche Eliminierungsbedingung aufgestellt:

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq v_1 - 1, \qquad \forall S \subseteq V \backslash \{1\}, S \neq \emptyset \tag{2.8}$$

Angenommen eine Subtour enthält $v_1 < |V|$ Knoten. Damit keine Schleife vorliegt, muss die Summe über x_{ij} für alle Kanten (ij) kleiner gleich $v_1 - 1$ betragen. Durch die Formulierung der Bedingung verhält sich die Rechenzeit jedoch exponentiell bezüglich einer zunehmenden Anzahl von Knoten.

Zum Lösen des $\mathcal{NP}$-vollständigen TSP und MTSP werden Metaheuristiken genutzt. Zu nennen sind hier Evolutionäre Algorithmen, Simulated Annealing, neuronale Netze, Ameisenalgorithmen und Tabu Search [XLF08]. Häufig werden Knoten vorher mittels einer Distanzmetrik geclustert. Zum Einsatz kommt unter anderem das k-means Clusteringverfahren [NDDP09]. Exakte Lösungen können mit dem Branch and Cut Verfahren ermittelt werden [GS86, ABC$^+$09].

2.2.2. Das Multiple Covering Salesman Problem

Das Covering Salesman Problem (CSP) ist erstmals von Current im Jahr 1981 beschrieben und untersucht worden [Cur81]. In [CS89] wird ein anschauliches Beispiel formuliert: In einer ländlichen Region sind mehrere mobile Ärzteteams für die medizinische Versorgung der Bevölkerung zuständig. Um die Einsatzdauer der Ärzteteams zu minimieren und eine möglichste gute Abdeckung/Versorgung der Bevölkerung zu gewährleisten, müssen geeignete Touren ermittelt werden. Auf den Touren sollen Zwischenstops eingelegt werden, zu denen die Bevölkerung kommen kann. Das Ziel, des zugrundeliegenden Optimierungsproblems, liegt darin, möglichst kurze/schnelle Touren für die Ärzteteams zu ermitteln. Des Weiteren müssen die Zwischenstops derart gewählt sein, dass jeder Einwohner im Mittel möglichst nah an einem der Zwischenstops wohnt.

Trotz weiterer anwendungsnaher Problemstellungen ist dem CSP sowie dem MCSP in der Literatur eine untergeordnete Rolle zugeschrieben. Das MCSP stellt unter anderem eine Generalisierung des MTSP sowie CSP dar. Im Allgemeinen liegt jeder Knoten $v_i \in V$, der nicht Bestandteil einer der Touren ist, in einer vordefinierten Abdeckungsdistanz d^* von mindestens einem Knoten einer der Touren. Die Abdeckungsdistanz kann beispielsweise durch den Wirkungsgrad der Sensoren repräsentiert werden. Demnach steigt mit zunehmender Sensorreichweite der Agenten $a \in A$ die Wahrscheinlichkeit, dass weniger Knoten direkt besucht werden müssen. Das MCSP lässt sich auf das MTSP reduzieren, genau dann wenn $d^* < \min_{i,j} d_{i,j}$ vorliegt. Hierbei definiert $d_{i,j}$ die kürzeste Distanz zwischen allen Knoten i und j. Für diesen Fall muss wie im MTSP jeder Knoten mindestens einmal besucht werden, um gleichzeitig auch abgedeckt zu sein [CS89, GNR$^+$12].

Dementsprechend kann das MCSP, angelehnt an [Bek06, GNR+12], wie folgt formuliert werden:

$$\min Z = \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \tag{2.9}$$

Unter den Nebenbindungen:

$$\sum_{j=2}^{|V|} x_{1j} = |A|, \tag{2.10}$$

$$\sum_{j=2}^{|V|} x_{j1} = |A|, \tag{2.11}$$

$$\sum_{i=1}^{|V|} x_{ij} = w_j, \qquad j = 2, ..., |V| \tag{2.12}$$

$$\sum_{j=1}^{|V|} x_{ij} = w_i, \qquad i = 2, ..., |V| \tag{2.13}$$

$$\sum_{j=1}^{|V|} b_{ij} w_j \geq 1, \qquad \forall i \in \{i | i \in V, w_i = 0\} \tag{2.14}$$

$$\text{Subtour-Eliminierungsbedingung} \tag{2.15}$$

$$x_{ij} \in \{0, 1\}, \forall (i, j) \in E, \tag{2.16}$$

$$w_i \in \{0, 1\}, \forall i \in V, \tag{2.17}$$

mit

$$x_{ij} = \begin{cases} 1, & \text{wenn Kante } \textit{(ij)} \text{ auf der Multiple Covering Salesman Tour liegt} \\ 0, & \text{sonst} \end{cases}$$

$$w_i = \begin{cases} 1, & \text{wenn Knoten } i \text{ konkret besucht wird} \\ 0, & \text{sonst} \end{cases}$$

$$b_{ij} = \begin{cases} 1, & \text{wenn Knoten } i \text{ ausgehend von } j \text{ abgedeckt wird } (d^* \geq d_{i,j}) \\ 0, & \text{sonst} \end{cases}$$

Bezüglich dem MTSP werden die Nebenbedingung (2.12) und (2.13) derart angepasst, dass besuchte Knoten genau zu einer Route gehören. Ein redundanter Besuch wird ausgeschlossen. Die Nebenbindung (2.14) stellt sicher, dass jeder Knoten, der nicht Bestandteil einer Route ist, durch mindestens einen besuchten Knoten abgedeckt wird.

Die angeführte Definition des MCSP ist auf das statische offline Problem beschränkt. Es müssen alle Informationen über die Systemumwelt, einschließlich der Kostenmatrix, vorliegen. Änderungen, im Sinne einer dynamischen Umwelt, können nicht im Verlauf der Abdeckung berücksichtigt werden. Zudem kann auf Störungen der Agenten, hinsichtlich der Sensoren und Bewegung, nicht dynamisch reagiert werden. Hierbei können kleine Umweltänderungen und Störungen zu er-

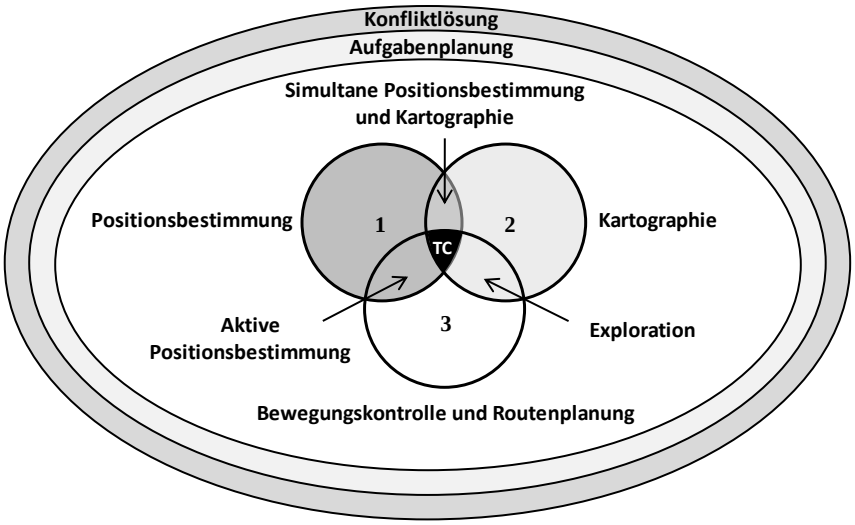


Abbildung 2.3.: Teilprobleme des Terrain Covering (erweitert und angelehnt an [GHA+08]).

heblichen Abweichungen, bezüglich der zuvor ermittelten optimierten Lösung, führen. Aus diesen Gründen ist die Lösung des dynamischen online MCSP komplexer und bedarf des Einsatzes geeigneter Methoden der Entscheidungsunterstützung unter Unsicherheiten. In den Kapiteln 4 und 5 werden Verfahren der robusten Routenplanung, Aufgabenplanung und Konfliktlösung speziell für ein MARS vorgestellt und überprüft, die eine effiziente und robuste Lösung des online MCSP und MTSP ermöglichen.

Neben der Bestimmung geeigneter Routen müssen die autonomen Agenten eine Vielzahl von weiteren Aufgaben durchführen. Im nachfolgenden Abschnitt wird eine Abgrenzung des TCP im Rahmen von realen Anwendungen vorgenommen. Hierbei wird aufgezeigt, dass das TCP eine Schnittmenge von Teilproblemen der Positionsbestimmung, Kartographie und Bewegungskontrolle sowie Routenplanung ist. Zusätzlich werden die Bedeutung der Aufgabenplanung und Konfliktlösung für Anwendungen des TCP eingeordnet und erörtert.

2.3. Eine Abgrenzung des Terrain Covering Problems

Aufbauend auf der formalen Beschreibung des Terrain Covering Problems wird eine weiterführende Abgrenzung verschiedener Teilprobleme angeführt. Die Abgrenzung der Teilprobleme ist notwendig, weil neben der Routenplanung sowie Bewegungssteuerung die Positionsbestimmung und die Kartographie für eine Modellierung von autonomen, mobilen Robotern/Agenten und deren Verhaltensweisen betrachtet werden müssen. Für alle drei Teilprobleme einschließlich des TCP sollten für effiziente und robuste Implementierungen zwei weitere Teilprobleme die Aufgabenplanung und Konfliktlösung betrachtet werden. In der Abbildung 2.3 werden die nachfolgend beschriebenen Zusammenhänge dargestellt.

- **Positionsbestimmung:**
  Die Positionsbestimmung ist ein fundamentales Problem, um Agenten mit autonomen Fähigkeiten und Verhaltensweisen ausprägen zu können. Basierend auf der absoluten beziehungsweise relativen Position in der Systemumwelt ist es den Agenten möglich, selbstständig Entscheidungen, bezüglich den nachfolgenden Aktionen, treffen zu können [Cox91]. Es werden zwei Klassen von Positionsbestimmungen unterschieden. Die relative Positionsbestimmung beruht auf der Odometrie und dem inertialen Navigationssystem, welche Sensoren nutzen, um beispielsweise die zurückgelegte Strecke und Beschleunigungs- sowie Neigungsänderungen zu messen. Weiterhin kann die relative Positionsbestimmung unabhängig von Informationen der Systemumwelt eingesetzt werden. Die Klasse der absoluten Positionsbestimmung basiert auf Referenzsystemen wie dem Global Positioning System, Modellabgleich, Magnetkompass sowie Signalstationen und Landmarken [BEFW97].

- **Kartographie:**
  Das Mapping Problem umfasst die Erstellung einer räumlichen Abbildung der Systemumwelt. Eine Abbildung der Systemumwelt kann zur Navigation und insbesondere für die Positionsbestimmung genutzt werden. Die Agenten benötigen verschiedene Sensoren, um eine Systemumwelt wahrnehmen zu können. Die Sensoren unterliegen häufig Reichweitenbeschränkungen. Aus diesem Grund ist es notwendig, dass sich die Agenten in der Systemumwelt bewegen und orientieren können. Aus den Agentenbewegungen können Informationen, bezüglich der Systemumwelt, abgeleitet werden. Die Odometrie unterliegt hierbei einer Vielzahl von negativen Messeinflüssen und dies kann zu Positionsabweichungen führen. Auch wenn die Messeinflüsse, wie Bodenbeschaffenheit oder Radgeometrie minimiert werden, resultieren Messfehler, die sich mit zunehmender Zeit kumulieren. Die Ursache ist die statistische Abhängigkeit der Messfehler bezogen auf die zurückgelegte Strecke [Thr03]. Aus diesem Grund sollten weitere Verfahren der Positionsbestimmung, wie beispielsweise Peilungen im Raum oder das Global Positioning System eingesetzt werden.

- **Bewegungskontrolle und Routenplanung:**
  Um die negativen Auswirkungen der Messfehler kontrollieren zu können, stellt das Teilproblem der Bewegungskontrolle und Routenplanung den Prozess der Bestimmung und des Befahrens einer geeigneten Route zwischen zwei Punkten dar. Die Bestimmung der Route kann unterschiedlichen Restriktionen unterliegen. Es können drei Ansätze der Routenplanung unterschieden werden, die das Problem der Bewegungs- und Routenplanung auf eine diskrete Graphensuche reduzieren. Dementsprechend können etablierte Heuristiken [Dij59, Flo62, HNR68, Dor92, GCW97] für die Ermittlung einer geeigneten Route genutzt werden. Für den Ansatz der Zellzerlegung wird die Systemumwelt in Zellen zerlegt. Die Zellen entsprechen den Knoten im Graphen und die Kanten definieren die direkten Übergänge zwischen den Zellen [HZ91, LC98]. Mit Hilfe der Einführung der Potentialfeldmethode führt der Agent einen Gradientenabstieg zum anziehenden Potentialfeld, dass heißt zum Ziel der Route, durch. Hindernisse führen zu einer Abstoßung der Agenten [Kha86]. Der Stichprobenansatz generiert zufallsbasiert geeignete lokale Teilrouten. Die Gesamtroute wird durch das Zusammensetzen der Teilrouten generiert [KSLO94, YJA13]. Ist eine Route bestimmt, muss diese mit einer möglichst geringen Abweichung befahren werden. Die fortlaufende absolute und relative Positionsbestimmung der Agenten nimmt hierbei einen hohen Stellenwert ein. Sie dient gleichzeitig der Positionskontrolle sowie den notwendigen Korrekturen [Can88]. Aufgrund der

Freiheitsgrade und Beschränkungen, in bis zu zwölf Dimensionen, existieren keine exakten Algorithmen für die Bewegungskontrolle von autonomen Agenten in Echtzeit. Mit Hilfe von Heuristiken, welche auf optimalen Lösungen kleinerer Teilprobleme basieren, ist eine Positionsschätzung und resultierende Positionskontrolle möglich [BF96, GKM10].

Die drei wesentlichen Teilprobleme kommen in unterschiedlichen Anwendungen zum Tragen. Für Anwendungen, bei denen die Agenten in einer unbekannten Systemumwelt operieren, müssen die Verfahren der simultanen Lokalisierung und Kartenerstellung, die Position des Agenten in der Systemumwelt sowie die eigentliche Repräsentation der Systemumwelt zugleich schätzen. Aufgrund der Tatsache, dass weder ein Abbild der Systemumwelt noch die absolute Position der Agenten bekannt ist, wird das Abbild der Systemumwelt inkrementell aufgebaut. Die Startposition des Agenten wird dem Koordinatenursprung gleichgesetzt. Aufgrund von eindeutigen Navigationsmarken in der Systemumwelt sowie den Grundlagen der Kinematik können neue Regionen in der Abbildung eingeordnet und die Bewegung des Agenten bestimmt werden [But00, BRH00, CN01]. Um möglichst effizient ein Abbild der Systemumwelt erstellen zu können, muss die Exploration beziehungsweise Erkundung der Systemumwelt koordiniert erfolgen. Hierbei müssen Entscheidungen bezüglich der Erkundungsreihenfolge und Zuweisung unbekannter Regionen, beziehungsweise allgemein für die Aufgabenreihenfolge, getroffen werden [GM04b]. Resultierend werden optimierte Routen hinsichtlich der Performancekriterien ermittelt, um die Aufgaben bearbeiten zu können.

Um eine optimierte Implementierung der drei Teilprobleme hinsichtlich der Effizienz und Robustheit ermöglichen zu können, müssen zusätzlich die Probleme der Aufgabenplanung und Konfliktlösung betrachtet werden.

- **Aufgabenplanung:**
  Die Aufgabenplanung untergliedert sich in die Probleme der Aufgabenzerlegung und der Aufgabenverteilung. Das Problem der Aufgabenzerlegung untersucht die optimierte Unterteilung des TCP in Teilaufgaben, sodass die Gesamtperformance verbessert wird. Die Unterteilung ist abhängig von den Abhängigkeiten der Aufgaben untereinander, den Ausprägungen der Agenten, der Ressourcenverteilungen, der Systemumwelt und den Zielen des TCP selbst [CYW10]. Das Problem der Aufgabenverteilung stellt eine Instanz des Zuweisungsproblems dar. Hierbei erfolgt eine optimierte Zuteilung der Teilaufgaben an die Agenten derart, dass der Gesamtnutzen maximiert wird. Des Weiteren kann die Zuteilung mit dem Ziel verfolgt werden, dass die Gesamtkosten der Aufgabenbearbeitung oder die Performance optimiert wird. Für die Aufgabenzuteilung können zwei Ansätze unterschieden werden. Zum einen können alle Aufgaben zu Beginn offline verteilt werden. Zum anderen kann eine online Verteilung zur Laufzeit organisiert werden. In beiden Ansätzen können Konflikte bei der Zuweisung von Aufgaben auftreten. Zusammenfassend kann insbesondere die Effizienz der Bewegungssteuerung und Routenplanung gesteigert werden. Zudem wird in der Regel das Aufgabenplanungsproblem der Klasse der $\mathcal{NP}$-schweren Probleme zugeordnet [YJA13]. Ein etabliertes Verfahren der Aufgabenzuweisung stellt das Contract Net Protokoll dar [SDS78].

- **Konfliktlösung:**
  Für die zuvor erwähnten Teilprobleme müssen geeignete Verfahren der Konfliktlösung implementiert werden. Eine Konfliktsituation kennzeichnet sich durch inkompatible oder sich ausschließende Ziele und Parameter. Es resultiert eine Situation von unvereinbaren Kräf-

ten, die von äußeren Bedingungen oder den Entitäten selbst verursacht werden [TCM01]. Es werden allgemein drei Konfliktklassen unterschieden [BLR01]. Hierbei können Zielkonflikte aufgrund von unterschiedlichen Zielausprägungen auftreten. Insbesondere wenn mehrere Ziele parallel betrachtet werden müssen, können die Konflikte mit Hilfe der Einführung von Prioritäten oder der Modifikation der Zielausprägungen gelöst werden. Planungskonflikte liegen genau dann vor, wenn sich die Bedingungen im Verlauf der geplanten Aktionen ändern. Resultierend können nachfolgende Aktionen ineffizient oder nicht ausführbar sein. Die Klasse der Meinungskonflikte umfasst Konflikte bezüglich von inkonsistenten Umständen sowie Begebenheiten oder widersprüchlichen Bewertungen. Meinungen umfassen die Erkenntnisse eines Agenten bezüglich des definierten Zuständigkeitsbereichs. Hierbei werden die Meinungen durch die Agenten entweder aus einer globalen Menge entnommen oder die Agenten tauschen die unterschiedlichen Meinungen untereinander aus. Des Weiteren bestehen die Möglichkeiten, dass die individuellen Meinungen auf Grundlage der ausgeführten Aktionen oder aus der unmittelbaren Wahrnehmung der Systemumwelt extrahiert und gebildet werden [BM97]. Meinungskonflikte können aufgrund von unterschiedlichen Strukturbeschreibungen, Abstraktionsstufen oder Informationsständen vorliegen.

Ähnlich der Störungslösung [Fer94] untergliedert sich die Konfliktlösung in die Probleme der Konflikterkennung und der Konfliktaufhebung. Konflikte können beispielsweise durch Ineffizienzen sowie Redundanzen [SB05] oder mittels direkter Kommunikation und dem logischem Schließen [BLR01] identifiziert werden. Bei der Positionsbestimmung können aufgrund fehlerbehafteter Messwerte der Sensoren und Referenzsysteme abweichende Realpositionen ermittelt werden. Die abweichenden Realpositionen können negative Auswirkungen auf die Erstellung eines Abbildes der Systemumwelt und die Planung von Aufgaben und Routen nehmen. Beispielsweise können Routen nicht wie geplant befahren werden oder es folgen Kollision. Resultierend können sich Ineffizienzen für die Aufgabenverteilung ergeben. Des Weiteren bedingen vor allem eine dynamische sowie unscharfe Systemumwelt, eine dezentrale Koordination und heterogene Agenten die Implementierung von Verfahren der Konfliktlösung. Hierbei können ähnliche Situationen wie bei einer fehlerbehafteten Positionsbestimmung resultieren. Aufgrund sich ändernden Eigenschaften der Systemumwelt und einer individuellen Aufgabenplanung kann eine wiederholte Zuteilung von bereits zugewiesenen Aufgaben notwendig sein [SB05].

Die Konfliktlösung kann mit Hilfe von Normen unterstützt werden, welche zur Regulierung der Agenten beitragen. Normen beschreiben hierbei Verhaltensregeln für Situationen, in denen mehrere Agenten beteiligt sind. Entsprechend sind Normen ein übergeordnetes Regelwerk und kann dazu führen, dass individuelle Interessen der Agenten zurückgestellt werden [LL03]. Demzufolge dienen die Normen zur Vermeidung von störendem und unsozialem Verhalten, ohne das Design der Agenten vorzuschreiben oder die Autonomie einzuschränken [Dig99]. Aufbauend auf den Normen können Strategien der Prioritätenbewertung [IS89], Abstimmung [ER93], Vermittlung oder Modifikation [SIO+95] zur Konfliktlösung beitragen [BLR01].

Zusammenfassend werden die elementaren Ausführungsschritte eines Agenten für das TCP in der Abbildung 2.4 dargestellt. Hierbei können, angelehnt an die Teilprobleme des TCP, die drei nachfolgenden und wesentlichen Ausführungsphasen für einen Agenten unterschieden werden [VR95].

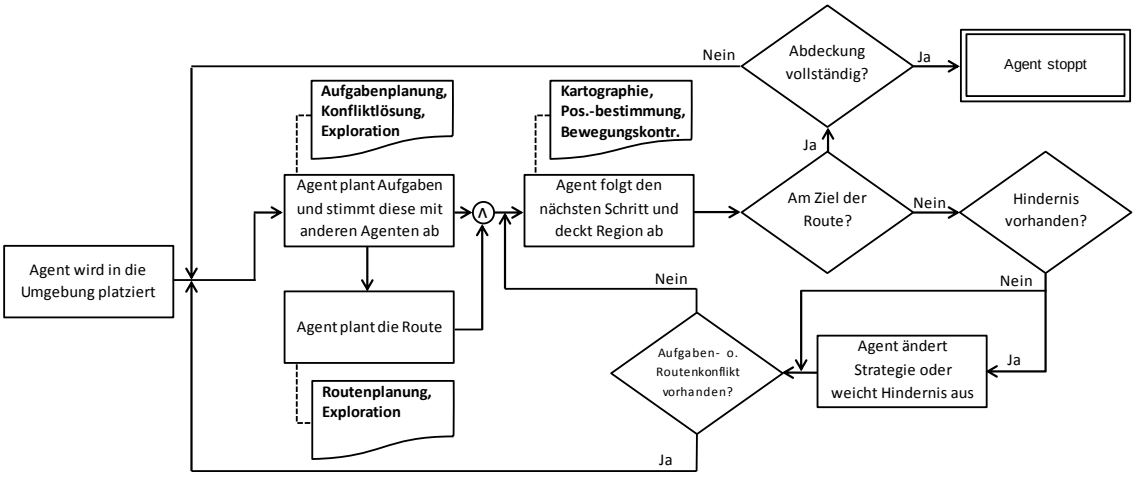


Abbildung 2.4.: Elementare Ausführungsschritte eines Agenten für das Terrain Covering Problem (erweitert und angelehnt an [VR95]).

- **Phase 1 - Aufgabenplanung:**
  Zu Beginn einer Abdeckung wird ein Agent in der Systemumwelt platziert. Ohne Informationen über die Systemumwelt muss der Agent zuerst die unmittelbare Systemumwelt mit den Sensoren wahrnehmen. Entsprechend werden Aufgaben/Ziele ermittelt, um die vollständige Abdeckung der Systemumwelt voranzutreiben. Aus der Menge möglicher Aufgaben ermittelt der Agent eine Reihenfolge der Bearbeitung. Die Aufgabenreihenfolge richtet sich nach den Optimierungszielen. Als mögliche Optimierungsziele können, beispielsweise die Abdeckung mit minimaler Redundanz oder auch die schnellste Abdeckung vorliegen. Nimmt der Agent im Bereich der Sensoren- und Kommunikationsreichweite andere Agenten wahr, dann sollte eine Abstimmung der Aufgaben erfolgen.
  Zu betrachtende Teilprobleme: Aufgabenplanung, Konfliktlösung, Exploration

- **Phase 2 - Routenplanung:**
  Die Routenplanung hingegen ermittelt geeignete Routen, um die übernommene Aufgabe bearbeiten zu können. Die Routenplanung erfolgt hierbei unter Berücksichtigung der Optimierungsziele.
  Zu betrachtende Teilprobleme: Routenplanung, Exploration

- **Phase 3 - Aufgabenerfüllung:**
  Nachdem die Aufgaben- und Routenplanung abgeschlossen ist, erfolgt die Erfüllung/Bearbeitung der übernommenen Aufgabe. Dafür befährt der Agent die zuvor geplante Route. Während die Route befahren wird, muss der Agent fortlaufend die eigene Position bestimmen, um Kursabweichungen feststellen und ein Abbild der Systemumwelt erstellen zu können. Des Weiteren erfolgt ein Informationsaustausch mit den anderen Agenten. Der Informationsaustausch dient der Identifizierung von Aufgabenkonflikten und kann zur Effizienzsteigerung hinsichtlich der Optimierungsziele genutzt werden. Wird ein Konflikt oder Optimierungspotenzial identifiziert, dann wird eine neue Aufgabenplanung der Phase 1 durchgeführt. Die Route wird solange befahren, bis der Agent am Ziel angekommen. Ist der Agent am Ziel der Route angekommen, dann geht der Agent wieder in die Phase 1 der Aufgabenplanung über, solange bis keine Aufgaben mehr zu vergeben sind und somit eine vollständige Abdeckung

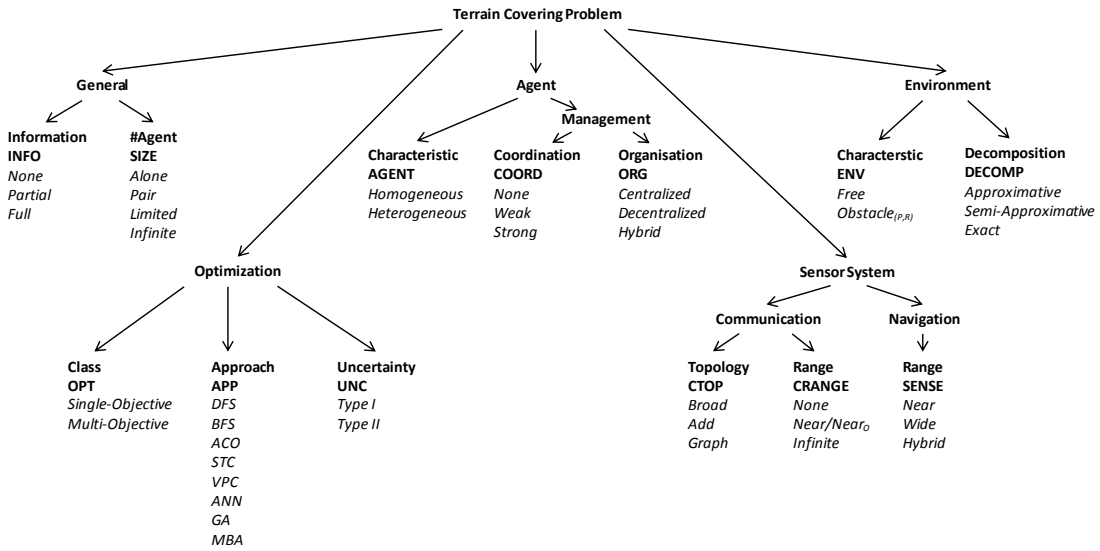


Abbildung 2.5.: Taxonomie der Modellierungsmöglichkeiten für ein TCM.

vorliegt. Wird beim Befahren der Route ein Hindernis identifiziert, dann umfährt der Agent das Hindernis. Bestehen hierbei Konflikte mit der aktuellen Aufgabe oder Befahrung der Route, dann führt der Agent ebenfalls eine Neuplanung der Aufgaben durch.
Zu betrachtende Teilprobleme: Positionsbestimmung, Kartographie, Bewegungskontrolle und Routenplanung, Exploration

Die Teilprobleme des TCP können mit unterschiedlichen Modellen und Ansätzen geeignet bearbeitet werden. Nachfolgend wird eine Taxonomie des TCP eingeführt, welche die Annahmen und Anforderungen an die Modelle zusammenfasst.

2.4. Eine Taxonomie für Multi Agent Terrain Covering Modelle

Nachfolgend wird eine Taxonomie für TCMs und Terrain Mapping Modelle (TMM[4]) eingeführt. Es werden verschiedene Kategorien definiert, die sich an den in der Literatur publizierten TCMs orientieren. Hierbei werden die Taxonomien in [DJW93, Cho01] erweitert. Zum einen werden neue Annahmen und Ansätze eingeordnet und zum anderen werden Eigenschaften und Anforderungen für das MA(R)S aufgenommen. Die TCMs können sich hinsichtlich allgemeiner TCP Charakteristika einschließlich Anforderungen an die Systemumwelt sowie Annahmen bezüglich der Agenten und deren Sensoren unterscheiden. Der Vollständigkeit halber wird die Unterscheidung hinsichtlich der Anzahl eingesetzter Agenten zusätzlich aufgeführt. In der Abbildung 2.5 werden die Kategorien der Taxonomie zusammengefasst.

Nach der Taxonomie wird eine Darstellung von bestehenden TCMs und TMMs vorgenommen. Anhand der Taxonomie wird daran anschließend eine Einordnung der vorgestellten Modelle durchgeführt. Neben einer Einordnung wird zudem eine Bewertung der Methoden und Annahmen sowie der verschiedenen Ausprägungen der Kategorien vorgenommen. Ferner wird das Ziel verfolgt, ver-

[4]Engl. Terrain Mapping Model

schiedene Fragestellungen und Forschungslücken aufzuzeigen, die bisher in der Literatur noch gar nicht oder nur ansatzweise dargestellt sind.

Grundsätzlich kann das TCP in die Klassen des online und offline TCP unterschieden werden. Abhängig von der jeweiligen Klasse werden Informationen über die Systemumwelt benötigt.

- **Verfügbarkeit von Vorwissen:**
  Den Agenten können Informationen bezüglich der Systemumwelt vorliegen, welche für die optimierte Abdeckung berücksichtigt werden können.

  - *INFO-NONE* Vor Beginn der Abdeckung stehen keine Informationen über die Systemumwelt zur Verfügung. Die Agenten nehmen alle Informationen aus der Systemumwelt während der Abdeckung wahr. Aus diesem Grund müssen die Agenten stets Anpassungen der Routen vornehmen, um eine optimierte Abdeckung durchzuführen. Ein TCM dieser Klasse wird als online TCM bezeichnet.
  - *INFO-PART* Vor Beginn der Abdeckung stehen einige Informationen über die Systemumwelt zur Verfügung. Die Informationen können für eine effiziente Abdeckung einbezogen werden. Beispielsweise können Umgebungsstrukturen bekannt sein oder auch Eigenschaften über die Systemumwelt. Hierbei können Informationen bezüglich des Energieaufwandes für das Befahren von Regionen bekannt sein. Neue Informationen müssen stets für eine optimierte Abdeckung in die Routenplanung einbezogen werden.
  - *INFO-FULL* Vor Beginn der Abdeckung ist die gesamte Systemumwelt vollständig bekannt. Die Abdeckung entspricht dem MCSP und es können optimale Routen vor der Abdeckung ermittelt werden. In der Regel sind die Ergebnisse für das offline TCM am effizientesten und optimaler als die der anderen beiden Klassen. Ein TCM dieser Klasse wird als offline TCM bezeichnet.

  Eine weitere Unterscheidung kann anhand der Anzahl eingesetzter Agenten vorgenommen werden. Grundsätzlich kann das Singleagent und Multiagent TCP unterschieden werden.

- **Anzahl der Agenten:**
  In Abhängigkeit von den zu erfüllenden Teilaufgaben, den Umweltbegebenheiten oder der einzuhaltenden Performance kann es notwendig sein, dass mehrere Agenten zum Einsatz kommen müssen. In dieser Dissertation werden Modelle der Klasse *SIZE-LIM* betrachtet.

  - *SIZE-ALONE* Es wird genau ein Agenten eingesetzt, um die zu erfüllenden Ziele zu erreichen. Der Agent verfügt über alle Fäigkeiten zur Erreichung der übergeordneten Ziele. Fällt der Agent aus, ist die Zielerreichung unmöglich.
  - *SIZE-PAIR* Es wird die kleinste Gruppe von genau zwei Agenten eingesetzt. Die Agenten können sich in den Fähigkeiten ergänzen. Kritische Fähigkeiten, die für eine Zielerreichung notwendig sind, sollten redundant vorhanden sein.
  - *SIZE-LIM* Es wird eine, im Verhältnis zur Systemumwelt oder den zu erfüllenden Zielen, begrenzte Anzahl von Agenten eingesetzt. Es gelten die gleichen Eigenschaften wie für *SIZE-PAIR*. Ferner können die übergeordneten Ziele in Teilaufgaben zerlegt werden, welche parallel und disloziert bearbeitet werden können.
  - *SIZE-INF* Es wird eine, im Verhältnis zur Systemumwelt oder den zu erfüllenden Zielen, sehr hohe Anzahl von Agenten eingesetzt.

Des Weiteren gibt es verschiedene Optimierungsansätze für den einkriteriellen und mehrkriteriellen Fall.

- **Ausprägung der Optimierung:**
  Für ein TCM werden unterschiedliche Optimierungsziele betrachtet. Häufig wird nur die totale Explorationszeit minimiert. Unter Umständen kann es in einigen Anwendungsfällen von Nöten sein, dass verschiedene Optimierungsziele zeitgleich betrachtet werden.

  - *OPT-SOO* Das Verhalten der eingesetzten Agenten ist auf ein einkriterielles Optimierungsziel ausgerichtet.
  - *OPT-MOO* Mit Hilfe der mehrkriteriellen Optimierung werden verschiedene und zumeist konträre Optimierungsziele zeitgleich betrachtet. Hierbei kann beispielsweise die Minimierung der Explorationszeit sich gegensätzlich zur Minimierung der Energiekosten oder Bedrohungen verhalten. Aus diesem Grund müssen fortlaufend geeignete Kompromisslösungen gefunden werden.

- **Zugrundeliegender Optimierungsansatz:**
  Es werden verschiedene Optimierungsansätze des Operations Research genutzt, um eine möglichst geeignete Abdeckung zu ermitteln. Im nachfolgenden werden die wesentlichen Heuristiken aufgezeigt, die einen unterschiedlichen Ursprung aufweisen. Es werden unter anderem betriebswirtschaftliche, biologische und graphentheoretische Beobachtungen für Verfahren und Algorithmen des TCP adaptiert.

  - *APP-GREEDY/BFS* Die Klasse der Greedy Verfahren um das Depth First Search (DFS) stellen intuitive Ansätze dar, um geeignete Routen im Graphen bestimmen zu können. Es wird stets die Zelle abgedeckt, die zum Zeitpunkt der Entscheidungsfindung der optimalen Lösung entspricht. Das Auswahlkriterium kann sich beispielsweise nach dem maximalen Informationszugewinn oder den minimalen Kosten richten. Ein ähnlicher Ansatz wird mit dem Breadth First Search (BFS) Verfahren verfolgt. Ausgehend von einer Zelle werden zuerst alle umliegenden, unbekannten Zellen abgedeckt [PS90, GF05, YJL08].
  - *APP-ACO* Ein dem GREEDY und BFS ähnliches Verfahren umfasst die Klasse der ACO Metaheuristik [Dor92]. Es werden die Verhaltensweisen von Ameisen simuliert, welche Pheromone an die Systemumwelt abgeben. Die Entscheidungsfindung der Agenten basiert auf den lokalen Informationen, die aus den Pheromonen abgeleitet werden. Die Auswahlwahrscheinlichkeit der Zellen verhält sich proportional zu dem zugehörigen Pheromonwert. Aufgrund der Auswahlwahrscheinlichkeit werden auch Zellen betrachtet, die gemäß der lokalen Informationen nicht zwangsläufig optimal sind. Des Weiteren wird eine Evaporation und Diffusion der Pheromonwerte simuliert. Mit Hilfe der Evaporation wird der Pheromonwert kontinuierlich erhöht/verringert. Die Diffusion bietet hingegen die Möglichkeit, dass besonders geeignete Zellen einen positiven Einfluss auf die Pheromonwerte in der Nachbarschaft nehmen [WLB96, Pre14].
  - *APP-STC* Die graphentheoretischen Algorithmen zur Bestimmung von Spannbäumen können zur Ermittlung und Planung von Routen eingesetzt werden. Ein Spannbaum enthält alle Knoten eines ungerichteten und zusammenhängenden Graphen. Für gewichtete Graphen existieren minimale Spannbäume, die alle Knoten mit minimalen Kantengewicht verbinden. In Übertragung auf das TCP können DFS und BFS Algorithmen

genutzt werden, um einen Spannbaum zu ermitteln. Der Spannbaum verbindet alle befahrbaren Zellen miteinander. Resultierend kann eine vollständige Abdeckung durch die Befahrung des Spannbaumes garantiert werden. Ein weiterer Vorteil der Nutzung von Spannbäumen ist, dass für eine große Menge von TCP-Instanzen eine optimale Abdeckung, im Sinne der Redundanz, garantiert werden kann [AHK06, SB10].

- *APP-MBA* Die Klasse der MBA Verfahren orientiert sich an betriebswirtschaftlichen Beobachtungen sowie an ökonomischen Märkten mit dem Ziel der optimierten Koordination und Verteilung von Ressourcen beziehungsweise Aufgaben. Die Grundlage von marktbasierten Systemen ist das zugehörige Preissystem. Für das TCP werden die Kosten der einzusetzenden Ressourcen zur Übernahme/Tausch einer Aufgabe in Form einer Kosten-Nutzenfunktion definiert. Der ökonomische Wettbewerb unter den Agenten, im Sinne des Selbstinteresses[5], führt zu einer Optimierung der globalen Zielfunktion. Der Handel kann mittels Marktmechanismen, wie beispielsweise den Auktionen erfolgen. Die Agenten können, abhängig von dem jeweiligen Zustand der Systemumwelt, die Rolle des Auktionators und/oder des Bieters ausführen und bedürfen keines statischen, zentralen Koordinators [SB05, SYTX06].
- *APP-WEITERE ANSÄTZE* Neben der ACO gibt es noch weitere (Meta-)Heuristiken, der Klasse des Natural Computing, die für das TCP genutzt werden. Der evolutionäre genetische Algorithmus (GA[6]) kann mit Hilfe simulierter Generationen und der zugehörigen Selektion, Rekombination und Mutation geeignete Routen für die Agenten ermitteln. Eine Schwierigkeit besteht in der Transformation der Routen in einen Genotypen [OYKP09]. Aus dem Forschungsfeld der künstlichen Intelligenz können künstlich neuronale Netze (ANN[7]) für das TCP genutzt werden. Hierbei wird jede Zelle durch ein Neuron repräsentiert. Die Aktivierungsfunktion der Neuronen kann beispielsweise die Attraktivität unbekannter Zellen steigern und die Bewegungsrichtung der Agenten beeinflussen [LY02].

Weiterhin unterscheiden sich die TCP Modelle hinsichtlich der Ausprägung von Unsicherheiten. Hierbei kann zum einen idealtypisch davon ausgegangen, dass keine Unsicherheiten bezüglich der Systemumwelt und der Agenten bestehen. Zum anderen können Unsicherheiten unterstellt werden, die in die Optimierung des TCP einbezogen werden.

- **Ausprägung der Unsicherheiten:**
  Es können zwei unterschiedliche Klassen von Unsicherheiten auf ein TCM einwirken. Hierbei können Unsicherheiten und Störungen von den eingesetzten Sensoren und Effektoren ausgehen. Zum anderen treten, insbesondere für die Routenplanung bei einem online TCM, Unsicherheiten bezüglich der unbekannten Systemumwelt auf. Werden die Unsicherheiten im Optimierungsprozess nicht betrachtet, können auf Grundlage der falschen Informationen Fehlentscheidungen hinsichtlich der Optimierungsziele getroffen werden. Um derartige Fehlentscheidungen zu minimieren, werden die Unsicherheiten bei der robusten Optimierung einbezogen. In Kapitel 5 wird ein neuer Ansatz für das TCP unter Unsicherheiten vorgestellt und untersucht.

[5] Im ökonomischen Sinne beschreibt das Selbstinteresse einen Zustand, in dem ein Individuum bestrebt ist, ausschließlich die eigenen Ziele zu erreichen. Eine gesteigerte Form des Selbstinteresses ist der Opportunismus, für das ein Individuum versucht, die eigenen Ziele, ohne Rücksicht negativer Auswirkungen auf andere Individuen, zu erreichen [Wil75].
[6] Engl. Genetic Algorithm
[7] Engl. Artificial Neural Network

- *UNC-TYPE I* Der Typ I umfasst unkontrollierbare Faktoren, beziehungsweise Unsicherheiten, die aus der Systemumwelt resultieren. Die Routenplanung unterliegt Unsicherheiten, weil für die Systemumwelt keine beziehungsweise nur unvollständige Informationen vorliegen. Des Weiteren kann eine dynamische Systemumwelt vorliegen, für die Änderungen auftreten und bei einer optimierten Entscheidungsfindung einbezogen werden müssen.
- *UNC-TYPE II* Der Typ II umfasst Unsicherheiten, die bezüglich der Designvariablen auftreten. Die Sensoren und Effektoren unterliegen negativen Messeinflüssen, die Unsicherheiten und Störungen verursachen können. Die Folge sind Positionsabweichungen bezüglich der geplanten Route oder Fehler in den übermittelten Kommunikationsnachrichten.

Abhängig von dem implementieren Optimierungsansatz und eingesetzten Agenten werden Anforderungen und Annahmen an die Agenten selbst gestellt.

- **Ausprägung der Agenten:**
  Die Agenten können sich hinsichtlich der technischen Umsetzung und den Verhaltensweisen unterscheiden. Kritische Fähigkeiten sollten möglichst über viele Agenten verteilt sein.

  - *AGENT-HOM* Die Agenten sind in der Ausprägung der Hard- und Kontrollsoftware identisch und demnach austauschbar. Aus diesem Grund kann jeder Agent die Aufgaben eines ausgefallenen Agenten übernehmen.
  - *AGENT-HET* Die Agenten verfügen über unterschiedliche Sensoren, Effektoren, Kommunikationsmöglichkeiten, Kontrollmechanismen oder einem Regelsatz von Verhaltensweisen.

  Werden mehrere Agenten eingesetzt, können Kategorien unterschieden werden, welche die Eigenschaften der Kooperation beschreiben. Insbesondere in dynamischen und unbekannten Umgebungen ist eine robuste Kooperation für autonome Agenten unabdingbar, um Aktionen planen zu können. Eine geeignete Kooperation folgt hierbei den vier Phasen: Handlungsnotwendigkeit erkennen, Notwendigkeit melden, Aktionen planen und auswählen sowie Aktionen zuweisen.

- **Koordination der Agenten:**
  Die Koordination der Agenten gibt an, in welchem Ausmaß das Verhalten anderer Agenten in den Entscheidungsprozessen, bezüglich zukünftiger Aktionen der Agenten, einbezogen wird [INS00]. Für TCMs wird ausschließlich die strukturierte Koordination betrachtet, um ein kohärentes und performantes Systemverhalten auszuprägen.

  - *COORD-NONE* Es besteht keine Koordination der Agenten in dem Multiagenten System. Resultierend werden keine Informationen im MAS ausgetauscht und die Aktionen der Agenten sind unabhängig voneinander [INS00].
  - *COORD-WEAK* Die Koordination der Agenten basiert auf individuellen Interaktionen und folgt hierbei lediglich einem schwach strukturierten Koordinationsprotokoll. Inwiefern das Verhalten anderer Agenten berücksichtigt wird, ist stark situationsabhängig [INS00].

- *COORD-STRONG* Die Koordination der Agenten basiert auf einem gemeinsamen Regelwerk, welches die Interaktion zwischen den Agenten definiert. Das Verhalten anderer Agenten wird bei den Entscheidungsprozessen stets berücksichtigt [INS00].

- **Organisation der Agenten:**
  Die Organisation der Agenten gibt an, von welchen Agenten die Entscheidungsprozesse durchgeführt werden. Limitierende Faktoren bezüglich der Zuweisung von Aufgaben stellen die Kommunikationsmöglichkeiten und die Sensorausprägung dar. Die nachfolgenden Organisationsklassen setzen eine strukturierte Koordination voraus.

  - *ORG-CEN* Entscheidungen werden für alle Agenten zentral von einem Koordinator getroffen. Der Koordinator kann selbst ein Kontroll- und Planungsagent, eine stationäre Einrichtung oder eine übergeordnete Kontrollinstanz sein. Die Agenten müssen alle Informationen dem Koordinator bereitstellen, welcher die Informationsverarbeitung durchführt und ein einheitliches, operationelles Lagebild erstellt. Resultierend bezieht der Koordinator alle vorhanden Informationen in die Entscheidungsfindung ein. Ferner besteht die Notwendigkeit einer uneingeschränkten Kommunikationsreichweite. Insofern keine Anweisungen durch den Koordinator übermittelt werden können, fallen die Agenten aus.
  - *ORG-DEC* Entscheidungen werden autonom, basierend auf einem Regelsatz, den Beobachtungen und den lokal vorhandenen Informationen, dezentralisiert von jedem Agenten selbst getroffen. Demzufolge ist die Aufgabenplanung für einen dezentralen Ansatz robust gegenüber Kommunikationsstörungen. Die Güte der Abdeckung und Effizienz sind von der Informationsverfügbarkeit abhängig.
  - *ORG-HYB* Die Koordination setzt sich aus zentralen sowie dezentralen Elementen zusammen. Die Rolle des Koordinators kann dynamisch, in Abhängigkeit von Änderungen der Systemumwelt oder unter Effizienzbetrachtung, zwischen den Agenten wechseln [INS00]. Folglich ist die Aufgabenplanung für einen hybriden Ansatz robuster gegenüber Kommunikationsstörungen.

Eine geeignete Kooperation der Agenten setzt eine Kommunikation zwischen den Agenten voraus, welche durch nachfolgende Kategorien charakterisiert werden kann.

- **Topologie der Kommunikation:**
  Die Topologie der Kommunikation beschreibt zwischen welchen möglichen Agenten ein Informationsaustausch stattfindet. Die Agenten können über einen eindeutigen Bezeichner verfügen, der einen gezielten Informationsaustausch ermöglicht. Es kann ebenso ein Informationsaustausch mit allen Agenten erforderlich sein.

  - *CTOP-BROAD* Jeder Agent kommuniziert mit allen Agenten, die sich in der definierten Kommunikationsreichweite befinden. Es ist nicht möglich, mit einem Agenten gezielt zu kommunizieren. Der Ansatz ist robuster gegenüber Kommunikationsfehler, weil die Wahrscheinlichkeit hoch ist, dass mindestens ein Agent die gesendete Nachricht erhält.
  - *CTOP-ADD* Jeder Agent kann mit Hilfe des eindeutigen Bezeichners gezielt Informationen mit anderen Agenten austauschen.
  - *CTOP-GRAPH* Welche Agenten miteinander kommunizieren können, wird durch einen ungerichteten Graphen definiert. Das Kommunikationsnetzwerk kann sich dynamisch

anpassen. Informationen können hierbei über vermittelnde Agenten weitergesendet werden, so dass die Kommunikationsreichweite bestmöglich ausgenutzt werden kann.

Bezüglich der Robotik wird beispielsweise in [Yuh00] eine Klassifizierung von Sensoren für mobile Roboter in drei Klassen vorgenommen. Es werden die Navigations-, System- und Missionssensoren unterschieden. Aufgrund der eingesetzten Kommunikationssensoren, welche Bestandteil der Missionssensoren sind, kann die Kommunikationsreichweite limitiert sein.

- **Ausprägung der Kommunikationsreichweite:**
  Basierend auf der Kommunikationstechnologie und den Umweltbedingungen kann eine Kommunikation zwischen den Agenten, gemäß der Reichweite, eingeschränkt sein.

  - *CRANGE-NONE* Die Agenten können nur indirekt miteinander kommunizieren. Gemäß der Stigmergie[8] kann zwischen einer aktiven und passiven, indirekten Kommunikation unterschieden werden. Bei der aktiven, indirekten Kommunikation werden Informationen von anderen Agenten aus der Systemumwelt gesammelt. Hierfür werden Informationen, bezüglich Handlungsempfehlungen und der Systemumwelt, in Form von Markern in der Systemumwelt platziert. Es ist auch möglich, dass mittels stationärer Kommunikationseinrichtungen ein Informationsaustausch vollzogen wird. Bei der passiven, indirekten Kommunikation werden mittels der Sensoren die Änderungen der Systemumwelt oder die Anwesenheit anderer Agenten registriert sowie das Verhalten der Agenten beobachtet [YJA13].
  - *CRANGE-NEAR* Die Agenten können nur innerhalb einer vordefinierten Distanz miteinander kommunizieren. Topologisch kann sich die Definition der Reichweite auch auf den am nächstliegenden Agenten beziehen.
  - *CRANGE-NEAR*$_O$ Die Kommunikationsreichweite richtet sich nach CRANGE-NEAR mit der Einschränkung, dass die Kommunikation durch Hindernisse unterbrochen wird.
  - *CRANGE-INF* Die Agenten können ohne Einschränkungen über die gesamte Systemumwelt hinweg miteinander kommunizieren. Diese Annahme ist auf eine praktische Anwendung bezogen eher unwahrscheinlich.

  Des Weiteren können verschiedene Navigationssensoren hinsichtlich der Sensorreichweite unterschieden werden.

- **Ausprägung der Navigationssensoren:**
  Die Navigationssensoren dienen der Positionsbestimmung, Kartographie, Bewegungssteuerung und Routenplanung. Hingegen dienen die Systemsensoren der Überwachung von systemspezifischen Parametern. Des Weiteren werden die Missionssensoren unterschieden, welche in Abhängigkeit des Auftragsziels integriert werden. Im Bereich des TC sind Sensoren für die Wahrnehmung der Systemumwelt notwendig, die sich hinsichtlich der Reichweite unterscheiden können.

[8] Gemäß der Stigmergie nach P.-P. Grassé kommunizieren die Individuen indirekt miteinander. Hierfür werden Informationen beziehungsweise Stoffe in der Systemumwelt platziert. Bei der aktiven Stigmergie wird durch die Individuen eine gezielte Veränderung der Systemumwelt vorgenommen, um das Verhalten anderer Individuen zu beeinflussen. Hingegen werden bei der passiven Stigmergie Änderungen der Systemumwelt durch die Aktionen der Agenten hervorgerufen werden [Gra59, Ell07].

- *SENSE-NEAR* Die Agenten verfügen über Navigationssensoren, die ausschließlich die Beschaffenheit der unmittelbaren Systemumwelt wahrnehmen können. Zu dieser Klasse gehören auch die taktilen Sensoren.
- *SENSE-WIDE* Die Agenten verfügen über Navigationssenoren mit deren Hilfe die Beschaffenheit der Systemumwelt innerhalb der Sichtgrenzen wahrgenommen werden kann. Die Reichweite kann durch Umgebungsstrukturen beschränkt werden.
- *SENSE-HYB* Die Agenten verfügen über Navigationssensoren für den Nah- und Fernbereich.

Mit der Hilfe des Sensorsystems können sich die Agenten in der Systemumwelt bewegen. Die TCP Modelle stellen hierbei verschiedene Anforderungen an die Ausprägung der Systemumgebung.

- **Ausprägung der Systemumgebung:**
  Die Algorithmen der TCMs und TMMs setzen eine unterschiedliche Beschaffenheiten der Umgebung voraus. Des Weiteren wird in simulativen Experimenten häufig eine Abstrahierung der Systemumgebung vorgenommen. Die Grenzen der Systemumgebung können hierbei auf polygonale Strukturen reduziert werden.

  - *ENV-FREE* Die Systemumgebung verfügt ausschließlich über äußere Systemgrenzen und ist frei von Hindernissen.
  - *ENV-OBS* Die Systemumgebung kann Hindernisse enthalten.
  - *ENV-[FREE|OBS]*$_P$ Die Systemumgebung setzt sich aus polygonal gearteten Umgebungsstrukturen zusammen.
  - *ENV-[FREE|OBS]*$_R$ Die Systemumgebung beinhaltet nur rechteckige Umgebungsstrukturen. Jede Umgebungsform kann in eine *ENV-[FREE|OBS]*$_R$ überführt werden. Zellen, die in Form und Größe nicht der Einheitszelle entsprechen, werden abstrahiert. Somit stellt die Vereinigung der Zellen nur eine Annäherung der zugehörigen *ENV-OBS* dar.

  Abschließend werden unterschiedliche Verfahren zur Repräsentation und internen Abbildung der Systemumwelt durch die Agenten genutzt.

- **Topologie der Zerlegung einer Systemumwelt:**
  Die Zerlegung der Systemumwelt in Zellen ermöglicht den Agenten eine eindeutige Beschreibung von Regionen vorzunehmen. Infolgedessen können Aufgabenzuweisungen und Beobachtungen den jeweiligen Zellen zugeordnet und kommuniziert werden. Es werden drei Formen der Zerlegung einer Systemumwelt unterschieden. Bei der exakten Zerlegung einer Systemumwelt *DECOMP-EXACT* sind die Zellen disjunkt. Die Zellengröße ist unbestimmt und richtet sich nach den Begrenzungen der Systemumwelt und den Hindernissen. Für die semi-approximative Zerlegung der Systemumwelt *DECOMP-SEMIAPPROX* ist jeweils eine räumliche Dimension, beispielsweise die Höhe oder Breite, vordefiniert. Die jeweils andere räumliche Dimension passt sich hingegen der Systemumwelt an. Die approximative Zerlegung der Systemumwelt *DECOMP-APPROX* ist die einfachste Form, für die eine Zerlegung in gleich große Zellen vorgenommen wird.

- *DECOMP-EXACT*$_{SL}$ Die Slice Zerlegung unterteilt die Systemumwelt in Zellen, die anhand von kritischen Punkten in der Systemumwelt bestimmt werden. Es stehen sich zwei Agenten in der Systemumwelt gegenüber und bewegen sich parallel auf einer Dimension fort. Ein kritischer Punkt, beziehungsweise Hindernis, wird genau dann ermittelt, wenn die Sichtlinie zwischen beiden Agenten unterbrochen wird [CP97, CFK97]. In [But00, BRH00] wird ein Verfahren beschrieben, dass die Slice Zerlegung mit einem Agenten durchführen kann. Hierzu bewegt sich der Agent in einer Boustrophedonbewegung[9] durch die Umgebung. Stößt der Agent auf ein Hindernis wird ebenfalls die aktuelle Zelle abgeschlossen.
- *DECOMP-EXACT*$_{MSA}$ Die Zerlegung der Minimal Sum of Altitude ist nur als offline Verfahren und für *ENV-* $\cdot_P$ Systemumgebungen anwendbar. Die Zellen werden zuerst, gemäß der semi-approximativen Zerlegung, ausgehend von jeder Kante in der Systemumgebung, ermittelt und übereinandergelegt. Daran anschließend wird ein Nachbarschaftsgraph erzeugt, deren Knoten die Teilregionen repräsentieren. Der Nachbarschaftsgraph wird derart geteilt, dass jene Kombination ermittelt wird, für welche die Gesamtanzahl der Richtungsänderungen minimiert wird [Hua01].
- *DECOMP-EXACT*$_{VB}$ Für die Visibility-Based Zerlegung werden die Zellen entsprechend der Sensorreichweite um einen Agenten herum bestimmt. Hindernisse und Begrenzungen der Systemumwelt können die Sensorreichweite einschränken und sind zugleich die Begrenzung einer Zelle [CH94, JT08].
- *DECOMP-EXACT*$_{VP}$ Für die Voronoi-Based Zerlegung werden die Zellen entsprechend der Agentenpositionen in der Systemumwelt bestimmt. Es wird ein Punkt im Raum der zugehörigen Systemumwelt, genau dem Agenten zugeordnet, welcher gemäß der definierten Abstandsmetrik näher zu dieser liegt, als jeder andere Agent. Die Gesamtheit aller Punkte werden als Voronoi-Region bezeichnet und ergeben die von dem Agenten abzudeckende Zelle [GWD12].
- *DECOMP-SEMIAPPROX* Die semi-approximative Zerlegung basiert auf der Boustrophedonbewegung der Agenten. Die Breite einer Boustrophedon Abdeckung richtet sich nach der Sensorreichweite. Die Ausprägung der Zelle in Bewegungsrichtung ist abhängig von den Begrenzungen der Systemumwelt. Hindernisse werden in gleicher Weise erkundet [HTL96].
- *DECOMP-APPROX* Die Zellengröße und -form entspricht oftmals der Sensorreichweite der Agenten, kann aber beispielsweise auch eine trigonometrische [OCPZ03], quadratische [LC98] oder hexagonale [HZ91] Form annehmen. Demzufolge entspricht der Besuch einer Zelle der Abdeckung. Die Zellen können hierarchisch zusammengefasst und effizient verwaltet werden. Zellen die aufgrund der Begrenzungen der Systemumwelt nicht vollständig, gemäß der vorgegebenen Struktur sind, werden nicht abgedeckt. Eine Verkleinerung der Zellen kann diesem Problem entgegenwirken, hat jedoch eine exponentielle Auswirkung auf die Speicheranforderung [Thr98]. Ferner stellt die approximative Zerlegung ein nicht exaktes Abbild der Systemumwelt dar [Elf87, GR01].

Nachfolgend werden verschiedene TCMs und TMMs aus der Literatur vorgestellt. Hierbei wird aufgezeigt, wie nun die Teilprobleme des TCP, mittels geeigneter Methoden, umgesetzt werden. Des Weiteren wird ein Vergleich der TCMs anhand wesentlicher Eigenschaften eines MA(R)S vorgenom-

[9] Sinnbildlich wird unter der Boustrophedonbewegung der Weg eines Ochsen verstanden, welcher mit geradlinigen Bewegungen entlang einer Kante ein Feld bestellt.

men. Ein Vergleich zeigt dabei die Stärken und Schwächen der TCMs, hinsichtlich der Robustheit, Flexibilität, Kohärenz und Aufgabenverteilung auf. Daraus folgend kann eine erste Bewertung der umgesetzten Methoden und Vorgehensweisen durchgeführt werden.

2.5. Modelle für das Terrain Covering Problem

Im Forschungsumfeld des Terrain Covering existieren unterschiedliche Ansätze. Die Ansätze basieren beispielsweise auf konzeptionellen Überlegungen aus den Bereichen der Robotik, der Biologie, der Wirtschaftswissenschaften, der Graphentheorie und dem Operations Research. Hierbei werden etablierte Verfahren genutzt, um die an ein autonomes MAS gestellten Voraussetzungen zu erfüllen. Dieses Kapitel gibt einen Überblick bezüglich bestehender, onlinebasierter TCMs wieder, die eine Koordinationsstruktur für eine Gruppe von Agenten aufweisen. Die nachfolgenden TCMs unterscheiden sich hinsichtlich der Kommunikationsstruktur, der Zerlegung einer Systemumwelt und des Koordinationsmechanismus. Des Weiteren werden TMMs erfasst, die durch wenige Anpassungen in ein TCM überführt werden können.

Für die nachfolgenden TCMs wird eine Evaluation hinsichtlich der in Kapitel 3.2.2 aufgeführten wesentlichen Eigenschaften eines MA(R)S vorgenommen.

2.5.1. Multi Agent Terrain Covering Modelle

- **APP-GREEDY/BFS:**
  In [PS90] wird ein sehr einfaches TCM vorgestellt. Hierbei basiert das Node Counting Modell auf einfachen Regeln der Koordination, welche mittels einer indirekten Kommunikation realisiert wird. Jede Zelle verfügt über einen Zählstand. Die Zellen können verschiedene polygonale Ausprägungen annehmen [OCPZ03, Pre14]. Besucht ein Agent eine Zelle wird der Zählstand um eins erhöht und die Zelle ist vollständig abgedeckt. Die Agenten bewegen sich zu der Zelle, die den niedrigsten Zählstand aufweist. Weisen mehrere Zellen den gleichen Zählstand auf, dann wird für das einfachste Modell zufällig eine dieser Zellen gewählt.

  Eine Erweiterung des Node Counting Modells wird in [FTL07] beschrieben. Der Brick&Mortar Algorithmus verhindert, dass die Agenten in einer bereits besuchten Region eingeschlossen werden und Redundanzen in der Abdeckung folgen. Der Algorithmus garantiert, dass Zellen maximal zweimal abgedeckt werden. Bei einer erstmaligen Abdeckung einer Zelle markieren die Agenten die Zelle als besucht, genau dann wenn alle zugänglichen Zellen in der Nachbarschaft über alternative Pfade erreicht werden können. Entsprechend wird die aktuelle Zelle als unzugängliche beziehungsweise blockierte Zelle behandelt. Existieren keine alternativen Pfade, dann wird die aktuelle Zelle als erkundet markiert. Im nächsten Schritt werden alle Zellen der Nachbarschaft priorisiert. Hierbei werden nicht erkundete Zellen mit bereits besuchten oder unzugänglichen Zellen in der zugehörigen Nachbarschaft höher priorisiert. Enthält die Nachbarschaft keine unbekannten Zellen, bewegt sich der Agent zufallsbedingt zu einer erkundeten Zelle.

Ähnlich dem TCM von Rekleitis et al. [RNRC08] wird in [YJL08] ein Multi Agent TCM vorgestellt, welches eine semi-approximative Zerlegung der Systemumwelt nutzt. Hierbei orientieren sich Jung et al. an dem zugrundeliegenden Singleagent TCM in [HTL96]. Hierbei werden, unter Zuhilfenahme von lokalen Informationen, Effizienzsteigerungen für die Erkundung von Hindernisstrukturen ermittelt. Die Systemumwelt wird in Zellen untergliedert, deren horizontale Ausdehnung entlang der x-Achse gemäß der Sensorreichweite vordefiniert ist. Die vertikale Ausdehnung entlang der y-Achse passt sich flexibel der Systemumwelt an. Für konvexe und konkave Hindernisstrukturen werden Markierungspunkte in der Systemumwelt gesetzt, um eine redundante Abdeckung zu minimieren. Die Agenten werden in Erkunder und Abdecker unterteilt. Aufgrund einer dezentralen Rollenzuweisung ermitteln die Agenten, auf Grundlage von Informationen aus der Systemumwelt sowie den Systemzuständen aller Agenten, die individuelle Rolle dynamisch selbst. Die Erkunder bewegen sich entlang der Zellen und übermitteln die Markierungspunkte der Hindernisse an alle Agenten. Die Hindernisse werden dann von den Abdeckern untersucht. Ist kein Abdecker verfügbar, kann der Erkunder die Rolle wechseln und selbst als Abdecker fungieren. Die generische Formulierung des TCM ermöglicht die Abdeckung eines dreidimensionalen Raumes. Es wird ebenfalls eine Erweiterung vorgestellt, die eine robuste Abdeckung bei Einwirkung von externen Störungen und internen Navigationsfehlern ermöglicht. Hierbei richtet sich die Ermittlung der horizontalen Ausdehnung nach der maximalen Abweichung vom Kurs.

Butler et al. haben ausgehend von dem CC_R[10] Singleagent TCM in [But00] ein DC_R[11] Multiagent TCM [BRH00] entwickelt, welches für rechteckig ausgeprägte Systemumgebungen eingesetzt werden kann. Die Agenten benötigen nur Tastsensoren und die Zellenzerlegung wird ähnlich dem Slice TCM vorgenommen. Ausgehend von den Startpunkten wird die Systemumwelt mittels der Boustrophedonbewegung untersucht. Wird eine Änderungen der maximalen Ausdehnung registriert, wird die Zelle abgeschlossen. Die Zugänge zu unbekannten Zellen werden in einem Stack verwaltet. Die Koordination der Agenten wird durch eine zentrale Steuerungseinheit übernommen, welche die Zelleninformationen zu einer Gesamtsystemumwelt zusammenfügt.

In [GF05] wird ein TCM vorgestellt, mit dem Ziel die Abdeckungsredundanzen für ein MARS zu minimieren. Es werden die Annahmen vorausgesetzt, dass es keine Kommunikationseinschränkungen gibt und die Agenten jeweils eine Zelle vollständig abdecken. Nach dem Informationsaustausch über die bereits erkundete Systemumwelt, wird die nächste Zelle aus der unmittelbaren Systemumwelt bestimmt. Hierbei liegt diese Zelle mindestens zwei Zellen von Hindernissen und vier Zellen von den bereits abgedeckten Zellen entfernt. Können diese Bedingungen nicht erfüllt werden, wird die Grenze der bereits selbstständig erkundeten Systemumwelt gegen den Uhrzeigersinn untersucht. Dieser Vorgang wird solange fortgeführt bis die Bedingungen wieder erfüllt werden können. Konkave Hindernisse werden mit der Boustrophedonbewegung untersucht. Aufgrund dieser Regeln bleibt die noch unbekannte Systemumwelt zusammenhängend und Abdeckungsredundanzen werden minimiert.

In [BS03] wird eine indirekte Kommunikationsstruktur genutzt, die mittels stationärer Kommunikationseinrichtungen umgesetzt wird. Kommen die Agenten in den Kommunikationsbereich einer Signalstation, werden Informationen über die Systemumwelt ausgetauscht. Ba-

[10] Engl. Contact-Based Coverage Of Rectilinear Environment
[11] Engl. Distributed Coverage Of Rectilinear Environment

sierend auf den Informationen der Systemumwelt geben die Signalstationen den Agenten die Explorationsrichtung vor. Die Kommunikationseinrichtungen werden in vordefinierten Abständen in der Systemumwelt platziert, insofern keine von den Agenten wahrgenommen wird.

Sotzing et al. nutzen in [SJL08] einen skriptbasierten Koordinationsmechanismen für das MARS. In einem Stoplight System wird für jeden Agenten ein Ablaufskript definiert, für das in diskreten Zeitintervallen oder Markierungen im Ablauf bestimmte Synchronisationsereignisse ausgelöst werden. Hierbei werden die gesammelten Informationen bezüglich der Systemumwelt und der Aufgabenbearbeitung zwischen den Agenten ausgetauscht. Kommunikationsstörungen wirken sich negativ auf die Aufgabenplanung aus. Um diesen negativen Effekt entgegenzuwirken, wird die Frequenz von Synchronisationsereignissen erhöht. Resultierend steigt der Kommunikationsaufwand und zusätzlich die Komplexität der Kommunikation [SL08b].

Rekleitis et al. stellen in [RLNC04, RNRC08] ein online Slice TCM vor, in der eine exakte Zerlegung der Systemumwelt vorgenommen wird. Hierbei orientieren sich Rekleitis et al. an dem Boustrophedon Ansatz nach Choset et al. in [CP97, CARL00]. Es wird zwischen den Erkundern und den Abdeckern unterschieden. Mindestens zwei Erkunder stehen sich auf gleicher Höhe an der oberen und unteren räumlichen Begrenzung der Systemumwelt gegenüber. Die Agenten bewegen sich gleichmäßig in eine Richtung. Währenddessen bewegen sich die Abdecker in einer Boustrophedonbewegung zwischen den Erkundern [CFK97]. Es wird eine Zelle gebildet, wenn die Sichtverbindung unterbrochen wird. Insofern genügend Agenten in der Gruppe vorhanden sind, dann teilt sich die Gruppe in zwei Teilgruppen auf. Eine Gruppe muss über mindestens zwei Erkundern verfügen.

- **APP-ACO:**
  In [WLB96] wird von Wagner et al. eine Erweiterung des Greedy Algorithmus um die ACO Metaheuristik vorgestellt. Für den Mark-Ant-Walk Algorithmus hinterlassen die Agenten, gemäß der ACO Metaheuristik, eine Pheromonmenge auf jeder Zelle. Der Pheromonwert einer Zelle evaporiert mit fortschreitender Zeit. Aufgrund der lokalen Suchstrategie lassen sich durch die Evaporation, insbesondere zum Ende einer Abdeckung, unbesuchte Zellen schneller auffinden.

  In [Pre14] wird ein MOO Ansatz eingeführt. Hierbei wird der m-ACO_1 Algorithmus in [ASG07] genutzt, um Routen für die Agenten ermitteln zu können. Für die Routen wird die Länge minimiert und der Informationsgehalt maximiert. Eine Herausforderung stellt die Ermittlung von geeigneten Parametersets für die ACO Heuristik dar. Aufgrund der sich ändernden Systemumwelt muss das Parameterset fortlaufend angepasst werden. Bei einem ersten Ansatz [PDPH14] werden mit Hilfe der Quantitativen Graphentheorie [DES14] strukturelle Eigenschaften des Graphen quantifiziert.

- **APP-STC:**
  Die Klasse der MSTC Modelle basiert auf den Erkenntnissen der STC Modelle [GR01] und bildet für jeden Agenten in einem MARS jeweils einen Spannbaum. Die Vereinigung aller umfahrenen Spannbäume ergibt eine vollständige Abdeckung der Systemumwelt. In [AHK06] wird, basierend auf dem offline Algorithmus aus [HK05], ein MSTC Modell vorgestellt, wel-

ches stets die Entfernungen der Agenten zueinander betrachtet, um geeignete Spannbäume ähnlicher Größe ermitteln zu können. Für dieses MSTC Modell wird eine optimale Abdeckungsredundanz erzielt. Trotz der Entfernungsbetrachtung ist die totale Explorationszeit von den Startpositionen der Agenten abhängig.

Ausgehend von dem MSTC Modell in [SB08] stellen Senthilkumar und Bharadwaj in [SB10] ein erweitertes MSTC Modell vor. Mit diesem MSTC Modell wird das Ziel verfolgt, ausbalancierte Spannbäume generieren zu können, um den Einfluss der Startpositionen auf die totale Explorationszeit zu minimieren. Jeder Agent verfügt über einen individuellen Marker, um den Besuch einer Zelle zu markieren. Die Agenten wählen die nächste Zelle gemäß der Tiefensuche aus. Des Weiteren ermöglicht dieses MSTC Modell den Agenten einen redundanten Besuch von Zellen vorzunehmen, wenn dadurch eine gleichmäßigere Arbeitsaufteilung erreicht wird.

Um das Ungleichgewicht der Spannbaumgrößen reduzieren zu können, haben Zheng et al. die Klasse der Multi Robot Forest Terrain Covering Modelle eingeführt. Sie zeigen in [ZJKK05], das ausbalancierte Bäume sich positiv auf die totale Explorationszeit auswirken und die Agenten bei Multi Robot Forest Terrain Covering Modelle dazu tendieren, die Abdeckung nahe dem jeweiligen Startpunkt abzuschließen.

In [GWD12] wird ein generisches Multi Agent Voronoi TCM vorgestellt. Die Systemumwelt wird hierbei gemäß der Startpositionen der Agenten in Voronoiregionen eingeteilt. Das heißt, um jede Startposition wird eine Zelle gebildet, deren Grenzen näher zu der eigenen Startposition liegen als zu jeder anderen. Die ermittelte Zelle kann dann mit einem beliebigen Single TCM untersucht werden. In [GWD12] wird ein STC Ansatz umgesetzt. Fällt ein Agent aus, werden die Voronoiregionen neu bestimmt. Die Abdeckungsredundanz kann minimiert werden, aber die totale Explorationszeit ist stark abhängig von den Startpositionen und der räumlichen Ausprägung der Systemumwelt.

- **APP-ANN:**
  In [LY02] stellen Luo und Yang ein Multi Agent TCM vor, dass ein ANN nutzt, um eine optimierte Abdeckung zu ermöglichen. Die Systemumwelt wird in ein ANN überführt. Hierbei repräsentieren die Zellen jeweils ein Neuron und diese sind mit den jeweiligen Neuronen der Nachbarzellen verbunden. Das neuronale Netz benötigt keine Lernphase. Unbekannte Regionen wirken sich global anziehend aus und Hindernisse werden lokal vermieden. Die Aktivierungsfunktion eines Neurons ist zudem abhängig von den Positionen der Agenten und der Bewegungsrichtung, um eine Route mit minimalen Richtungsänderungen zu ermitteln. Resultierend ist das Bewegungsmuster der Agenten ähnlich der Boustrophedonbewegung.

- **APP-GA:**
  Habib et al. präsentieren in [HAS12] ein auf der MOO basierendes TCM, welches die MOO nutzt, um eine Routenoptimierung für die Agenten vorzunehmen. Zuerst wird zufällig eine vordefinierte Anzahl von Routen, entsprechend der Schwarmgröße, erzeugt. Anschließend richtet sich die Routenoptimierung nach der Minimierung der Abdeckungsredundanzen innerhalb der individuellen Route und nach der Minimierung der Überschneidungen bezüglich aller Routen untereinander. Der Graph enthält Regionen in denen die Routen eine unbekannte Systemumwelt umschließen. Ist diese resultierende Zelle kleiner als jedes Hindernis in der Systemumwelt, wird die Zelle mit der Boustrophedonbewegung abgedeckt. Die Routen-

optimierung wird mittels eines mehrkriteriellen genetischen Algorithmus durchgeführt. Die Größe der Hindernisse muss vor Beginn der Abdeckung bekannt sein. Des Weiteren kann nur dann eine vollständige Abdeckung gewährleistet werden, wenn die Vereinigung aller Zufallsplanungen eine vollständige Abdeckung entspricht. Dies wird in der Regel erst nach einer höheren Anzahl von Wiederholungen erzielt.

- **APP-MBA:**
  Komplexere Koordinationsmechanismen organisieren den Informationsaustausch mittels dynamischer Synchronisationsereignisse, welche beispielsweise durch eine Menge von Systemzuständen definiert sind. In martkbasierten Ansätzen [DS01, SYTX06] wird zu konkreten Ereignissen ein Informationsaustausch durchgeführt. Hierbei kann ein Informationsaustausch unter anderem bei Änderungen der Systemumwelt, dem Eintritt eines Agenten in das Mobile Adhoc Network (MANET) oder der Anfrage einer neuen Aufgabe ausgelöst werden. Daran anschließend wird für die nächste Aufgabe eine Auktion initiiert bei der alle Agenten innerhalb des MANET teilnehmen. Mit einer Aufgabe wird in der Regel der Besuch einer Zelle assoziiert. Der Agent mit dem geeignetsten Gebot bekommt die Aufgabe übertragen. Zu Beginn der Abdeckung kann eine Aufgabenliste ermittelt werden, die eine Reihenfolge der angebotenen Zellen vorgibt. Die Aufgabenliste wird in [Pre14] durch die Quadtree Zerlegung definiert.

  Neben der Quadtree Zerlegung untersuchen Zlot et al. in [ZSDT02] die Aufgabenreihenfolge mit Hilfe des Greedy Ansatzes nach [KTH01] und einer zufallsbasierten Auswahl. Für den Greedy-Ansatz ist die angebotene Zelle unbesucht und liegt in einer vordefinierten Nachbarschaft des Agenten der die Auktion eröffnet hat.

  Ein ähnlicher Ansatz wird in [SYTX06] verfolgt. Der Unterschied besteht in der Kommunikations- und Koordinationsstruktur sowie der Auktion. Einem Agenten wird die Rolle des Auktionators für die gesamte Abdeckung übertragen. Die Agenten können innerhalb einer definierten Reichweite kommunizieren. Damit möglichst viele Agenten im gleichen MANET sind und an der Auktion teilnehmen können, wird ein Nähemaß eingeführt. Resultierend führt der einheitliche Informationsstand zu einer Effizienzsteigerung. Dieses Nähemaß ermittelt die Abstände der Agenten zueinander und wird in den Geboten berücksichtigt. Die Agenten können mit unterschiedlichen Aufgaben an der Auktion teilnehmen. Die Auswahl der geeignetsten Aufgabe, mit der ein Agent an der Auktion teilnimmt, wird anhand einer Fitnessfunktion bestimmt. Für dieses TCM wirkt sich ebenfalls der Ausfall des Auktionators oder das Verlassen einzelner Agenten des MANET negativ auf die Robustheit aus.

In Tabelle 2.1 werden die vorgestellten TCMs entlang der im nachfolgenden Kapitel 3.2.2 aufgeführten Eigenschaften eingeordnet. Der Tabelle kann entnommen werden, dass kein TCM identifiziert werden konnte, das Unsicherheiten bezüglich der Systemumwelt betrachtet. Insbesondere für das online TCM kann eine Routenplanung meistens nur unter Unsicherheiten vorgenommen werden. Das räumliche Ziel einer Route kann hierbei in einer unbekannten Region liegen, um die vollständige Abdeckung voranzutreiben. Für eine unbekannte Systemumwelt müssen Annahmen über deren Ausprägung getroffen werden, die beispielsweise auf Rückschlüssen der bekannten Systemumwelt basieren. Erste Ansätze einer Routenplanung unter Unsicherheiten werden für die TMMs in [SAB$^+$00, SB05] angegeben und im nachfolgenden Abschnitt 2.5.2 erläutert. In Kapitel 5 wer-

den diese Ansätze aufgegriffen und um eine Schätzung der unbekannten Systemumwelt erweitert. Des Weiteren zeigt sich, dass die TCMs über keine Verfahrensweisen verfügen, um sich robust gegenüber Änderungen und Störungen, im Sinne einer dynamischen Systemumwelt, verhalten zu können. Demzufolge kann bei Änderungen der Systemumwelt keine vollständige Abdeckung garantiert werden. Hingegen werden Methoden für die TCMs implementiert, um ein robustes Verhalten, gegenüber dem Ausfall von Agenten ermöglichen zu können. Hierbei wird grundsätzlich das Verfahren verfolgt, dass Regionen, über die keine gesicherten Informationen einer Abdeckung vorliegen, wiederholt untersucht werden. Des Weiteren zeigt sich, dass TCMs, die für Agenten statisch definierte und zugewiesene Rollen festlegen, bei Störungen oder dem Ausfall von Agenten nicht robust reagieren können. Ursache hierfür ist, dass Fähigkeiten vordefinierten Agenten zugeschrieben sind, die nicht durch andere Agenten kompensiert werden können. Die Robustheit gegenüber Änderungen der Systemumwelt ist bisher kaum berücksichtigt. Die Flexibilität der TCMs ist, hinsichtlich eines angepassten Verhaltens, nur sehr eingeschränkt umgesetzt. In der Regel können die Agenten nicht auf die Gesamtheit aller möglichen Situationen trainiert werden. Aus diesem Grund setzt die Flexibilität zum einen das Identifizieren von Änderungen der Systemumwelt voraus. Zum anderen müssen die Agenten die Fähigkeit aufweisen, eine optimierte und situative Handlungsstrategie eigenständig zu entwickeln. Hierbei können Methoden der künstlichen Intelligenz genutzt werden, um ein lernendes Verhalten der Agenten zur Systemlaufzeit im Sinne einer opportunistischen Verhaltensausrichtung zu entwickeln. Unter anderem ist hierfür die Kooperation zwischen den Agenten grundlegend. Eine unzureichende Kooperation führt zudem zu einer verminderten Kohärenz. Entsprechend können beispielsweise Aufgabenkonflikte nur schwer erkannt und gelöst werden. Hinsichtlich einer optimierten Aufgabenverteilung zeigt sich, dass oftmals eine Minimierung redundanter Abdeckungen verfolgt wird. Hierbei werden unter anderem die zuvor beschriebenen Methoden des Operations Research implementiert. Beispielsweise können Methoden der Graphentheorie sowie des Natural Computing[12] implementiert werden.

2.5.2. Multi Agent Terrain Mapping Modelle

Weiterhin können multi-agent TMMs für das TCP adaptiert werden. Das TMP unterscheidet sich vom TCP dahingehend, dass nicht jede Zelle durch einen Agenten besucht werden muss. Eine Kartographierung der Zellen ist hinreichend. Entsprechend müssen die vorgestellten TMMs um eine Abdeckungsstrategie der kartographierten Zellen erweitert werden.

- **APP-GREEDY/BFS:**
  In [GHA+08] wird ein modifizierter Multiple Depth First Search Algorithmus [FTL07] genutzt, um eine Systemumwelt vollständig erkunden zu können. Die Koordination der Agenten wird von einem Masteragenten übernommen, der statisch in der Systemumwelt platziert wird und die Informationen zusammenträgt. Die erkundenden Agenten kommunizieren ausschließlich mit dem Masteragenten. Mittels Sensoren werden die Nachbarzellen um die aktuelle Zelle herum erkundet und die Agenten bewegen sich zufallsbedingt zu einer freien Zelle. Zellen die bereits schon einmal zuvor erkundet wurden, weisen eine geringere Priorität auf. Zudem wird

[12] Das Natural Computing versucht zum einen, aus Beobachtungen der Natur neue Techniken und Verfahren zu entwickeln, um Probleme aus verschiedenen Anwendungsfeldern lösen zu können. Zum anderen können mittels Simulation und Emulation verschiedene Verhaltensmuster, der Aufbau und die Struktur von natürlichen Prozessen und Lebewesen untersucht werden. Des Weiteren umfasst das Natural Computing die Erforschung von natürlichen Materialien und Strukturen, um beispielsweise verbesserte Prozessoren oder Datenträger zu entwickeln [Cas06].

Ansatz	Modell	Robustheit Umwelt	Robustheit Agent	Flexibilität	Kohärenz	Aufg.-verteilung
GREEDY/BFS	[PS90]	NEIN	JA	TW	NEIN	TW
	[YJL08]	TW	JA	TW	JA	JA
	[GF05]	NEIN	JA	TW	JA	JA
	[BRH00]	NEIN	JA	TW	NEIN	TW
	[BS03]	NEIN	JA	JA	JA	JA
	[RLNC04]	NEIN	TW	NEIN	TW	JA
STC	[GWD12]	NEIN	JA	TW	TW	JA
	[AHK06]	NEIN	TW	NEIN	TW	JA
	[SB10]	NEIN	TW	NEIN	TW	JA
	[ZJKK05]	NEIN	NEIN	NEIN	NEIN	JA
Andere	[LY02]	NEIN	JA	JA	JA	JA
	[HAS12]	NEIN	JA	NEIN	JA	JA
	[SYTX06]	NEIN	JA	JA	JA	JA
	[WLB96]	NEIN	JA	TW	JA	JA
	[Pre14]	NEIN	JA	JA	JA	JA
	Kapitel 4	TW	JA	JA	JA	JA

Robustheit (Umwelt): Robustes Verhalten gegenüber Änderungen/Störungen der Systemumwelt
Kriterien: Gewährleistung vollständiger Abdeckung; Situationsabhängige Anpassungen von Strategien
Robustheit (Agent): Robustes Verhalten gegenüber dem Ausfall von Agenten
Kriterien: Gewährleistung vollständiger Abdeckung; Strukturerhaltung
Flexibilität: Adaptiertes Verhalten aufgrund von Änderungen
Kriterien: Lernendes Verhalten; Opportunismus; Systemanpassungen zur Laufzeit; Erkennen von Änderungen
Kohärenz: Optimiertes Verhalten bezüglich der Konfliktlösung
Kriterien: Einheitliche Zielverfolgung; Erkennen von Konflikten und Redundanzen
Aufgabenverteilung: Optimierte Zuweisung von Aufgaben
Kriterien: Aufgabenverteilung basierend auf Optimierung; Vermeidung von Redundanzen

Tabelle 2.1.: Vergleich von TCMs hinsichtlich der Eigenschaften für ein Multiagent Robotic System. Mit **JA** = vollständig erfüllt, **TW** = teilweise erfüllt und **NEIN** = nicht erfüllt.

die Verbindung von der aktuellen Zelle zu der neuen Zellen gespeichert. Ist der Agent ausschließlich von bereits abgedeckten Zellen umgeben, bewegt sich der Agent auf der Route der gespeicherten Elternzellen zurück. Diese absteigende Route wird solange verfolgt, bis eine freie und nicht abgedeckte Zelle in der Nachbarschaft liegt. Bei der Rückwärtsbewegung auf der Elternroute werden die Zellen anschließend wie ein Hindernis behandelt. Demnach wird jede Zelle maximal zweimal besucht.

De Hoog et al. präsentieren in [HCV09] ein rollenbasiertes TMM. Ein Agent übernimmt die Rolle des Koordinators und ist statisch in der Systemumwelt positioniert. Die anderen Agenten untergliedern sich in die Gruppe der Erkunder und der Übermittler. Jedem Erkunder ist ein Übermittler zugeordnet. Die Exploration der Erkunder richtet sich nach dem Algorithmus in [VS08]. Jeder unbekannten Zelle wird eine Belegungswahrscheinlichkeit hinsichtlich eines Hindernisses zugeordnet. Der unbekannten Systemumwelt kann somit ein mittlerer Informationsgehalt zugeordnet werden. Gemäß des Greedy Ansatzes bewegen sich die Agenten in Richtung der unbekannten Systemumwelt mit dem höchsten Informationsgehalt. Der Informationsgehalt repräsentiert die Wahrscheinlichkeit, dass keine Hindernisse vorhanden sind.

Die Übermittler folgen den zugehörigen Erkundern bis zu einem Treffpunkt. Dieser Treffpunkt wird dynamisch festgelegt und es erfolgt ein Informationsaustausch. Der Übermittler kehrt zurück zum Koordinator und übermittelt die hinzugewonnen Informationen der Systemumwelt. Anschließend kehren beide Agenten zum vereinbarten Treffpunkt zurück. Die Treffpunkte verlagern sich mit zunehmender Exploration weiter vom Koordinator weg.

In [JT08] wird ein Visibility TMM für polygonale Systemumgebungs- und Hindernisstrukturen vorgestellt. Die Systemumwelt wird in Zellen untergliedert, welche durch Navigationssensoren des Fernbereiches gebildet werden. Die Koordination der Agenten wird mittels von Prioritäten organisiert. Zu Beginn der Exploration wird jedem Agent eine Priorität zugewiesen. Der Agent mit der höchsten Priorität verteilt die Informationen der Systemumwelt innerhalb der Kommunikationsreichweite. Anschließend untersucht der Agent vollständig die sichtbare Systemumwelt um die eigene Position herum. Die einsehbare Zelle wird durch Hindernisse und der äußeren, räumlichen Begrenzung der Systemumwelt eingeschränkt. Die angrenzende und somit nicht einsehbare Systemumwelt wird durch Sichtgrenzen definiert. Mit absteigender Priorität werden weitere Aufgaben an die anderen Agenten zugewiesen. Zuerst werden die erkundeten Hindernisse mit Navigationssensoren des Nahbereiches untersucht und danach bewegen sich die Agenten zu den Mittelpunkten der Sichtgrenzen. Die Agenten führen wiederholt vollständige Untersuchungen der angrenzenden Systemumwelt durch. Sind für die aktuelle Zelle alle Aufgaben verteilt, folgen die Agenten genau dem Agenten mit der niedrigsten Priorität. Ändert sich die Anzahl der Agenten innerhalb der Kommunikationsreichweite, dann wird die Reihenfolge gemäß der Priorität neu bestimmt.

- **APP-MBA:**
  In [SAB+00] wird ein TMM vorgestellt, welches auf einem Auktionsalgorithmus basiert. Wenn ein Agent über keine Aufgabe mehr verfügt, wird eine Auktion eröffnet und es werden Gebote für unterschiedliche Zellen abgegeben. Die Zellen sind jeweils frei und liegen am äußerten Rand der individuellen Sensorreichweite. Das Gebot für eine Zelle setzt sich zusammen aus dem geschätzten Informationszugewinn und den Kosten der Route, um die Zelle zu erreichen. Der Informationszugewinn berücksichtigt die Erkundungen anderer Agenten und nimmt eine Schätzung der Belegungswahrscheinlichkeit aufgrund vorhandener Informationen über die Systemumwelt vor. Die Auktionen werden zentral von einer übergeordneten Instanz gesteuert und es gibt keine Kommunikationseinschränkungen.

  Sariel und Balch ermitteln in [SB05] die Aufgabenreihenfolge gemäß einer vordefinierten Priorität und den geschätzten Kosten. Die Priorität richtet sich danach, ob Aufgaben für eine effizientere Ausführung getauscht oder in der Reihenfolge als nächstes bearbeitet werden müssen. Alle anderen Aufgaben weisen eine niedrigere Priorität auf. Die Autoren empfehlen bei Wissensänderungen, bezüglich der Umwelt, beispielsweise durch ein Hinzukommen von neuen Aufgaben und Änderungen bei der Entfernungsabschätzung sowie Kostenabschätzung, eine erneute Auktion der aktuell zu bearbeitenden Aufgaben durchzuführen. Mit Hilfe von Konfliktverhütungsmaßnahmen und wiederholten Auktionen werden Unsicherheiten aufgrund von Kommunikationsstörungen entgegengewirkt.

Modellierungsansatz			Eigenschaften der Modellierung
Optimierung	OPT	SOO	• Eindeutige Lösung • Vereinfachte Problemstruktur
		MOO	• Lösungsmenge bedarf Auswahlentscheidung • Anwendungsnahe Problemstruktur • Berücksichtigung verschiedener Optimierungsziele • Flexible Betrachtung von Optimierungszielen möglich • Erhöhter Rechenaufwand
	UNC	TYPE I	• Approximation von unvollständigen Informationen • Robustere Lösung • Verbessertes Wahrnehmung von Inkonsistenzen
		TYPE II	• Verbesserte Reaktion auf Fehler • Verbesserte Konfliktlösung
Agent	CHAR	HOM	• Aufgabenübernahme durch alle Agenten • Komplexere Agenten da alle Fähigkeiten implementiert sein müssen
		HET	• Aufgabenübernahme durch Agenten nur eingeschränkt möglich • Spezialisierte Agenten mit Teilmenge von Fähigkeiten
	ORG	CEN	• Eindeutiger Entscheidungsträger • Entscheidungsträger muss mit Informationen versorgt werden • Optimierung auf Grundlage aller zur Verfügung stehenden Informationen • Keine Problemlösung bei Ausfall des Entscheidungsträgers • Erhöhte Rechenkapazitäten des Entscheidungsträgers notwendig • Weniger robust gegenüber Kommunikationsstörungen
		DEC	• Jeder Agent ist Entscheidungsträger • Robuster gegenüber Umweltveränderungen und Kommunikationsstörungen • Robuster gegenüber dem Ausfall von Agenten
		HYB	• Dynamische Auswahl des Entscheidungsträgers • Optimierung auf Grundlage aller zur Verfügung stehenden Informationen • Problemlösung kann auf mehrere Agenten verteilt werden • Robuster gegenüber Umweltveränderungen und Kommunikationsstörungen • Robuster gegenüber dem Ausfall von Agenten

Tabelle 2.2.: Auswirkungen verschiedener Modellierungsansätze auf die Eigenschaften eines TCM (Teil 1).

2.5.3. Eine Entscheidungsunterstützung zur Modellierung von Terrain Covering Modellen

In den Tabellen 2.2 (Teil 1) und 2.3 (Teil 2) werden die grundlegenden Auswirkungen der Modellierungsansätze gemäß der Taxonomie auf die Eigenschaften der TCMs und ferner auf die Abdeckung einer Systemumgebung zusammengefasst.

In Abhängigkeit der Anforderungen und Besonderheiten, die an die übergeordnete Anwendung gestellt werden, muss bei der Implementierung eines MA(R)S eine geeignete Auswahl der verschiedenen Modellierungsansätze vorgenommen werden. Nachfolgend werden Modellierungsansätze dargestellt, die einen positiven Einfluss auf die Ausprägung bestimmter Eigenschaften eines MA(R)S nehmen.

- **Robustheit:**
  Neben der Betrachtung von Unsicherheiten in Form der Typen I und II, tragen vor allem Modellierungen zu einer verbesserten Robustheit bei, die ein autonomes Verhalten jedes einzelnen Agenten ermöglichen. Der Ausfall homogener Agenten wirkt sich oftmals nur auf die Gesamtperformance aus. Teilaufgaben können von anderen Agenten mit gleichem Fähigkeitsspektrum übernommen werden. Eine dezentrale oder hybride Organisationsform bietet den

Modellierungsansatz			Eigenschaften der Modellierung
Sensor	CTOP	BROAD	• Informationen werden immer an alle Agenten innerhalb des Kommunikationsradius gesendet • Höherer Informationsfluss • Agenten müssen sich einander nicht kennen
		ADD	• Informationen werden gezielt an bestimmte Agenten innerhalb des Kommunikationsradius gesendet • Geringerer Informationsfluss • Sender muss Empfänger kennen
		GRAPH	• Informationen können mittels anderer Agenten über den individuellen Kommunikationsradius hinweg gesendet werden • Besserer Informationsstand • Sender muss Empfänger kennen
	CRANGE	NONE	• Indirekte Kommunikation mittels in der Systemumgebung platzierten Markern • Logisches Schließen und Antizipieren ist notwendig • Informationen können veraltet sein • Agenten müssen in der Lage sein, selbstständig Entscheidungen treffen zu können • Notwendigkeit von Sensoren
		NEAR	• Begrenzte Kommunikationsreichweite (z. B. Sichtweite) • Schlechterer Informationsstand bei dislozierten Agenten • Agenten müssen in der Lage sein, selbstständig Entscheidungen treffen zu können • Geringere Wahrscheinlichkeit, dass Informationen durch andere abgehört werden können
		INF	• Unbegrenzte Kommunikationsreichweite • Besserer Informationsstand • Höhere Wahrscheinlichkeit einer optimierten Entscheidungsfindung bei Berücksichtigung aller Agenten • Höhere Wahrscheinlichkeit, dass Informationen durch andere abgehört werden können
	SENSE	NEAR	• Höhere Messgenauigkeit • Geringere Reichweite • Geringere Wahrscheinlichkeit, dass Sensormessungen durch andere abgehört werden können
		WIDE	• Geringere Messgenauigkeit mit zunehmender Entfernung • Höhere Reichweite • Höhere Wahrscheinlichkeit, dass Sensormessungen durch andere abgehört werden können
		HYB	• Ausstattung von Nah- und Fernbereichssensoren • Höhere Messgenauigkeit im Nahbereich • Geringere Messgenauigkeit mit zunehmender Entfernung • Notwendigkeit einer höheren Rechenkapazitäten und Platzbedarf • Höhere Wahrscheinlichkeit, dass Sensormessungen durch andere abgehört werden können
Umgebung	DECOMP	EXACT	• Höhere Genauigkeit in der Befahrung und Kartographie • Inkrementelle Befahrung einer Umgebung • Zellzerlegung aufgrund von Sensormessungen und Umgebungsstrukturen
		SEMIAPPROX	• Höhere Genauigkeit in der Befahrung und Kartographie • Inkrementelle Befahrung einer Umgebung • Zellzerlegung aufgrund von Sensormessungen und Umgebungsstrukturen
		APPROX	• Geringere Genauigkeit in der Befahrung und Kartographie • Zellzerlegung aufgrund von Sensorreichweite und notwendiger Genauigkeit • Vereinfachte Zuweisung und Beschreibung von Zellen möglich • Hierarchische Speicherstruktur von Zellen • Höherer Speicherbedarf mit zunehmender Auflösung

Tabelle 2.3.: Auswirkungen verschiedener Modellierungsansätze auf die Eigenschaften eines TCM (Teil 2).

Vorteil, dass jeder Agent in der Lage, ist selbstständig Entscheidungen zu treffen. Eine möglichst große Kommunikationsreichweite wirkt sich grundsätzlich positiv auf einen einheitlichen Informationsstand aus. Hierbei kann eine geeignete Kommunikationstopologie die Reichweite durch das Weiterleiten von Informationen erhöhen. Resultierend werden die Informationen möglichst vieler Agenten in der Entscheidungsfindung berücksichtigt. Ferner können Inkonsistenzen aufgrund falscher Informationen besser erkannt und behoben werden. Eine sichere Navigation wird mit Hilfe von Nahbereichssensoren mit hoher Messgenauigkeit sichergestellt. Darüber hinaus bieten Fernbereichssensoren die Möglichkeit, frühzeitig zusätzliche Informationen bezüglich der Systemumwelt in eine optimierte Entscheidungsfindung einfließen zu lassen.

- **Flexibilität:**
  Ähnlich der Robustheit wirkt sich die Ausprägung eines autonomen Verhaltens jedes einzelnen Agenten positiv auf die Flexibilität aus. Insbesondere können Agenten dadurch auf Änderungen der Systemumwelt oder Teilziele flexibel und angepasst reagieren. Die Bereitstellung aller Informationen an alle Agenten verbessert hierbei den Informationsstand und trägt dazu bei, dass ein abgestimmtes Verhalten auf neue Situationen ausgeprägt wird.

- **Kohärenz:**
  Eine gemeinsame Zielverfolgung und Konfliktlösung bedarf eines kontinuierlichen Informationsflusses, damit möglichst alle Informationen zur Verfügung stehen. Ferner wirkt sich die Prüfung und Bewertung der Informationen mit Hilfe eines zentralen Entscheidungsträgers (DM[13]) positiv auf das Wahrnehmen von Ziel- und Planungskonflikten sowie Inkonsistenzen aus. Mit Hilfe von Nahbereichs- und Fernbereichssensoren können möglichst viele Informationen gesammelt und falsche Informationen mit neuen Sensormessungen überprüft werden.

- **Aufgabenverteilung:**
  Homogene Agenten ermöglichen eine verbesserte Aufgabenverteilung, weil potenziell alle Agenten für die Bearbeitung einer Teilaufgabe ausgewählt werden können. Um Redundanzen zu vermeiden, müssen möglichst alle gesammelten Informationen dem DM vorliegen. Eine große Kommunikationsreichweite verbessert den Informationsfluss und trägt zu einem einheitlichen Informationsstand bei. Ein zentraler DM bietet zusätzlich den Vorteil, dass stets alle zuvor getroffenen Entscheidungen berücksichtigt werden können. Die Aufgabenzuweisung kann durch eine eindeutige Unterteilung der Systemumgebung in Zellen vereinfacht werden.

Neben den grundlegenden Eigenschaften eines MA(R)S können ganzheitliche Szenare betrachtet werden für die eine Vielzahl von Eigenschaften vorliegen müssen, um eine Abdeckung sicherstellen zu können.

- **Unstrukturierte Systemumgebung:**
  Eine stark unstrukturierte Systemumgebung ist durch ein herausforderndes Terrain mit Hindernissen und unscharfen Umgebungsstrukturen gekennzeichnet. Für diese Systemumgebung müssen die Agenten mit Sensoren ausgestattet werden, die eine hohe Messgenauigkeit aufweisen. Die Implementierung von verschiedenen Sensortypen im MA(R)S, die beispielsweise eine Lasermessung oder ein taktiles Messen durchführen, ermöglicht die Anpassung an unterschiedliche Anforderungen und Notwendigkeiten hinsichtlich der Umgebungsstrukturen.

[13] Engl. Decision Maker

Die erhöhte Komplexität der Agenten, resultierend auf der Möglichkeit die Umgebungsstrukturen bewältigen zu können, bedarf unter Umständen heterogene Agenten, die Fähigkeiten für das MA(R)S bereitstellen. Eine geeignete Kommunikationstopologie, die das Weiterleiten von Informationen über Agenten ermöglicht, erhöht die Wahrscheinlichkeit, dass der Informationsfluss trotz Hindernissen erhalten bleibt. Mögliche Unsicherheiten bezüglich der Systemumwelt und den Agenten können mit einer robusten Optimierung einbezogen werden.

- **Agilität:**
  Anwendungsfelder, die eine hohe Agilität der Agenten voraussetzen, stellen besondere Anforderungen an die Implementierung und Nutzung von Agenten. Hierbei liegt für die Agenten eine leichtgewichtige Implementierung vor. Grundfähigkeiten, wie beispielsweise einen Informationsaustausch oder ein Positionsbestimmung werden bereitgestellt. Einsatzspezifische Fähigkeiten können modular erweitert werden. Die Agenten senden alle Informationen zu einer zentralen, leistungsfähigen Steuerungsstation. Die Steuerungsstation aggregiert alle Informationen zu einem fusionierten Lagebild und führt eine Entscheidungsfindung durch. Die Anweisungen der Steuerungsstation enthalten alle zur Aufgabenerfüllung notwendigen Informationen, um die Anforderungen an die Rechenleistung zu reduzieren. Ferner wird durch eine leichtgewichtige Implementierung der Agenten eine schnelle Inbetriebnahme unterstützt.

- **Hohe Bedrohungslage:**
  Eine hohe Bedrohungslage stellt besondere Anforderungen an die Robustheit gegenüber dem Ausfall von Agenten. Aus diesem Grund eignen sich homogene Agenten, die Aufgaben ausgefallener Agenten übernehmen können. Aufgrund von Gegenmaßnahmen und einer möglichen Aufklärung durch gegnerische Kräfte ist die Kommunikation zwischen den Agenten zu reduzieren. Informationen werden selektiv verteilt. Zudem wirkt sich eine dezentrale Organisation positiv auf eine reduzierte Kommunikation aus. Das eigene Verhalten wird auf der Grundlage von Beobachtungen anderer Agenten selbstständig bestimmt. Mit Hilfe von passiven Sensoren kann die Wahrscheinlichkeit einer Aufklärung der Agenten reduziert werden.

In Tabelle 2.4 ist eine Zusammenfassung beigefügt, die den positiven Einfluss einzelner Modellierungsansätze auf die Eigenschaften des MA(R)S darstellt. Des Weiteren sind exemplarisch verschiedene Szenare aufgeführt. Resultierend dient die Übersicht als Entscheidungsunterstützung für die Realisierung und den Einsatz eines MA(R)S für Anwendungen im Rahmen des TCP.

2.6. Forschungsfragen und Untersuchungsgegenstände

Die Untersuchung und Gegenüberstellung bestehender Multi Agent Online TCMs in Kapitel 2.5 zeigt, dass verschiedene Ausprägungen bezüglich der Agenten und deren Sensorsystemen publiziert worden sind. Ferner existieren Ansätze, um eine zielorientierte Kommunikation und Koordination zu implementieren. Mit der Ausnahme weniger TCMs [SYTX06, HAS12, Pre14] werden Optimierungsansätze verfolgt, die ausschließlich eine Zielfunktion bei der Ermittlung einer vollständigen Abdeckung betrachten. Hierbei nimmt die Minimierung der Abdeckungsredundanzen einen hohen Stellenwert ein und führt gleichzeitig zur Minimierung der Abdeckungszeit.

Modellierungsansatz			Robustheit: *Umwelt*	*Agent*	*Flexibilität*	*Kohärenz*	*Aufgaben-verteilung*	*Unstrukturierte Systemumgebung*	*Agilität*	*Hohe Bedrohungslage*
Optimierung	OPT	SOO								
		MOO			+	+		+		+
	UNC	TYPE I	+	+	+	+		+		
		TYPE II	+	+	+	+		+		
Agent	CHAR	HOM		+			+			+
		HET						+	+	
	ORG	CEN				+	+		+	
		DEC	+	+	+			+		+
		HYB	+	+		+	+	+		+
Sensor	CTOP	BROAD	+	+	+	+	+	+	+	
		ADD							+	+
		GRAPH	+		+		+	+		
	CRANGE	NONE		+				+		+
		NEAR						+		+
		INF	+	+		+	+	+	+	
	SENSE	NEAR	+	+		+		+	+	
		WIDE	+			+		+		
		HYB	+			+		+		
Umgebung	DECOMP	EXACT						+		
		SEMIAPPROX					+	+		
		APPROX					+			

Tabelle 2.4.: Entscheidungsunterstützung zur Modellierung eines MA(R)S. Positive Einflussnahme von einem Modellierungsansatz hinsichtlich einer Eigenschaft ist durch ein (+) gekennzeichnet.

In realen Anwendungen zeigt sich, dass mobile Robotersysteme aufgrund der Größe und Kapazitäten über ein begrenztes Energieniveau verfügen. Resultierend ist ein effizientes Verhalten notwendig, um Energieeinsparungen und ferner eine längere Betriebsdauer zu ermöglichen. Für ein effizientes Verhalten müssen neben der Routenlänge zusätzliche Informationen wie beispielsweise Bodenbeschaffenheit, Steigungs- und Neigungsgrad oder Luft- und Wasserströmungen betrachtet werden, die einen Einfluss auf den Energieverbrauch oder Zeitbedarf nehmen. Daneben kann ebenfalls eine Betrachtung der Bedrohungslage oder allgemeinen Befahrbarkeit für eine zielorientierte Auftragserfüllung einen positiven Einfluss nehmen. Insbesondere für die Klasse von Online TCMs liegen diese Informationen aus der Systemumgebung gar nicht oder nur unvollständig vor. Im Verlauf einer Abdeckung werden durch die Agenten kontinuierlich neue Informationen gesammelt. Mit Hilfe dieser Informationen können Abschätzungen für noch unbekannte Regionen der Systemumgebung vorgenommen werden.

Das TCM nach Sariel und Balch [SB05] versucht, mit Hilfe eines vollständigen Informationsaustausches zwischen allen Agenten Inkonsistenzen aufzudecken. Erst nachdem ein einheitlicher Informationsstand hergestellt ist, werden Entscheidungen zur weiteren Abdeckung getroffen. Simmons et. al stellt in [SAB+00] einen erweiterten Ansatz vor. Es wird mittels einer Wahrscheinlichkeitsverteilung die Ausprägung [frei|blockiert|unbekannt] einer Zelle beschrieben. Die Routenplanung wird über eine Optimierung der gewichteten Zielfunktion, zusammengesetzt aus der Routenlänge zur Zielzelle und dem geschätzten Informationszugewinn, durchgeführt. Dass eine Betrachtung mehrerer Optimierungsziele einen positiven Einfluss auf die Effizienz und Zielerreichung nehmen kann, wird in [Pre14] gezeigt.

Zusammenfassend gewinnt die Betrachtung von Unsicherheiten, mit immer komplexeren Anwendungen und dem Einsatz von autonomen Agenten in einer unbekannten Systemumgebung, an Bedeutung. Im Vergleich zu einer robusten Koordination und Aufgabenverteilung ist die robuste Routenplanung für das TCP noch nicht hinreichend genug untersucht worden. Fehlende Untersuchungen lassen sich unter Umständen darauf zurückführen, dass autonome Robotersysteme bisher noch nicht umfassend in komplexen Anwendungen eingesetzt werden. Vielmehr werden unbemannte Robotersysteme eingesetzt. Ursache hierfür ist die Fehleranfälligkeit bei autonomen Robotersystemen. Vor allem können Störungen in der Bewegung und Sensormessung auftreten [CM03b, CM05, MKP+08, Mur14]. Ferner stellt eine robuste Routenplanung ein rechenintensives Problem dar, welches vor allem von kleinen Robotersystemen aufgrund der begrenzten Rechenkapazität nicht in hinreichender Zeit gelöst werden können. Des Weiteren gefährden mögliche Fehlentscheidungen von autonomen Robotersystemen die Zielerreichung.

Resultierend lautet die übergeordnete Forschungsfrage dieser Dissertation:

Forschungsfrage 1. : *Wie können Unsicherheiten bezüglich der Systemumgebung geeignet in die Optimierung des Terrain Covering Problems einbezogen werden?*

Ausgehend von ersten Ergebnissen [SAB+00] werden die Kosten, welche aus einer Abdeckung einer Zelle resultieren, für unbekannte Regionen der Systemumgebung approximiert. Hierbei wird die Annahme getroffen, dass die Kostenstruktur benachbarter Zellen ähnlich ist. Die Annahme trifft auf eine Vielzahl von verschiedenen Anwendungen zu. Bei Unterwassermissionen wie der Suche nach Seeminen sind die Strömungsverhältnisse für benachbarte Zellen ähnlich. Des Weiteren bestimmen bei Landmissionen die Bodenbeschaffenheit und das Profil im Wesentlichen die Kosten

zur Abdeckung der Systemumgebung. Hierbei kann ebenfalls die Kostenstruktur unbekannter, benachbarter Zellen mit einer bestimmten Wahrscheinlichkeit approximiert werden. Mit zunehmender Entfernung von bekannten Zellen nimmt die Wahrscheinlichkeit, dass die Zellen eine ähnliche Kostenstruktur aufweisen, ab. Es werden Untersuchungen anhand verschiedener Systemumgebungen durchgeführt. Hierbei werden die Auswirkungen auf die Effizienz und Robustheit mittels verschiedener Metriken überprüft.

Die Approximation unbekannter Zellen wird genutzt, um eine robuste Routenplanung vornehmen zu können. Es existieren Ansätze des SSPP, die eine robuste Route ermitteln. Es können jedoch nur eingeschränkte Aussagen über die Vergleichbarkeit der Robustheit verschiedener Routen gemacht werden. Aus diesem Grund wird die Forschungsfrage

Forschungsfrage 2. : *Wie kann für das Terrain Covering Problem eine geeignete Lösung des Stochastic Shortest Path Problems angegeben werden?*

untersucht.

In der Literatur sind keine Untersuchungen, bezüglich den Auswirkungen einer optimierten Routenplanung unter Unsicherheiten, für das TCP bisher publiziert. Für das TCP nimmt die Vergleichbarkeit von Routen einen hohen Stellenwert ein. Insbesondere wenn das TCP mit Hilfe von mehreren Agenten bearbeitet wird, müssen die Routen, im Sinne einer koordinierten, effizienten und robusten Entscheidungsfindung, in ein Verhältnis zueinander gesetzt werden können. Die Mehrkriterielle Optimierung bietet mit dem Pareto-Konzept eine Möglichkeit, aus einer mehrdimensionalen Lösungsmenge die dominierenden Alternativen bestimmen zu können. Mit dieser Dissertation wird erstmals anhand des SSPP gezeigt, wie ein einkriterielles Optimierungsproblem unter Unsicherheiten in ein mehrkriterielles Optimierungsproblem überführt werden kann. Die Transformation in das schwierigere Problem führt zu dem Vorteil, dass eine Vergleichbarkeit der Lösungen entsprechend der Robustheit gegeben ist.

2.7. Zusammenfassung

Zusammenfassend kann mit der Tabelle 2.5 eine Einordnung der TCMs und TMMs anhand der zuvor definierten Taxonomie durchgeführt werden. Hierbei wird die Systemumwelt für die aufgeführten Modelle in Form eines repräsentativen Graphen abgebildet. In der Regel stellt jede Zelle einen Knoten dar und die Kanten entsprechen den direkten Verbindungen zwischen den Knoten. Es zeigt sich, dass eine Vielzahl von verschiedenen Optimierungsverfahren genutzt werden, um eine geeignete Lösung für das TCP ermitteln zu können. Es werden häufig intuitive DFS und BFS Heuristiken genutzt, die jedoch unter bestimmten Umständen kostenintensive Routen verursachen können. Insbesondere für BFS Heuristiken können Hindernisstrukturen zu einer redundanten Abdeckung führen, weil die Agenten zur jeweiligen Ausgangszelle zurückkehren müssen. Zum anderen können für Greedy Heuristiken zum Ende der Abdeckung einzelne unbesuchte Zellen disloziert in der Systemumwelt verteilt sein [Pre11]. Aufgrund zusätzlicher Informationen, die in Form von Pheromonwerten gespeichert werden, können ACO Implementierungen vorteilhafter sein. Gegenüber dem klassischen DFS und BFS können mit Hilfe der Pheromonevaporation und -diffusion unbesuchte Zellen schneller aufgefunden werden [KL01]. Zudem können zusammenhängende Regionen

	Modell	Ansatz	OPT	UNC	AGENT	ORG	CRANGE	CTOP	SENSE	ENV	DECOMP
TCM	[PS90]	GREEDY	SOO	-	HOM	DEC	NONE	BROAD	NEAR	OBS_R	APPROX
	[YJL08]	GREEDY	SOO	TYPE II	HOM	DEC	INF	BROAD	NEAR	OBS	SEMIAPPROX
	[GF05]	GREEDY/BFS	SOO	-	HOM	DEC	INF	BROAD	NEAR	OBS_R	-
	[BRH00]	GREEDY/BFS	SOO	-	HOM	CEN	-	-	NEAR	OBS_R	EXACT_{SL}
	[BS03]	BFS	SOO	-	HOM	HYB	NONE	BROAD	NEAR	OBS	APPROX
	[RLNC04]	BFS	SOO	-	HOM	DEC	INF	ADD	NEAR	OBS	EXACT_{SL}
	[GWD12]	STC/VPC	SOO	-	HOM	DEC	INF	BROAD	NEAR	OBS	EXACT_{VP}
	[AHK06]	STC	SOO	-	HOM	DEC	INF	BROAD	NEAR	OBS_R	-
	[SB10]	STC	SOO	-	HOM	DEC	NONE	BROAD	NEAR	OBS_R	APPROX
	[ZJKK05]	STC	SOO	-	HOM	CEN	-	-	NEAR	OBS_R	APPROX
	[LY02]	ANN	(SOO)	-	HOM	DEC	NONE	BROAD	NEAR	OBS	SEMIAPPROX
	[HAS12]	GA	(MOO)	-	HOM	CEN	INF	ADD	NEAR	OBS_R	SEMIAPPROX
	[SYTX06]	MBA	MOO	-	HOM	DEC	NEAR	GRAPH	NEAR	OBS_R	APPROX
	[WLB96]	ACO	SOO	-	HOM	DEC	NONE	BROAD	NEAR	OBS_R	APPROX
	[Pre14]	ACO/MBA	MOO	-	HOM	HYB	NEAR	GRAPH	NEAR	OBS_R	APPROX
	Kapitel 4	ACO/MBA	MOO	TYPE I	HOM	HYB	NEAR_O	GRAPH	NEAR	OBS_R	APPROX
TMM	[HCV09]	GREEDY	SOO	-	HET	CEN	NEAR_O	ADD	NEAR	OBS_R	(EXACT_{VB})
	[GHA+08]	GREEDY	SOO	-	HET	CEN	INF	ADD	NEAR	OBS_R	APPROX
	[JT08]	GREEDY/BFS	SOO	-	HOM	DEC	NEAR	BROAD	HYB	OBS_P	EXACT_{VB}
	[SAB+00]	MBA	SOO	TYPE I & II	HET	CEN	INF	ADD	WIDE	OBS	EXACT_{VB}
	[SB05]	MBA	SOO	(TYPE I & II)	HOM	DEC	NEAR	BROAD	NEAR	OBS_R	APPROX

Tabelle 2.5.: Vergleich von TCMs und TMMs hinsichtlich der Modellierungen und Strukturmerkmale.

gezielter abgedeckt werden. Bezüglich einer optimierten Aufgabenverteilung sind globale Informationen gegenüber ausschließlich lokalen Informationen vorzuziehen. Mit Hilfe der STC Algorithmen können minimale Spannbäume in Graphen gefunden werden, die als Routen fungieren. Eine Vielzahl von Systemumwelten kann somit unter minimaler Abdeckungsredundanz vollständig erkundet werden. Hierbei müssen die Agenten über keine direkte Kommunikation verfügen [SB08]. Hybride TCMs können eine effiziente Abdeckung gewährleisten [GWD12]. Eine Möglichkeit für hybride TCMs stellt die Unterteilung des TCP in das Aufgabenverteilungsproblem und das Routenplanungsproblem dar. Die MBA Heuristiken können für die Verteilung von Aufgaben genutzt werden. Hierbei wirken sich die MBA Heuristiken, in Übertragung an einen ökonomischen Markt und deren Marktteilnehmer, positiv auf die Flexibilität, Anpassungsfähigkeit und Robustheit gegenüber Änderungen aus. Ursache hierfür ist zum einen die Fähigkeit der Selbstorganisation und der dezentralen Organisation der Agenten [DS01]. Zum anderen können, mittels eines marktbasierten Regelwerkes, dynamische Anpassungen an den Kosten- und Nutzenfunktionen der Agenten durchgeführt werden [KYA07]. Infolge von Änderungen bezüglich der Systemumwelt können Aufgaben neu verteilt werden [SB05]. Dennoch sind die Abdeckungsredundanzen von der Aufgabenreihenfolge abhängig. Für die Routenplanung zum Aufgabenziel, können eine Vielzahl von Heuristiken zur Anwendung kommen. Aufgrund der Trennung in die Teilprobleme der Aufgabenverteilung und Routenplanung können Anpassungen des TCM in Abhängigkeit zum Anwendungsgebiet durchgeführt werden, ohne das die Grundstruktur und Organisation verändert werden muss.

Für das TCP werden ausschließlich TCMs betrachtet, die eine strukturierte Koordination der Agenten aufweisen. Hierbei resultiert aus einer schwach ausgeprägten Koordination in der Regel eine geringere Komplexität und zugleich gesteigerte Robustheit des MAS. Nachteilig wirkt sich hingegen die geringe Effizienz aus. Aufgaben werden unter Umständen mehrfach bearbeitet oder die Agenten behindern sich gegenseitig. Zusammenfassend eignet sich eine stark strukturierte Koordination insbesondere für ein dynamisches und komplexes TCP [INS00]. Des Weiteren zeigt sich, dass überwiegend eine dezentrale Organisationsstruktur genutzt wird. Hierbei beruht die Entscheidungsfindung auf den Informationen, die indirekt aus der Systemumwelt, im Sinne der Stigmergie [Dor92, WLB96], beziehungsweise mittels stationärer Kommunikationseinrichtungen [BS03], bezogen oder direkt zwischen den Agenten im MANET ausgetauscht werden [WM04, YJL08, Pre14]. Jeder Agent ermittelt die nachfolgenden Aktionen eigenständig. Das TCM ist demnach gegenüber dem Ausfall und Hinzukommen von Agenten sowie Kommunikationsstörungen robust. Für ein dezentrales MAS kann sich der inkonsistente Informationsstand der Agenten auf die Effizienz negativ auswirken [SYTX06]. Aus diesem Grund muss zum einen der kontinuierliche Informationsaustausch organisiert werden. Zum anderen sollten Ereignisse festgelegt werden, zu denen ein zusätzlicher Informationsaustausch sinnvoll ist [SB05]. Die Kommunikationsstruktur kann sich ebenfalls positiv auf den einheitlichen Informationsstand auswirken. Hierbei wird teils eine Kommunikationsreichweite über die gesamte Systemumwelt realisiert. Für Anwendungen mit Luftfahrzeugen ist diese Annahme sinnvoll. Für Land- und vor allem Unterwasseranwendungen sollte die Kommunikationsreichweite der Systemumwelt angepasst werden. Um dennoch möglichst viele Agenten erreichen zu können, bietet sich neben der Broadcaststruktur [GWD12, YJL08] die Umsetzung eines dynamischen MANET an [SYTX06, Pre14]. Informationen können direkt und indirekt über vermittelnde Agenten ausgetauscht werden. Aus diesem Grund können Agenten miteinander kommunizieren ohne das sie sich innerhalb der Kommunikationsreichweite befinden. Der Aufbau von einem dynamischen MANET und die Berücksichtigung von Hindernisstrukturen wird nur von wenigen Modellen [DS03a, SB05, Pre14] umgesetzt.

Zusätzlich werden in den Modellen nahezu ausschließlich homogene Agenten eingesetzt, um die Fähigkeiten möglichst redundant zwischen den Agenten zu verteilen. Demzufolge kann die Abdeckung auch von einem Agenten erfolgreich beendet werden. Ein heterogenes MAS hingegen verfügt, aufgrund der verschiedenen Fähigkeiten und Ausprägungen der Agenten, über eine verbesserte Anpassungsfähigkeit in einer dynamischen Systemumwelt. Zudem können Änderungen von Agenten individuell vorgenommen werden, um neue und unvorhersehbare Anforderungen der Systemumwelt zu erfüllen [INS00]. Des Weiteren weist ein heterogenes MAS in der Regel eine komplexere Gesamtstruktur auf. Insbesondere das Aufgabenverteilungsproblem stellt sich in Folge der zusätzlichen Nebenbedingungen als komplexer dar [YJA13].

Die vorgestellten TCMs ermöglichen den Einsatz von Agenten in einer Systemumwelt mit Hindernissen. Die Systemumwelt wird hierbei oftmals in eine approximative und semi-approximative Zellstruktur transformiert. Jedes TCM, das eine rechteckig ausgeprägte Systemumgebung voraussetzt, kann näherungsweise in einer realen Systemumwelt eingesetzt werden [GR01]. Approximative Zellen, die nicht vollständig frei von Hindernissen sind, werden hierbei nicht abgedeckt. Die approximative Zellstruktur biete den Vorteil, dass die Identifizierung von Zellen hinsichtlich von Hindernissen eindeutig ist und unabhängig des aktuellen Standortes durchgeführt werden kann. Zudem ist die Speicherverwaltung der Zellzustände einfach umsetzbar, obwohl die Speicheranforderung in Abhängigkeit der Auflösung stark zunehmen kann. Die Bestimmung der kürzesten Route kann mit Hilfe etablierter Algorithmen realisiert werden. Hierfür besteht jedoch auch die Notwendigkeit einer exakten Positionsbestimmung [TB96]. Die exakte Zerlegung der Systemumwelt und Repräsentation kann in verschiedenen Ausprägungen umgesetzt werden [Cho01]. Unscharfe Sensormessungen wirken sich für die Identifikation von Zellen negativ aus. Folglich kann eine optimierte Abdeckung für die größeren und zusammenhängenden Regionen ermittelt werden. Zudem wird eine geringe Speicherkapazität für die Verwaltung der Zellen benötigt. Die Speicheranforderung ist hierbei abhängig von der Komplexität der Systemumwelt [TB96].

In der Literatur ist nur das TCM nach [Pre14] bekannt, das eine echte mehrkriterielle Optimierung umsetzt. Für dieses TCM können unterschiedliche Optimierungsziele parallel und online zur Abdeckung betrachtet werden. Hierfür basiert die Routenplanung auf einer Heuristik der mehrkriteriellen Ameisenalgorithmen (MACO[14]) nach [ASG07]. Im Mittelpunkt steht die Minimierung der Routenlänge sowie die Maximierung des Informationszugewinns der Routen. Somit wird für das TCM die totalen Explorationszeit und die redundante Abdeckung minimiert. Eine Erweiterung des MACO Algorithmus um die Betrachtung der minimalen Richtungswechsel oder Minimierung von Bedrohungen ist möglich. In [SYTX06] wird eine pseudo mehrkriterielle Optimierung eingesetzt. Es wird eine gewichtete Zielfunktion genutzt, um ebenfalls eine Minimierung der Routenlänge und Maximierung des Informationszugewinns sowie die Maximierung eines Nähemaßes betrachten zu können. Hingegen wird in [HAS12] unter Zuhilfenahme eines GA versucht die Redundanzen innerhalb der Routen und zwischen allen Routen zu minimieren. Die Optimierung wird für ein pseudo online TCM mit bekannter Systemumwelt durchgeführt. Für alle anderen TCMs wird jeweils nur ein Optimierungsziel für die Routenplanung untersucht. In der Regel wird nur die Abdeckungsredundanz minimiert.

Für ein online TCM ist die Systemumwelt zu Beginn der Abdeckung unbekannt. Aus diesem Grund können Routenplanungen nur unter Unsicherheiten vom TYP I vorgenommen werden. In der Lite-

[14] Engl. Multiple Ant Colony Optimization

ratur ist kein TCM vertreten, dass eine Routenoptimierung unter Unsicherheiten betrachtet. Ein erster Ansatz wird in dem TMM [SB05] vorgestellt. Aufgrund von Kommunikationsstörungen und Änderungen der Systemumwelt können Inkonsistenzen auftreten und getroffene Entscheidungen im Verlauf der Abdeckung ineffizient werden. Demzufolge wird vor jeder Entscheidungsfindung ein einheitlicher Informationsstand ermittelt. Ergeben sich Änderungen beim Befahren der geplanten Route, wird eine neue Auktion initiiert. Ein erweiterter Ansatz wird in [SAB+00] beschrieben. Für dieses TMM unterliegen die ermittelten Sensordaten, bezüglich der Odometrie und des Sichtsystems, einem Rauschen und sind resultierend mit Unsicherheiten vom TYP I und II behaftet. Die Unsicherheiten bezüglich des Zustandes einer Zelle [frei|blockiert|unbekannt] wird mittels einer Wahrscheinlichkeitsverteilung abgebildet. Für jede Zelle wird kontinuierlich die Wahrscheinlichkeit bezüglich des Zellzustandes bestimmt. Hierbei bestimmt sich der Zellzustand anhand von Schwellenwerten. Beispielsweise steigt die Wahrscheinlichkeit, dass eine Zelle frei ist, mit der Anzahl an Sensormessungen ohne Hindernisse in der Nachbarschaft. Für das TCM in [YJL08] wird für die Unsicherheiten vom TYP II eine verbesserte Methode angegeben. Aufgrund von Messfehler kann die befahrene Route von der geplanten abweichen. Eine vollständige Abdeckung wird durch die Anpassung der maximalen Ausdehnung der semi-approximativen Zerlegung der Systemumgebung an die maximale Abweichung von der geplanten Route gewährleistet.

Es können zwei Ansätze unterschieden werden, um auf Änderungen der Systemumwelt reagieren zu können. Zum einen kann ein Systemverhalten ausgeprägt werden, dass eine Gruppe von Agenten befähigt, die Gruppenstrategie anzupassen und die Aufgaben der Agenten zu reorganisieren. Hierbei sind global abgestimmte Informationen über die Systemumwelt für die Problemlösung auf Gruppenebene nützlich. Mit diesem Verhalten wird das Ziel verfolgt, die verfügbaren Ressourcen effizient einzusetzen. Zum anderen wird für die Ausprägung der Reaktivität das Systemverhalten durch individuelle Anpassungen der Agenten auf Änderungen der Systemumwelt bestimmt. Die Agenten können die eigenen Aufgaben reorganisieren, um das ursprüngliche Ziel zu erreichen. Der wesentliche Unterschied zwischen beiden Ansätzen liegt im Verteilungsgrad der Problemlösung mit dem das MAS auf geänderte Situationen reagiert [INS00].

Ein direkter Vergleich bestehender Ansätze erweist sich als schwierig. Zum einen liegt das an der Mannigfaltigkeit verschiedenartiger Ausprägungen hinsichtlich der Eigenschaften der einzusetzenden Agenten und Sensoren sowie der Ausprägung der Systemumwelt. Zum anderen existieren keine einheitlichen Testinstanzen für das TCP. Es wird empfohlen, ein TCM anhand von drei wesentlichen Strukturen der Systemumgebung zu überprüfen [SAB+00, ZJKK05]. Die freie Systemumgebung ist die einfachste Form. Die outdoor-like-Systemumgebung enthält oftmals konvexe Hindernisstrukturen und stellt somit eine Herausforderung an die Koordinierung und Routenplanung. Eine indoor-like-Systemumgebung, die einer Gebäudestruktur mit konkaven Hindernisstrukturen entspricht und zudem nur durch enge Zugänge erschlossen werden kann, ist besonders schwierig abzudecken. Es besteht die Gefahr, dass die Agenten die Zu- und Ausgänge nur schwer mit Abdeckungsredundanzen auffinden können. Des Weiteren existieren neben der totalen Explorationszeit und der Abdeckungsredundanz keine etablierten Metriken zum Messen der Effizienz, Robustheit, Flexibilität und Komplexität.

Insbesondere bei der frühzeitigen und zielorientierten Auswahl von Gestaltungsdimensionen möglicher Modellierungsansätze kann die Tabelle 2.4 im Sinne einer Entscheidungsunterstützung fungieren. Darüber hinaus kann die Entscheidungsunterstützung zusammen mit der in dieser Dissertation

vorgestellten Taxonomie ebenfalls genutzt werden, um bestehende TCMs/TMMs hinsichtlich möglicher Anwendungsbereiche zu klassifizieren. Für die in Tabelle 2.5 aufgeführten TCMs/TMMs können Modelle identifiziert werden, die sich beispielsweise für Anwendungen mit einer hohen Bedrohungslage eignen. Das TCM nach [PS90] und das TMM nach [SB05] weisen eine eingeschränkte Kommunikation auf und verfügen ausschließlich über Sensoren für den Nahbereich. Eine Aufklärung der Agenten kann dadurch erschwert werden. Des Weiteren werden homogene Agenten eingesetzt, die eine dezentrale Koordinationsstrategie verfolgen. Hingegen sind für das TCM nach [Pre14] und das TMM nach [SAB+00] Gestaltungsdimensionen umgesetzt, die eine Anwendung in stark unstrukturierten Systemumgebungen unterstützen. Vor allem die Betrachtung von Unsicherheiten bezüglich der Systemumgebung kann in der Koordination der Agenten berücksichtigt werden. Ferner wird die Konfliktlösung bei auftretenden Inkonsistenzen unterstützt. Der Kommunikationsfluss wird durch eine hohe Kommunikationsreichweite begünstigt. Die Ausrüstung der Agenten mit verschiedenen Navigationssensoren ermöglicht zusätzlich eine sichere Navigation. In Anwendungsbereichen, die einen agilen Einsatz von Agenten voraussetzen, implementiert das TMM nach [GHA+08] geeignete Modellierungsansätze. Es werden heterogene Agenten eingesetzt die sehr spezifisch, abgegrenzte Aufgaben übernehmen. Die Koordination der Agenten wird von einer zentralen Steuerungseinheit wahrgenommen. Resultierend wird der Rechenaufwand der Agenten reduziert. Ferner werden Informationen von den Agenten ungefiltert in einem hohen Kommunikationsradius an die zentrale Steuerungseinheit übermittelt.

In dem sich anschließenden Kapitel 3 wird der Agent im Sinne eines Robotersystems eingeführt. Es werden die Eigenschaften eines Agenten und darüber hinaus von einem MARS beschrieben. Als wesentliche Eigenschaft für ein online TCM wird die Robustheit eingeführt und verschiedene Ebenen dargestellt auf denen Unsicherheiten im Verlauf des TCP auftreten können. Abschließend werden mögliche Maßnahmen zur Steigerung der Robustheit für ein MARS erläutert.

3. Das Multi Agent Robotic System

Verbunden werden auch die Schwachen mächtig.
Johann Christoph Friedrich von Schiller (1759–1805)

Der Einsatz von einem MA(R)S kann grundsätzlich dazu beitragen, dass die Lösung des TCP im Sinne der Zielverfolgung und speziell der Effizienz, Robustheit sowie Flexibilität verbessert werden kann. Aufgrund der verteilten Agenten können mehr Informationen in eine kooperierende Entscheidungsfindung einfließen. Hierbei können beispielsweise Ressourcen geeignet verteilt oder Strategien angepasst werden. Die Zielerreichung wird durch die Organisation der Agenten über die Ausprägung einer Kommunikation und Kooperation sowie die Fähigkeit, dass Verhalten anderer Agenten wahrnehmen sowie basierend auf einem Regelwerk identifizieren, klassifizieren und antizipieren zu können, unterstützt.

Es ist unzureichend, ein MARS ausschließlich als Spezialfall eines MAS zu sehen, weil die Herausforderungen resultierend aus einer realen Systemumwelt signifikant sind und zu Anpassungen der Strukturen innerhalb eines MARS führen. Vor allem Unsicherheiten und unvollständige Informationen sollten in einem MARS im Rahmen der Entscheidungsfindung berücksichtigt werden. Des Weiteren führt die Generalisierung von einem Roboter hin zu einem MARS zu Erweiterungen hinsichtlich der internen Organisation und Annahmen gegenüber der Systemumwelt [FIN04].

Grundsätzlich muss im Rahmen der konzeptionellen und simulationsgestützen Überprüfung von Ansätzen im Forschungsfeld der MARSs das Real- und Modellsystem unterschieden werden. Das Modellsystem bildet hierbei ein in der Realität beobachtbares, konkretes System ab. Oftmals wird nur ein Ausschnitt des Realsystems modelliert. Die Abstraktion ist abhängig von dem Modellierungszweck und kann durch eine Aggregation von Einflussgrößen erreicht werden. Entscheidend ist die Validität, welche die Gültigkeit und Korrektheit des Modellsystems ausdrückt. Das zu modellierende Gesamtverhalten muss dem Realsystem entsprechen. Im Allgemeinen kann ein System als eine komplexe und begrenzte Entität beziehungsweise ein Prozess von agierenden Komponenten verstanden werden. Die Komponenten können hierbei Strukturen, Beziehungen, Interaktionen und Funktionen untereinander aufweisen [LK98].

Ein offenes System ist hierbei stets von einer Systemumwelt umgeben, welche die Menge aller Objekte umfasst, die einen Einfluss auf das System nehmen oder aufgrund des Systemverhaltens selbst einer Änderung unterliegen. Hingegen grenzt eine Systemgrenze das geschlossene System von der Systemumwelt ab. Resultierend bestehen keine Interaktionen mit Objekten außerhalb der Systemgrenze [HF56]. Jedes offene System kann in ein geschlossenes System überführt werden, indem die Systemgrenze um die in Interaktion stehenden Objekte der Systemumwelt erweitert wird.

Eine elementare Eigenschaft von Systemen ist die strukturelle Teilbarkeit, bei gleichzeitiger funktioneller Unteilbarkeit. Die Ursache dafür liegt in dem Verlust der Systemintegrität begründet. Dementsprechend treten Charakteristika auf der Systemebene nicht notwendigerweise in isolierten Komponenten auf [LK98]. Weiterhin muss ein (abstrahiertes) Modellsystem, gemäß dem Systemzweck, die Gültigkeit bezüglich der Struktur, des Verhaltens, der Empirie und der Anwendung gegenüber dem Realsystem aufweisen. Im Bezug auf die Strukturgültigkeit entspricht die Wirkungsstruktur des Modellsystems mit den elementaren Zustandsparametern und deren Verknüpfungen dem des Realsystems. Die Verhaltensgültigkeit hingegen zeigt, dass die Eingabemenge und Umwelteinwirkungen ein qualitativ gleiches Verhalten auslösen. Des Weiteren verhalten sich die empirischen Systemergebnisse unter gleichen Bedingungen homogen und konsistent. Die Anwendungsgültigkeit fordert eine stringente Modellierung hinsichtlich des Systemzwecks und der Anforderungen des Anwenders [Bos07]. Zusammenfassend können grundlegend nachfolgende Eigenschaften für ein System abgeleitet werden [LK98, Bos07, Rya07]:

- Ein System besteht aus mehreren Komponenten.
- Ein System ist durch Beziehungen zwischen den Komponenten geprägt.
- Ein System verfügt über eine Struktur.
- Ein System basiert auf einem Regelwerk.
- Ein System weist eine Systemgrenze auf.
- Ein System transformiert Eingaben in Ergebnisse.
- Ein System verfügt über eine Systemfunktion.
- Ein System ist durch emergente[1], funktionelle Systemeigenschaften geprägt.
- Ein System weist ein nicht triviales Systemverhalten auf.
- Ein System ist durch wechselwirkende Beeinflussungen mit der Systemumwelt geprägt.
- Ein System, beziehungsweise die Systemkomponenten sind nicht teilbar.

Der axiomatische Systemansatz nach Suh in [Suh01, Suh05] beschreibt mittels funktionaler Anforderungen (FR[2]) die minimale Menge an unabhängigen Erfordernissen, um die funktionellen Notwendigkeiten des Systems bereitzustellen. Demzufolge spezifiziert eine FR ein mit dem System verfolgtes Ziel. Ferner kann eine FR eine Funktion darstellen, die von dem System bereitgestellt wird. Des Weiteren können Nebenbindungen beschrieben werden, welche die Systemeingaben oder das Systemverhalten beschränken. Für die Ausführung des Systems werden Designparameter eingeführt, welche die physikalischen und funktionalen Aspekte des gewünschten Designumfangs unter Einhaltung der FR und der Nebenbedingungen charakterisieren. Zudem müssen die Designparameter die Unabhängigkeit der FR abbilden.

Nachfolgend werden in diesem Kapitel die Eigenschaften eines Agenten sowie darauf aufbauend eines MA(R)S eingeführt, mit dem Ziel einen geeigneten effizienten und robusten Ansatz zu entwickeln. Ferner wird das MA(R)S insbesondere im Kontext der Robustheit erörtert. Die Notwendigkeit einer Untersuchung der Robustheit eines MA(R)S begründet sich zum einen in der Zusammensetzung von individuellen Agenten zu einem koordinierten Gesamtsystem. Zum anderen agiert das MA(R)S in einer unbekannten Systemumwelt. Daraus resultieren Unsicherheiten, die geeignet bei einer effizienten Zielverfolgung beziehungsweise der Lösung des TCP betrachtet werden

[1] Die Emergenz beschreibt ein Phänomen, dass auf der globalen Systemebene beobachtbar ist. Hierbei können neue, übergeordnete Strukturen, Eigenschaften und Verhaltensweisen auf der Grundlage lokaler Interaktionen auftreten, welche nicht isoliert aus den individuelle Eigenschaften der Entitäten abgeleitet werden können[DGK05].

[2] Engl. Functional Requirement

müssen. Hierbei können fortlaufend Anpassungen an den Systemstrukturen durchgeführt werden, um die übergeordneten Ziele erreichen zu können. Es werden vier verschiedene Typen von Unsicherheiten beschrieben, die auf ein MA(R)S einwirken können. Abschließend werden Maßnahmen zur Steigerung der Robustheit vorgestellt.

3.1. Der Agent als Bestandteil eines Systems

Eine allgemeingültige Definition für einen Agenten wird in [WJ95] dargestellt. Demnach ist ein Agent ein hardware- oder softwarebasiertes Computersystem, das in einer Anwendungsumgebung autonom agiert und Kontrolle über die eigenen Aktionen und Systemzustände verfügt [Cas95]. Die Agenten können gemäß der Problemstellung und dem Anwendungsgebiet unterschiedliche Ausprägungen annehmen [Wei13]. Im Bereich der Webintelligenz werden Agenten als ein mobiler Programmiercode umgesetzt, um beispielsweise Informationen über Webinhalte zu sammeln [BVS06, LZZ02]. Für die Softwareentwicklung werden Agenten als Hintergrundprogramme beziehungsweise Module eingesetzt, um eine Softwareumgebung überwachen und bei Bedarf vordefinierte Funktionen ausführen zu können. Im Forschungsfeld der künstlichen Intelligenz können Agenten zu einem autonomen System zusammengesetzt werden, um flexibel Aufgaben zu erfüllen. Die Agenten lernen im Verlauf der Ausführung angepasst und effizient auf Änderungen zu reagieren [Fer99]. Hierbei werden deduktive und induktive Agenten unterschieden. Ein deduktiver Agent kann auf der Grundlage von Logikregeln Rückschlüsse hinsichtlich der Systemumwelt und dem eigenen Verhalten ableiten. Hingegen nutzt ein induktiver Agent beispielsweise Techniken des Maschinellen Lernens, um Schlussfolgerungen aus einer Menge von repräsentativen Trainingsdaten zu erlernen [Vid10].

Anhand der meisten Definitionen kann übereinstimmend ein Agent als ein physisches oder virtuelles System verstanden werden. Die nachfolgenden vier simultan auftretenden Eigenschaften charakterisieren hierbei einen Agenten [Syc98, WJ95]:

- **Autonomie:**
  Der Agent erreicht die vorgegebenen Ziele mit Hilfe eines integrierten Prozesses, welcher die Wahrnehmung, Analyse, Kommunikation, Planung, Entscheidungsunterstützung und das Handeln umfasst. Dieser Prozess wird ohne eine direkte Einflussnahme und Unterstützung durch einen Menschen durchlaufen. Aus diesem Grund verfügt der Agent über die Kontrolle bezüglich der eigenen Aktionen und Systemzustände [She92, PHM03]. Hierbei stellen die internen Systemzustände des Agenten die Einstellung gegenüber anderen Agenten und der Systemumwelt dar [NL04].

- **Situationsgebundenheit:**
  Das Verhalten eines Agenten richtet sich nach der Ausprägung der Sensordaten. Demnach kann ein Agent die Systemumwelt wahrnehmen, und das Handeln richtet sich situationsbedingt in Abhängigkeit der Systemumwelt aus. Hierfür muss die Fähigkeit der Analyse und Entscheidungsunterstützung ausgeprägt sein.

- **Soziabilität:**
  Der Agent verfügt über eine soziale Fähigkeit, in dem Sinne, dass mit Hilfe einer vordefinierten Sprache eine unmittelbare Interaktion mit anderen Agenten ermöglicht wird [GK94]. Hierbei können beispielsweise Informationen über die Systemumwelt oder den eigenen Zustand ausgetauscht werden, mit dem Ziel effizient zusammenarbeiten und gegebenenfalls Probleme lösen zu können. Des Weiteren kann eine Interaktion indirekt über Veränderungen der Systemumwelt erfolgen [NL04].

- **Adaptivität:**
  Die Adaptivität und Flexibilität eines Agenten setzt die vorangegangen Eigenschaften voraus und ermöglicht die Ableitung von drei Schlussfolgerungen. Zum einen kann das Verhalten aufgrund der Kontrolle der Aktionen und Systemzustände zielorientiert beziehungsweise opportunistisch und resultierend proaktiv ausgerichtet werden. Dies bedingt zugleich die Fähigkeit reaktiv auf Änderungen der Systemumwelt reagieren zu können. Im Sinne der künstlichen Intelligenz kann es dem Agenten ferner möglich sein, aufgrund der eigenen Erfahrungen und der Interaktion mit der Systemumwelt ein verbessertes Verhalten zu erlernen.

Ferner werden in dieser Dissertation ergänzende Eigenschaften für einen Agenten betrachtet [WJ95]:

- **Rationalität:**
  Der Agent verhält sich rational, wenn keine Aktionen ausgeführt werden, die eine gemeinsame sowie übergeordnete Zielerreichung verhindern oder beeinträchtigen. Stattdessen wird das Verhalten auf die Performanceverbesserung ausgerichtet [Gal88].

- **Persistenz:**
  Der Agent führt eine Abfolge von Aktionen aus, bis das individuelle oder übergeordnete Ziel erreicht ist oder der Agent erkannt hat, dass das Ziel nicht erreicht werden kann [CL90].

- **Mobilität:**
  Der Agent verfügt über ein Bewegungssystem, um sich in der Systemumwelt fortbewegen und die gestellten Aufgaben erfüllen zu können.

Gemäß der Autonomie ist dem Agent der eigene Zustand bekannt, und er verfügt über die Fähigkeit, Zustandsänderungen der Systemumwelt wahrzunehmen. Aufgrund von individuellen Interaktionen mit der Systemumwelt können Agenten begrenzt Änderungen dieser durchführen [NL04]. Für die Fähigkeiten der Situationsgebundenheit und Adaptivität muss eine Bewertung von Handlungsalternativen durchgeführt werden, um geeignete und optimierte Aktionen, hinsichtlich der aktuell verfügbaren Informationen sowie der Ziele einer zugrundeliegenden Anwendung, auswählen zu können. Die Einführung einer Nutzenfunktion ermöglicht die Abbildung von Systemzuständen der Umwelt auf einen numerischen Wert, wie beispielsweise die Menge der reellen Zahlen $\mathbb{R}$. Demnach sei eine Menge von Systemzuständen $\mathfrak{Z}$ gegeben. Hierbei werden die wahrgenommenen Sensorinformationen der Agenten klassifiziert und auf die Menge der möglichen Systemzustände abgebildet. Die Definition einer Nutzenfunktion setzt ein vertieftes Verständnis bezüglich der Eingabeparameter, Wirkungsbeziehungen und der Problemstellung voraus. Die Nutzenfunktion θ_i des i-ten Agenten im MAS kann gemäß nachfolgender Form dargestellt werden [Vid10]:

$$\theta_i : \mathfrak{Z} \rightarrow \mathbb{R} \tag{3.1}$$

Die Nutzenfunktion stellt zudem auch die Grundlage für die Soziabilität und Koordination der Agenten dar. Mit Hilfe der Qualitätsbewertung möglicher Aktionen eines Agenten kann die Aufgabenbearbeitung optimiert werden. Hierbei kann der Nutzen über alle Agenten und Aufgaben maximiert werden und dadurch eine effizientere Aufgabenverteilung erfolgen. In diesem Zusammenhang kann das rationale Verhalten im Sinne des Selbstinteresses modelliert werden, welches zur Koordination eingeführt werden kann. Beispielsweise liegt der maximale Nutzen für jene Aktion vor, für die eine Unterstützung eines anderen Agenten beschrieben wird. Obwohl der Agent ausschließlich den eigenen Nutzen maximiert und somit im Selbstinteresse handelt, liegt eine Kooperation zwischen den Agenten vor. Demnach kann mit Hilfe der Nutzenfunktion das Verhalten der Agenten beschrieben werden [Vid10].

Zusammenfassend richtet ein Agent das Verhalten auf die Erfüllung von Zielen aus, stets unter Beachtung der eigenen Ressourcen, der Fähigkeiten, der Systemumwelt und der erhaltenden Informationen. Entsprechend kann ein Agent mittels der nachfolgenden Eigenschaften beschrieben werden [Wei13]:

- Ein Agent ist bestrebt, Ziele zu erreichen.
- Ein Agent verfügt über eigene Ressourcen.
- Ein Agent verfügt über individuelle Fähigkeiten.
- Ein Agent kann mit anderen Agenten kommunizieren.
- Ein Agent verfügt nur über ein begrenztes Abbild der Systemumwelt.
- Ein Agent kann in der Systemumwelt agieren.

Die Zusammensetzung der individuellen Agenten zu einem komplexen System erfordert verschiedene Methoden der Interaktion, um ein zielgerichtetes Verhalten des MA(R)S auszuprägen. Im nächsten Abschnitt werden Formen der Interaktion eingeführt. Hierbei werden die Kollektivität, Koordination, Kooperation und Kollaboration unterschieden.

3.2. Das Multi Agent System

Die Forschung Anfang der 1980er Jahre hinsichtlich Systemen, welche aus mehreren Agenten bestehen, resultiert primär aus dem Forschungsfeld der verteilten künstlichen Intelligenz. Neben den MAS hat sich die verteilte Problemlösung als weiteres Forschungsfeld etabliert [BG88b]. Für das MAS werden verschiedene Methoden untersucht, um mehrere Agenten zu einem komplexen System zusammenzusetzen, welches ein emergentes Verhalten erkennen lässt und auf ein Ziel ausgerichtet ist. Oftmals werden etablierte Methoden und Algorithmen aus den Forschungsfeldern des Operations Research adaptiert, um die Koordination der Agenten zu verbessern [Vid10].

Ein MAS besteht aus eine Menge von interagierenden und (autonomen) Agenten, um ein Problem lösen zu können [Syc98]. Ferner können die Agenten als ein lose gekoppeltes Netzwerk von Problemlösern betrachtet werden. Das gestellte Problem kann hierbei außerhalb der individuellen Fähigkeiten und dem Wissen der Agenten liegen und nur durch Kooperation aller Agenten gelöst werden. Des Weiteren wird das Problem durch die Agenten asynchron bearbeitet [DL89]. Unter der Asynchronität wird hierbei verstanden, dass die Durchführung von Prozessen, zur Lösung des Problems, unabhängig von anderen Prozessen ist. Dementsprechend werden Prozesse durchgeführt und

führen zu einem Ergebnis, unabhängig davon, ob andere Prozesse fehlerbehaftet sind oder zeitlich verzögert stattfinden [HS99]. Zusammenfassend erfüllt ein MAS die nachfolgenden Eigenschaften [JSW98]:

- Ein MAS besteht aus Agenten, die über begrenzte Informationen oder Fähigkeiten verfügen, um das gestellte Problem zu lösen.
- Ein MAS verfügt über keine globale Steuerungs- und Kontrollinstanz.
- Für ein MAS liegen die Daten und Informationen dezentralisiert über die Agenten verteilt vor.
- Für ein MAS ist die Problemlösung asynchron.

In Bezug auf ein MAS verfügen die Agenten über verteilte Informationen und Ziele. Mit Hilfe der Soziabilität und der dafür benötigten Kommunikation sind die Agenten bestrebt, ein übergeordnetes Ziel zu erreichen, welches den einzelnen Agenten nicht zwangsläufig bekannt sein muss [SL08a]. Die komplexen organisatorischen Strukturen ermöglichen eine Kollaboration sowie Kommunikation und somit den Wissensaustausch. Hierbei ist die Kommunikation wesentlich, um Aktivitäten koordinieren und somit mehrschichtige Abläufe ausführen zu können. Es kann die hierarchische und horizontale Organisationsstruktur unterschieden werden. Für die hierarchische Organisationsform liegt eine Aufbauorganisation vor, welche durch eine vertikale Kommunikation gekennzeichnet ist. Hierbei wird festgelegt, welche Agenten verschiedener Hierarchieebenen in einer Punkt-zu-Punkt Kommunikation miteinander Informationen austauschen können. Hingegen können für die horizontale Organisationsstruktur drei verschiedene Formen der Kommunikation vorliegen. Zum einen kann eine Punkt-zu-Punkt Kommunikation realisiert werden, bei der alle Agenten gezielt, miteinander Informationen austauschen können. Zum anderen kann eine nicht selektierte Kommunikation implementiert werden, bei der eine Information an alle Agenten versendet wird. Die dritte Form der Kommunikation wird mit Hilfe eines zentralen Vermittlers organisiert. Der Vermittler sammelt alle Informationen der Agenten und kann angefragte Informationen zurücksenden [WSS11]. Die Agenten können aufgrund einer dezentralen Koordination und Kommunikationseinschränkungen über einen unterschiedlichen Wissenstand bezüglich der Systemfunktionalitäten, des Systemzustandes und der Systemumwelt verfügen [NL04, ROE09].

Darüber hinaus ermöglicht ein verteiltes MAS, mit der zunehmenden Komplexität von Problemen und deren Lösungsfindung umzugehen. Es werden hierbei die Techniken der Modularisierung und Verteilung genutzt. Die Lösungsplanung eines Problems ist allgemeingültig für einen Agenten der Prozess, welcher die Ermittlung von aufeinanderfolgenden Aktionen bestimmt. Die Sequenz der Aktionen definiert sich anhand der Ziele, Fähigkeiten und Systemumwelt. Für ein MAS müssen hinsichtlich der Planung von Aktionen zusätzliche Nebenbedingungen betrachtet werden. Zum einen beruhen die Aktionen der Agenten auf den Aktivitäten der anderen Agenten. Zum anderen basieren die Verpflichtungen unter den Agenten selbst auf den eigenen Aktionen. Des Weiteren sind die Veränderungen innerhalb des MAS und der Systemumwelt, aufgrund der Dynamik und der verteilten Informationen, mit Unsicherheiten behaftet oder gar nicht vorhersagbar [JSW98]. Aus diesen Gründen ist es notwendig, die Interaktion zwischen den verteilten Agenten zu organisieren, um die Aktivitäten der Agenten effizient auf ein übergeordnetes Ziel ausrichten und planen zu können [SL08a, Wei99]. Hierbei können allgemein Kooperation, Kollaboration, Koordination und Kollektivität unterschieden werden.

Die einfachste Form der Interaktion zwischen Agenten stellt die Kollektivität dar. Hierbei teilen die Agenten gemeinsame Ziele, und die Handlungen der Agenten wirken sich positiv auf die Zielerreichung der anderen Agenten aus. Die Agenten können aber keine Rückschlüsse auf die Intentionen oder zukünftigen Handlungen anderer Agenten ableiten. In der Regel führen die Agenten einfache Handlungen aus, welche sich zu emergenten Eigenschaften des Gesamtsystemes zusammensetzen [Par08].

Unter der Koordination wird die Fähigkeit verstanden, dass von Agenten ausgehend von den aktuellen Zuständen und den zu erreichenden individuellen Zielen jeweils abgestimmte Pläne erstellt werden. Die Handlungen der Agenten nehmen dementsprechend keinen direkten Einfluss auf die Zielerfüllung der anderen Agenten. Hierbei können für den übergeordneten, gemeinsamen Plan unterschiedliche Ziele verfolgt werden, die ferner nicht notwendigerweise unter den Agenten bekannt sein müssen. Aus diesem Grund basiert die Koordination weniger auf den individuellen Zielen der Agenten, als vielmehr auf den Verpflichtungen, die beobachtbare Beziehungen untereinander darstellen [Sin11]. Des Weiteren kann von den optimalen individuellen Teilplänen allgemeingültig nicht auf den optimalen gemeinsamen Plan geschlossen werden [EJ02, DM06, Par08].

Die Kooperation hingegen stellt die Fähigkeit dar, Konflikte zwischen Agenten lösen und folglich ein zielgerichtetes Zusammenwirken erzielen zu können. In [DL89] werden verschiedene Ziele der Kooperation definiert. Zum einen soll durch eine zeitgleiche Bearbeitung von Aufgaben und den Austausch von Ressourcen die Effizienz der Aufgabenbearbeitung gesteigert werden. Zum anderen soll die Wahrscheinlichkeit der Aufgabenerfüllung durch verschiedene Alternativen der Realisierung erhöht werden. Vor allem sollen aber die Störungen beziehungsweise negativen Einflussnahmen zwischen Aufgaben minimiert werden. Die Handlungen wirken sich hierbei direkt auf die Zielerfüllung der anderen Agenten aus [Par08].

Des Weiteren ermöglicht die Koordination sowie Kooperation der Agenten und die Ausprägung eines Gruppenverhaltens, dass die Agenten ihre Fähigkeiten und das individuelle Wissen bereitstellen und gemeinsam nutzen [ROE09]. Zusammenfassend impliziert die Koordination nicht zwangsläufig eine Kooperation zwischen den Agenten. Ferner kann die Kooperation als Spezialfall der Koordination angesehen werden [BLR01].

Die Kollaboration kann als Mischform der Koordination und Kooperation betrachtet werden. Die Agenten verfolgen individuelle Ziele, aber die Handlungen haben einen direkten Einfluss auf die Zielerreichung anderer Agenten. Dementsprechend können sich die Agenten gegenseitig mit den individuellen oder komplementären Fähigkeiten unterstützen [Par08].

Mit Hilfe der Koordination und Kooperation kann der Gesamtnutzen Υ des MAS maximiert werden. Hierbei bildet Υ den Nutzen aller Agenten $a_i \in A$ im MAS ab. Entsprechend wird mit dem Nutzen der Agenten auch die Bewegung in der Systemumgebung assoziiert. Der Beitrag eines Agenten a_i am Gesamtnutzen Υ kann mittels der zugehörigen Nutzenfunktion θ_i beschrieben werden. Resultierend kann der Gesamtnutzen Υ eines MAS wie folgt definiert werden:

$$\Upsilon : \mathbb{R}^{|A|} \rightarrow \mathbb{R} \text{ mit} \tag{3.2}$$

$$\Upsilon(\theta_1(\mathfrak{z}_1), \ldots, \theta_{|A|}(\mathfrak{z}_{|A|})) = \sum_{i=1}^{|A|} \theta_i(\mathfrak{z}_i) \tag{3.3}$$

In dieser Dissertation wird der Gesamtnutzen Υ indirekt über die Maximierung der individuellen Nutzenfunktionen der Agenten maximiert. Entsprechend wird der Nutzen der individuellen Agenten untersucht. Das heißt, es wird weder für die Zielerreichung noch für die Aufgabenplanung ausschließlich der zu erwartende Gesamtnutzen des MAS betrachtet. Dieser Ansatz eignet sich für ein dezentralisiertes MAS sowie ein MAS mit eingeschränkter Kommunikation. Hierbei verfügen die Agenten nicht immer über die Gesamtheit aller Informationen, bezüglich der Systemumwelt und den Aktionen anderer Agenten. Für den Ansatz der direkten Maximierung von Υ sollte für die zentrale Kontroll- und Planungsinstanz stets alle Informationen über die Systemumwelt und die Aktionen der Agenten vorliegen.

Des Weiteren wird für die Agenten eine stark ausgeprägte Autonomie angenommen. Es werden in der Literatur verschiedene Skalen angegeben. Eine intuitive Unterscheidung der möglichen Ausprägungsstufen der Autonomie ist von Sheridan in [She92] angegeben. Demnach liegt in dieser Dissertation die höchste Stufe der Autonomie für die Agenten vor. Die Agenten treffen sämtliche Entscheidungen selbst und setzen keine Interaktionen mit dem Menschen voraus. Eine detailliertere Unterscheidung der Autonomiestufen wird in [PHM03] angegeben. Es werden die Betrachtungsebenen: Informationssammlung, Situationsverständnis, Entscheidungsprozess und Handlungsausführung unterschieden. In dieser Dissertation sammeln und analysieren die Agenten die vorhandenen sowie benötigten Daten selbstständig. Ferner können Vorhersagen und Rückschlüsse bezüglich der Systemumgebung vorgenommen werden. Auf der fusionierten Datengrundlage wird die Bearbeitung der Teilaufgaben festgelegt. In allen vier Betrachtungsebenen besteht keine Interaktion mit dem Menschen. Es liegt demnach auch für diese Skala die höchste Ausprägungsstufe vor.

Nachfolgend wird das MARS eingeführt und entsprechend die Agenten, im Sinne eines hardware- und softwarebasierten Robotersystems betrachtet. Insbesondere die Interaktion zwischen den Agenten sowie die Fortbewegung der Agenten sind durch technische Komponenten eingeschränkt und unterliegen besonderen Herausforderungen.

3.2.1. Das Multi Agent Robotic System

Ende der 1980er Jahre leiten sich aus der Klasse der MAS die MARS ab. Hierbei wird von nicht real existierenden Agenten in einer abstrakten Systemumwelt auf real agierende Roboteragenten[3] in einer physischen Systemumwelt gefolgert [Tri08]. Reale Systemumwelten erschweren die Koordination und Bewegung, weil eine Vielzahl von Umwelteinflüssen auf die Agenten einwirken. Die Lokalisierung und Dokumentation der Systemumwelt muss mittels Sensoren korrekt erfasst werden. Die verschiedenen Formen der Interaktion von Agenten in einem MAS werden für das MARS übernommen. Hierbei werden oftmals Optimierungsmethoden aus den Bereichen der Wirtschaftswissenschaften, Biologie und Graphentheorie adaptiert. Die Integration solcher Methoden wird durch die zunehmende Rechenleistung mobiler Roboter ermöglicht [GM04a].

[3] Der Begriff Roboter leitet sich von dem tschechischen Wort robot ab und wurde erstmals 1921 vom Schriftsteller Čapek in dem Drama R.U.R. - Rossum's Universal Robots eingeführt [ČS28].

Das MAS kann zudem um Methoden und Theorien der Schwarmrobotik (SR[4]) erweitert werden, um die Koordination im Schwarm [Par98], einer Kolonie [AB97] oder einer Gruppe von Robotern [DJMW96] zu modellieren. Das Wissenschaftsgebiet der SR basiert auf Beobachtungen aus der Natur. Insbesondere werden Insektenschwärme untersucht, die in einer Gemeinschaftsform organisiert sind. Der Schwarm zeichnet sich durch eine hohe Anzahl von homogenen Individuen aus, welche nur über begrenzte Kommunikationsmittel verfügen, um eine Aufgabe effizient zu bearbeiten [Dc04]. Weiterhin ist es dem Schwarm möglich, bei Ausfall einzelner Individuen diese zu ersetzen und sich bei Umweltveränderungen zu reorganisieren. Die Fähigkeit der Reorganisation wird durch einen modularen Aufbau unterstützt. Die Aufgaben werden dezentral und parallel bearbeitet. Hierbei werden durch die Insekten unter anderem die Konzepte der Stigmergie oder eines charakteristischen Bewegungsmusters wie bei Bienen verfolgt [Ben04, Dye02, Tri08]. In der Regel sind Agenten innerhalb eines MAS im Sinne des Selbstinteresses modelliert. Die Aktivitäten werden in einer gegenseitig unterstützenden Weise koordiniert. Die Eignung und Koordination des Selbstinteresses wird im Bereich der SR noch untersucht. Die Untersuchungen werden jedoch durch Kommunikationsprobleme und unscharfe Informationen erschwert [Sar07].

Das Ziel der SR ist es, künstlich intelligente Schwärme und Kolonien, welche idealerweise auf einem dezentralen, autonomen, selbst-organisierenden und adaptiven Verhalten basieren, zu entwickeln. Hierbei agieren Roboter, welchen eine Menge von Regeln beziehungsweise Handlungsanweisungen zugeordnet sind, lokal und autonom. Die Regeln können in einem nicht deterministisch basierten System definiert sein. Aufgrund des sich ändernden Umweltzustandes oder dem internen Zustand des Roboters, kann sich die Menge der Regeln anpassen [LB92]. Der Roboter kombiniert das mechanische Verhalten einer Maschine mit dem informationellen Verhalten eines Automaten [BW89]. Die lokalen Interaktionen führen zu einem komplexen Gesamtverhalten des Schwarms. Ein kennzeichnendes Merkmal ist, dass obwohl hierarchische Anweisungs- und Kontrollstrukturen fehlen, kein systematischer Mehrfachausfall oder hohe Anfälligkeit bezüglich der Robustheit auftritt [Bog08]. Vielmehr besitzen diese Systeme von nicht intelligenten Robotern eine Schwarmintelligenz. Die Schwarmintelligenz zeichnet sich durch ein intelligentes Kollektivverhalten aus, welches offensichtlich erkennbar wird in der Fähigkeit, unvorhersehbare spezielle Musterfolgen in der externen Systemumwelt zu analysieren und zu generieren [BW89]. Vor allem liegen die Vorteile der SR in den Bereichen der Skalierbarkeit, Anpassungsfähigkeit und Robustheit. Diese Vorteile werden erreicht, wenn die Auswirkungen von einzelnen Parametern auf das Gruppenverhalten bekannt sind und folglich das individuelle Verhalten der Agenten flexibel gestaltet werden kann [LWS07].

Grundlegend können die Vorteile eines MAS und der SR auf ein MARS übertragen werden. In [DJMW96, CFK97, YJA13] werden die Vorteile einer Gruppe beziehungsweise eines Schwarms gegenüber einer Anwendung mit einem einzelnen Agenten beschrieben. Zum einen können Aufgaben aufgrund der Parallelisierung von Teilaufgaben schneller bearbeitet werden. Zum anderen können vor allem komplexere Aufgaben aufgrund der gesteigerten und erweiterten Fähigkeiten bearbeitet werden. Die Fähigkeiten der individuellen Agenten können sich hierbei ergänzen und führen zu einem gesteigertem Leistungsspektrum. Resultierend sind die Agenten weniger komplex und das MARS ist demnach weniger fehleranfällig sowie flexibler. Des Weiteren verhält sich das MARS robuster gegenüber Störungen und dem Ausfall von Agenten. Die Agenten können zudem aufgrund der räumlichen Dislozierung mehr Informationen über die Systemumwelt sammeln und austauschen. Ferner kann mit den zusätzlichen Informationen die Aufgabenverteilung effizienter

[4] Engl. Swarm Robotic

vorgenommen werden [DS03a]. Weiterhin wirkt sich der Informationsaustausch positiv auf die Robustheit aus, weil aufgrund der zusätzlichen Informationen eine gesteigerte Fehlertoleranz vorliegt. Aus diesem Grund kann beispielsweise die Positionsbestimmung der Agenten robuster ermittelt werden [FBKT00]. Die aufgeführten Vorteile basieren auf den positiven Eigenschaften, die an ein MAS gerichtet werden, welche im nachfolgenden Abschnitt erläutert werden [WSS11].

Grundsätzlich erfüllt ein MA(R)S, welches aus mehreren Agenten, im Sinne eines Robotersystems, zusammengesetzt ist und ein übergeordnetes Ziel verfolgt, die allgemeinen Eigenschaften eines Systems. Das MA(R)S besteht demzufolge aus mehreren nicht teilbaren Systemkomponenten (Agenten), die in Beziehungen zueinander stehen. Beispielsweise können sich die Agenten untereinander, aufgrund verschiedener Fähigkeiten, bei der Aufgabenerfüllung unterstützen. Die Interaktionsbeziehungen und Verhaltensweisen basieren dabei auf einem Regelwerk. Zusätzlich kann eine Struktur vorliegen, die sich in einer Organisationsstruktur manifestieren kann. Das MA(R)S grenzt sich zu der Systemumwelt ab und steht zu dieser in einer wechselwirkenden Beziehung. Die Agenten können Manipulationen an der Systemumwelt vornehmen oder adaptieren die eigenen Verhaltensregeln aufgrund von Änderungen der Systemumwelt. Des Weiteren transformiert das MA(R)S die Sensormessungen zu Ergebnissen, die sich unter anderem in der Steuerung der Agenten widerspiegeln und die, für das TCP, zur vollständigen Abdeckung der Systemumwelt führen. Hierbei ist das Systemverhalten durch eine Abfolge von nicht trivialen Systemfunktionen geprägt. Ferner sind für das MA(R)S emergente Systemeigenschaften zu beobachten, die auf Ebene der individuellen Agenten nicht vorhanden sind. Hierbei kann die Maximierung der Nutzenfunktion auf Ebene des MA(R)S ein emergentes Systemverhalten darstellen, wenn die individuellen Agenten stets darauf bedacht sind, den individuellen Nutzen zu maximieren. Erst die Koordination der Agenten ermöglicht die Ausprägung des emergenten Systemverhaltens.

Nachfolgend werden weitere elementare Eigenschaften eines MA(R)S beschrieben. Hierbei können grundlegend die Ausprägungen der Autonomie, Situationsgebundenheit, Adaptivität, Rationalität, Persistenz und Mobilität eines Agenten auf das MA(R)S übertragen werden. Aufgrund einer autonomen, effizienten und robusten Interaktion mehrerer Agenten gleichzeitig, nehmen für ein MA(R)S die Flexibilität, Stabilität, Skalierbarkeit, Kohärenz, Selbstorganisation und Dezentralisation einen hohen Stellenwert ein.

3.2.2. Eigenschaften eines Multiagent (Robotic) Systems

An ein autonomes MAS beziehungsweise MARS werden Eigenschaften beziehungsweise Voraussetzungen gestellt, die eine zielführende und optimierte Interaktion sowie Koordination und Kooperation der Agenten oder zugrundeliegenden Systemprozesse ermöglichen. Hierbei können die Eigenschaften der individuellen Agenten auf das MA(R)S abstrahiert werden. Die Situationsgebundenheit der Agenten überträgt sich auf ein MA(R)S in dem Sinne, dass Entscheidungen des MA(R)S, in Abhängigkeit der Gesamtheit aller verfügbaren Sensordaten, sowie der Systemzustände der Agenten und der Systemumwelt, getroffen werden. Ferner können aufgrund der dynamischen Umwelt getroffene Entscheidungen ineffizient oder sogar ineffektiv werden. Aus diesem Grund können die drei Verhaltensweisen zur Ausprägung der Adaptivität eines Agenten nach [Syc98] auf ein MA(R)S abstrahiert werden. Demnach kann es dem MA(R)S möglich sein, das Verhalten sowohl proaktiv und reaktiv auszurichten als auch verbessertes Verhalten zu erlernen. Ein rationales Ver-

halten ist für ein MA(R)S genau dann ausgeprägt, wenn die Aktionen der individuellen Agenten die gemeinsame, übergeordnete Zielerreichung nicht beeinträchtigen [Gal88]. Hierbei verhält sich das MA(R)S persistent, genau dann wenn die individuellen Aktionen der Agenten solange ausgeführt werden, bis die Gesamtheit aller übergeordneten Ziele erreicht ist oder eine Erfüllung als unmöglich erkannt wird [CL90].

Die unter Kapitel 5 vorgestellten MOO TCMs erheben den Anspruch, die nachfolgenden Eigenschaften zu erfüllen, welche die Überlegung von Dias und Stenz in [DS01, Dia04] erweitern:

- **Autonomie:**
  Abgeleitet aus der vollständigen Autonomie der individuellen Agenten, erreicht das MAS das übergeordnete Aufgabenziel ebenfalls autonom. Weiterhin sind für ein autonomes MAS die Kooperation sowie Koordination der Aufgabenplanung und resultierend die komplexen Interaktionen wesentlich, um das übergeordnete Aufgabenziel erreichen zu können [Sin11]. Hierbei kann das komplexe, emergente Systemverhalten auf einem einfachen Regelwerk von individuellen Verhaltensanweisungen basieren [DGK05]. Des Weiteren kann ein autonomes Verhalten zugleich eine bestimmte Ungewissheit und Unvorhersehbarkeit bezüglich der Reaktionen auf einwirkende Bedingungen und Nachrichten bedeuten [NL04]. Für beobachtende Agenten folgen zwei grundlegende Probleme. Zum einen können die internen Zustände der Agenten und die Informationen bezüglich der Systemumwelt nicht vollständig bekannt sein. Zum anderen haben Lernprozesse einen unvorhersehbaren Einfluss auf zukünftige Entscheidungen [NL04].

- **Effizienz:**
  Das Aufgabenziel legt fest, welche Dimensionen optimiert werden sollen, um die Zielerreichung möglichst geeignet auszurichten. Für das TCP kann der Grad der übergeordneten Zielerreichung, beispielsweise durch die Minimierung der totalen Abdeckungszeit oder die Minimierung der redundanten Abdeckung, angegeben werden. Insbesondere für homogene und heterogene Agenten, mit sich überlagernden Fähigkeiten, nimmt die Effizienzbetrachtung einen hohen Stellenwert ein. Hierbei kann eine Aufgabe durch mehrere Agenten bearbeitet werden. Demnach sollte stets ein Agent bestimmt werden, der in der aktuellen Situation am geeignetsten hinsichtlich der Zielerreichung ist [Par94].

  Des Weiteren werden für ein MAS oftmals mehrere konträre Aufgabenziele parallel betrachtet. Der übergeordnete Zielerreichungsgrad kann in Abhängigkeit zum Ressourceneinsatz stehen. Als ein Aufgabenziel kann sich beispielsweise die Minimierung des Ressourceneinsatzes negativ auf den übergeordneten Zielerreichungsgrad auswirken. Liegt eine derartige konträre Zieldefinition vor, sollte eine individuelle Effizienzbetrachtung durchgeführt werden.

- **Robustheit:**
  Die Robustheit kann verschiedene Ausprägungen annehmen. Eine Definition muss daher anwendungsbezogen erfolgen. Allgemein wird durch die Robustheit die Fähigkeit der Persistenz eines Systems dargestellt, sich strukturerhaltend bei einwirkenden Störungen zu verhalten. Hierbei bleibt in der Regel die emergente Systemfunktion ebenfalls erhalten [Jen02, DGK05]. Bezogen auf ein MAS können beispielsweise Kommunikationsstörungen, Umweltveränderungen, Planungskonflikte und der Ausfall von Agenten und Sensoren auftreten. Entscheidender Faktor ist hierbei die Wahrnehmung der Veränderungen und Störungen, um geeignet reagieren zu können [Lim92, TCM01]. Ferner muss es dem MAS trotz der Störungen möglich sein,

das Aufgabenziel weiterhin möglichst effizient, in Anbetracht der verfügbaren Ressourcen, erreichen zu können. Aus diesem Grund müssen die individuellen Verhaltensweisen, Eigenschaften und Funktionen der Agenten über möglichst viele Agenten verteilt sein, um für kritische Ereignisse genügend alternative Ressourcen aufweisen zu können. In Folge dessen wird jedoch die optimierte Aufgabenzuweisung unter den Agenten komplexer.

Weiterhin sollte jeder Agent in der Lage sein, Entscheidungen über das eigene Verhalten treffen zu können. Abhängigkeiten bezüglich zentralisierter Agenten, die eine Koordination anderer Agenten vornehmen, sind zu minimieren, um das MAS robust gegenüber dem Ausfall einzelner kritischer Agenten modellieren zu können [Par94]. Eine detaillierte Darstellung der Robustheit und die Einordnung der aufgeführten Unsicherheiten wird im nachfolgenden Kapitel vorgenommen.

- **Allgemeingültigkeit:**
  Im Sinne der Generalisierung soll es einem MAS möglich sein, in verschiedenen Situationen agieren zu können. Hierbei sind eine generalisierte Systemstruktur und allgemeingültige Handlungsanweisungen von Vorteil, um mit den unterschiedlichen situativen Anforderungen umgehen zu können.

- **Flexibilität:**
  Die Flexibilität steht in Beziehung zur Allgemeingültigkeit und es werden die Überlegungen des situationsgebundenen und adaptiven Verhaltens der Agenten aufgenommen. Um ein effizientes und adaptiertes Verhalten ausprägen zu können, müssen Anpassungen an der allgemeingültigen Systemstruktur vorgenommen werden können. Insbesondere in einem dynamischen Umfeld sollte ein MAS die Fähigkeit zum Lernen besitzen, um Änderungen opportunistisch ausnutzen zu können. Ein lernendes Verhalten kann dazu beitragen, dass Parameter- und Systemanpassungen zur Laufzeit autonom und zeitgerecht realisiert werden. Häufig geht mit einer zunehmender Flexibilität eine gesteigerte Systemkomplexität einher [NL04].

- **Stabilität:**
  Die Stabilität steht in einem Zusammenhang mit der Robustheit. Ein MAS versucht nach einer Störung wieder in ein Gleichgewicht überzugehen. Zum einen kann das Gleichgewicht für ein MAS mit Hilfe der Performance definiert werden. Demzufolge liegt ein Gleichgewicht vor, wenn die statistischen Eigenschaften der Leistungsindikatoren unverändert bleiben. Wirkt sich hingegen eine Störung substanziell und anhaltend negativ auf die Performance aus, ist das MAS als instabil zu betrachten [LNNW98]. Wird hingegen das Gleichgewicht anhand des Systemverhaltens beschrieben, liegt ein instabiles MAS genau dann vor, wenn aufgrund einer nichtlinearen Dynamik geringe Änderungen der Systemumwelt fortlaufend zu signifikanten Änderungen des emergenten Systemverhaltens führen. Für den Fall, dass mehrere stabile Zustände existieren, können mehrere Gleichgewichte beobachtet werden [DGK05].

- **Skalierbarkeit:**
  Die Skalierbarkeit gibt an, inwiefern die Leistungsfähigkeit eines MAS mit steigender Anzahl von Agenten zunimmt. Aus diesem Grund müssen geeignete Koordinationsmechanismen im Sinne der Aufgabenverteilung vorhanden sein, die eine parallele Bearbeitung von Aufgaben ermöglichen. Der Umfang eines MARS wird häufig durch Anzahl der zur Verfügung stehenden Ressourcen beschränkt.

- **Kohärenz:**
  Mit der Kohärenz wird die Ausprägung eines MAS hinsichtlich der Geschlossenheit und einheitlichen Zielverfolgung angegeben. Die Summe aller individuellen Aktionen muss auf ein gemeinsamen Ziel ausgerichtet sein. Die Ausprägung der Kohärenz kann mit Hilfe der Güte oder Effizienz der ermittelten Lösung gemessen werden [BG88a]. Eine schwache Kohärenz liegt vor, wenn die selbe Teilaufgabe mehrfach durch verschiedene Agenten bearbeitet wird oder die Agenten inkompatible Aktionen ausführen. Aus diesem Grund müssen die Agenten über die Fähigkeit verfügen, auftretende Konflikte lösen zu können [Par94]. Unter Umständen müssen sogar Anpassungen hinsichtlich der Organisationsstruktur und Rollenverteilung vorgenommen werden, um Konflikte möglichst effizient lösen zu können. Die Konflikte können aufgrund von Störungen, Umweltveränderungen und Unsicherheiten auftreten [BG88a].

- **Selbstorganisation:**
  Die Selbstorganisation stellt die Eigenschaft dar, ohne externe, konkrete Anweisungen angepasste Änderungen an der Aufbaustruktur und dem Systemverhalten während der Ausführungszeit durchzuführen [DGK05]. Aus diesem Grund werden vor allem die Eigenschaften der Autonomie, Allgemeingültigkeit und Flexibilität vorausgesetzt. Die Selbstorganisation ist hierbei die Grundlage für eine optimierte Zuweisung der Teilaufgaben mit der Intention der Zielerreichung und nimmt resultierend einen integralen Bestandteil für die Koordination eines MAS ein [Par94]. Bei einer schwach ausgeprägten Selbstorganisation kann die Koordination des Systems auf internen Kontroll- und Planungsstrukturen beruhen, welche bei einer stark ausgeprägten Selbstorganisation nicht vorhanden sind. In der Regel liegen für die internen Kontroll- und Planungsstrukturen die Informationen verteilt vor und werden lokal unter den Agenten ausgetauscht, um optimierte Koordinationsentscheidungen dezentral treffen zu können [DGK05]. Eine optimierte und kontinuierliche Aufgabenverteilung muss gemäß dem Aufgabenziel unterschiedliche Faktoren berücksichtigen. Hierzu zählen unter anderem die verfügbaren Ressourcen und Agenten sowie die Umweltausprägungen, Risiken und Restriktionen. Ferner sollte es den Agenten im Sinne der Robustheit und Flexibilität möglich sein, die Aufgabenverteilung bei Systemstörungen und Umweltveränderungen sowie Performanceänderungen selbstständig erkennen, neu planen sowie zuweisen zu können. Diese Eigenschaft ist für eine dynamische Systemumwelt unabdingbar [Par94]. Wenn möglich, sollten im Sinne der Effizienz verschiedene Aufgaben durch die Agenten simultan bearbeitet werden [BMF+00].

Um die Autonomie und Robustheit eines MAS positiv ausprägen zu können, kann die Eigenschaft der Dezentralisation ergänzt werden. Grundsätzlich werden zentrale und dezentrale MAS unterschieden. Für ein zentrales MAS geht die Koordination der Agenten von einem Kontroll- und Planungsagenten aus. Hierbei muss für die Planung der Aufgabenverteilung der aggregierte Zustandsraum über die Kombination aller Zustandsräume der Agenten betrachtet werden. Insbesondere für eine hohe Anzahl von Agenten stellt sich dieser Ansatz als ineffizient und unpraktikabel dar. Hingegen wird die Koordinationsaufgabe für ein dezentrales MAS auf die Gesamtheit aller Agenten verteilt. Ein hybrides MAS kombiniert die Vorteile beider Ansätze. Unter bestimmten Umständen ist die Aufgabenerfüllung ausschließlich für einen dezentralen oder hybriden Ansatz gegeben. Beispielsweise sind die Begrenzungen der Systemumwelt derart strukturiert, dass ein monolithischer Kontroll- und Planungsagent nicht die notwendigen Kommunikationsbeziehungen zwischen den Agenten aufbauen kann. Resultierend können Aufgaben, für die mehrere Agenten benötigt werden, gar nicht oder nur unzureichend bearbeitet werden [Tri08].

- **Dezentralisation:**
  Die Koordination der Agenten und resultierend der Prozess der Entscheidungsunterstützung wird über die Gesamtheit der Agenten verteilt. Hierbei basieren die Entscheidungen und das Verhalten der Agenten auf lokalen Informationen. Jeder Agent ist für die Ausübung der Aufgaben eigenverantwortlich. Aus diesem Grund reduziert sich die Komplexität der Problemlösung und folglich die benötigte Rechenleistung. Das Verhalten der Agenten basiert bei einem dezentralen MAS grundlegend auf den Konzepten der Selbstorganisation, um autonom situative Entscheidungen treffen zu können [Tri08].

  Für ein dezentrales MAS liegen die Vorteile im Bereich der Flexibilität und Robustheit. Zum einen müssen keine direkten und dauerhaften Kommunikationsbeziehungen aufrechterhalten bleiben. Zum anderen können die Agenten dynamisch und zeitgerecht auf Änderungen reagieren. Die Kommunikation und Koordination mit anderen Agenten kann hierbei indirekt über Verfahren der Stigmergie erfolgen. Aufgrund einer fehlenden zentralen Kontroll- und Planungsinstanz kann das MAS in einen Deadlock-Zustand übergehen, für den sich die Aktionen der Agenten einander aufheben. Für diesen Zustand können zusätzliche Verhaltensregeln, Zufallsvariablen oder externe Maßnahmen definiert werden [Tri08].

Zusammenfassend nimmt die Robustheit einen wesentlichen Stellenwert für die effektive und effiziente Zielerreichung eines MA(R)S ein. Der hohe Stellenwert begründet sich dahingehend, dass eine Abhängigkeit zu den zuvor angegebenen Eigenschaften vorhanden ist. Vor allem für eine dynamische Systemumwelt muss das Verhalten des MA(R)S robust gegenüber Änderungen sein. Für das online TCP müssen neue Informationen über die Systemumwelt von dem MA(R)S analysiert und bewertet werden. Hierbei können ebenfalls Störungen, Änderungen und Ineffizienzen resultieren.

Nachfolgend wird die Robustheit hinsichtlich verschiedener Ebenen des MA(R)S beschrieben. Störungen können bereits bei der elementaren Sensorik der Agenten auftreten und sich über die Kommunikation bis hin zur Koordination und folglich auf die Zielerreichung des MA(R)S auswirken.

3.3. Die Robustheit bezogen auf ein Multiagent (Robotic) System

Die Robustheit für ein MAS sicherzustellen erweist sich aufgrund der hohen Komplexität und Flexibilität als schwierig. Hierfür liegen zwei wesentliche Aspekte zu Grunde. Zum einen kann die interne Modellierung der Agenten eine komplexe Struktur annehmen, um ein autonomes und dynamisches Verhalten abbilden zu können. Zum anderen wirken sich die übergeordneten Interaktionen zwischen den Agenten, welche für die kooperative Aufgabenlösung benötigt werden, auf eine Erhöhung der Komplexität aus [NL04, Woo09]. Beide Aspekte können zudem zu einem unvorhersehbaren individuellen als auch kooperativen Verhalten führen. Des Weiteren können technische Fehler auftreten, die zu Einschränkungen der Fähigkeiten bis hin zum Ausfall der Agenten führen können. In [FC12] wird der Aspekt der Sensitivität angefügt. Dementsprechend verhält sich ein System genau dann robust, wenn es sich nicht sensitiv gegenüber den Effekten des Rauschens und der Variabilität beziehungsweise der Veränderlichkeit von Systemparametern und Eingabevariablen verhält, selbst wenn die Ursprünge der Variabilität nicht eliminiert werden.

Im Nachfolgenden werden fünf verschiedene Ebenen betrachtet, in denen Störungen auftreten können. Die Ebenen umfassen die elementare Sensorik bis hin zu der globalen Koordination der Agenten. Die Störungen können lokal, global, gleichzeitig, vorübergehend oder permanent auftreten. Ferner können sich die Störungen über die Zeit akkumulieren. Ein negativer Einfluss auf die Performance ist zu vermeiden. Bei der Modellierung und Implementierung von einem MAS sollte die Robustheit für alle Ebenen berücksichtigt werden.

- **Ebene 1 - Sensorik:**
  Es können vor allem für mobile Roboter unterschiedliche interne Störungen und Fehler bezüglich der eingesetzten Sensoren und Effektoren auftreten. Es kann zwischen hardware- und softwarebasierten Störungen unterschieden werden. Die sensorbasierten Störungen gehören in der Regel zu der Klasse der softwarebasierten Störungen und können zu einem Fehlverhalten führen. Hierbei sind die mobilen Roboter den Einflüssen der Systemumwelt ausgesetzt, die zu Beanspruchungen der technischen Komponenten und Steckverbindungen führen. Für einen robusten Einsatz von Sensoren und Effektoren müssen Fehlzustände frühestmöglich erkannt werden, um weitere negative Auswirkungen auf die Performance sowie der Ausbreitung auf andere Komponenten entgegenzuwirken. Des Weiteren sollte ein paralleler Zugriff auf alle Sensoren und Effektoren sichergestellt sein sowie die Möglichkeit bestehen, dass Komponenten abgeschaltet werden können. Fehler in der Sensorik können beispielsweise auf die Lokalisation, Steuerung, Wahrnehmung und Messung interner Zustände sowie auf den Informationsaustausch negative Auswirkungen nehmen [Fer94, KAS12].

  Der Einsatz von mehreren Sensoren ermöglicht eine zuverlässigere und umfangreichere Messung der Systemumwelt. Ferner können die zusätzlichen Informationen zur robusten Bestimmung von Aktionen genutzt werden. Hingegen ermöglichen mehrere Effektoren der Fortbewegung eine höhere Anzahl von Freiheitsgraden [Fer94]. In [LL91] wird eine redundante Hardware eingesetzt, um die Robustheit und Zuverlässigkeit zu erhöhen.

- **Ebene 2 - Kommunikation:**
  Ein MAS basiert auf einer Kommunikationsinfrastruktur, um den Transfer von Datenpaketen zwischen den Agenten zu ermöglichen. Die Datenpakete können Informationen über die Systemumwelt sowie den eigenen Systemzustand und den Systemzustand anderer Agenten enthalten. Für eine robuste Kommunikationsinfrastruktur muss es den Agenten möglich sein, dem Kommunikationsnetzwerk flexibel bei- oder auszutreten. Zudem sollte das Kommunikationsnetzwerk über möglichst viele Agenten aufgespannt sein, um den Informationsfluss sicherzustellen. Bei der Kommunikation können verschiedene Störungen auftreten. Zum einen kann der Datentransfer durch Hindernisse und andere Einflüsse der Systemumwelt gestört sein. Zum anderen können die Sensoren zum Senden und Empfang der Datenpakete defekt sein. Aus diesen Gründen können die Datenpakete unvollständig sein oder an den falschen Agenten verschickt werden. Um Störungen vorzubeugen und identifizieren zu können, sollte die Kommunikation auf vier wesentlichen Maximen beruhen. Die Maxime zur Übertragung einer Nachricht umfassen den Umfang und Zusammenhang sowie die Qualität wie auch die Art und Weise. Die Nachricht sollte demnach nur die notwendigen Informationen, welche zudem fundiert und relevant zur Erfüllung der Aufgabe sind, präzise und eindeutig enthalten [SL08a]. Zusätzlich sollte ein eindeutiger Bezeichner für das Datenpaket selbst und falls notwendig für den Sender/Empfänger enthalten sein.

- **Ebene 3 - Konversation:**
  Eine Konversation zwischen Agenten umfasst die Gesamtheit aller Nachrichten, die ausgetauscht werden, um eine Aufgabe erfüllen zu können. Die Konversation umfasst dabei mindestens eine Anfrage und eine Antwort zwischen zwei Agenten. Hierbei kann eine effiziente und robuste Konversation zwischen Agenten auf der Grundlage von Regeln und Vereinbarungen basieren. Normen einer Konversation definieren die formale Ausprägung von Nachrichten. Demnach können verschiedene Arten von Nachrichten definiert werden, die einen vordefinierten Informationsumfang enthalten. Zusätzlich wird die Art und Weise, wie Informationen zwischen Agenten ausgetauscht werden, bestimmt. Beispielsweise bedingen unterschiedliche Rollen einen differenzierten Verlauf einer Konversation. Ferner ermöglichen Normen einen standardisierten Informationsaustausch und das Dekodieren von Nachrichten. Dadurch können Fehler und Inkonsistenzen in der Konversation einfacherer identifiziert werden, um einen robusteren Informationsaustausch gewährleisten zu können. Demnach sollte es jedem Agenten möglich sein, die Quelle von Kommunikations- und Agentenfehler in einer Konversation zu erkennen. Dazu muss der Agent in der Lage sein, den Verlauf einer Konversation nachvollziehen zu können. Die Fähigkeit, den Verlauf einer Konversation nachvollziehen zu können, wird im gleichen Maße benötigt, damit ein Agent zu einem späteren Zeitpunkt einer Konversation beitreten kann. Zusätzlich muss diese Fähigkeit implementiert werden, wenn Abhängigkeiten zwischen den Entscheidungen der Agenten bestehen [NU00].

  Zusammenfassend können Fehler in einer Konversation zum einen aufgrund von einer gestörten Kommunikation und zum anderen aus nicht abgestimmten Normen resultieren. Des Weiteren können Fehler zu einem ineffizienten oder gestörten Verhalten der Agenten führen, welches bis zu einem Ausfall von Agenten führen kann. Ein integriertes Beobachtungssystem kann dazu genutzt werden, um frühzeitig Fehler identifizieren zu können [Gri01].

- **Ebene 4 - Agent:**
  In [TSS+06, TS10] werden zwei verschiedene Kategorien von Fehlern unterschieden. Zum einen gibt es Fehler, die durch andere Agenten wahrgenommen und beobachtet werden können. Beobachtbare Fehler können durch ein auffälliges beziehungsweise untypisches Verhalten gekennzeichnet sein. Hierbei können Abweichungen von Normen und Regeln charakteristisch sein. Diese auch als Verhaltensfehler bezeichnete Klasse kann sich durch einer der Normen und Regeln falschen oder verspäteten Kommunikation manifestieren. Ebenfalls ist eine falsche Anfrage oder Informationsweitergabe möglich. Ferner können fehlerhafte Reaktion beziehungsweise allgemein Aktionen resultieren. Beispielsweise werden vereinbarte Regel und Normen nicht eingehalten. Des Weiteren können beobachtbare Fehler sich durch ein fehlerbehafteten Informationsstand manifestieren. Hierbei können Informationen allgemein falsch, unvollständig oder inkonsistent sein [BE96].

  Zum anderen können Fehler auftreten, die durch andere Agenten nicht beobachtet werden können. Hierbei kann es sich um interne Fehler handeln, die unter anderem Auswirkungen auf die Steuerung oder Informationsverarbeitung verursachen können. Des Weiteren können sich beispielsweise versteckte Fehler in der Programmierung zu einem späteren Zeitpunkt manifestieren. Andere Agenten können aufgrund von Konflikten einen Rückschluss auf mögliche nicht beobachtbare Fehler ziehen. Damit Agenten jedoch geeignet auf fehlerbehaftete Agenten reagieren können, sollten diese nach Möglichkeit identifizierte Fehler stets kommunizieren.

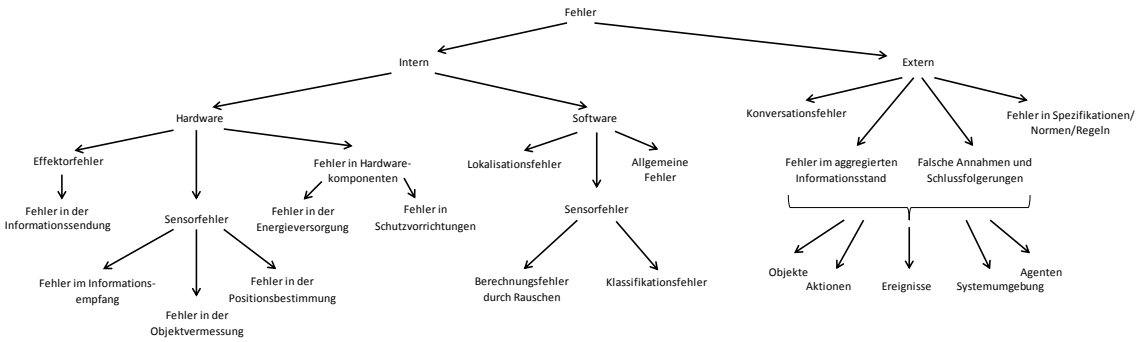


Abbildung 3.1.: Taxonomie von möglichen Fehlerquellen bezüglich dem Terrain Covering (angelehnt und erweitert nach [KAS12], Seite 103).

- **Ebene 5 - Koordination/Kooperation:**
  Die Koordination und Kooperation von Agenten umfasst die Planung und Abstimmung von individuellen Plänen aller Agenten, mit der gemeinsamen Absicht, ein übergeordnetes Ziel zu erreichen. Die Abstimmung schließt eine Aufgabenplanung der Agenten ein und ermittelt konfliktfreie Pläne. Das Vorliegen von gemeinsamen und geteilten Zielen, führt nicht notwendigerweise zu einer Koordination und Kooperation [CC95, Jen93].

  Fehler in der Koordination und Kooperation können aufgrund von kumulierten Fehlern aus den zuvor beschriebenen Ebenen resultieren. Hierbei können vor allem falsche oder fehlerbehaftete Informationen zu Ineffizienzen oder Konflikten in der Abstimmung von individuellen Plänen führen. Liegen einem Agenten hingegen nicht alle notwendigen Informationen vor, können Unsicherheiten in der Ausführung von Aktionen bestehen. Des Weiteren können nachfolgende Situationen auf Fehler in der Koordination und Kooperation hindeuten. Zum einen kann die Situation vorliegen, dass die Agenten keine adäquaten auf das Ziel ausgerichteten Aktionen identifizieren können, obwohl ein Zugang zu allen relevanten Informationen besteht. Zum anderen kann die Auswahl und Ausführung von Aktionen zu Ineffizienzen führen, wenn resultierende Nebeneffekte nicht vollumfänglich berücksichtigt wurden [DFJN97].

In Abbildung 3.1 ist zusammenfassend eine Taxonomie von möglichen Fehlerquellen im Rahmen des TC veranschaulicht. Es ist eine Unterscheidung nach internen und externen Fehlern aufgeführt. Hierbei resultieren interne Fehler durch Störungen oder Probleme, die von dem Agenten selbst ausgehen. Hingegen werden externe Fehler aus Konflikten mit anderen Agenten, den übergeordneten Regeln und Normen oder einem nicht konsistenten Informationsstand bezüglich der Systemumgebung hervorgerufen [KAS12].

Die Fehler und Störungen können sich über die Ebenen hinweg akkumulieren. Aus diesem Grund können fehlerbehaftete Messungen bei der Sensorik einen negativen Einfluss auf die Agenten, die Kommunikation, die Koordination und schließlich auf die Effizienz und Robustheit der Aufgabenerfüllung nehmen. Positionen von Agenten und Hindernissen können durch fehlerhafte Sensormessungen falsch ermittelt und kommuniziert werden. Zudem können, unter Nutzung der indirekten Kommunikation, Informationen falsch oder gar nicht wahrgenommen und verteilt werden. Resultierend wird eine ineffizientere oder nicht realisierbare Aufgabenplanung durchgeführt. Unter Umständen kann dies beispielsweise zu internen Fehlzuständen und inkonsistentem Verhalten sowie redundanten Abdeckungen oder Kollisionen führen.

Störungen können auf allen Ebenen einen negativen Einfluss auf den einheitlichen Informationsstand der Agenten nehmen. Verfügen die Agenten über unterschiedliche Informationen, können Konflikte auftreten. Für ein robustes Verhalten muss es den Agenten möglich sein, Konflikte bezüglich der Zielerreichung oder Planung von Aktionen erkennen und aufheben zu können. In Kapitel 2.3 werden die Teilprobleme der Konfliktlösung erläutert, um ein robustes Verhalten sicherzustellen. Bevor eine Konfliktlösung durchgeführt werden kann, muss eine Identifikation der Inkonsistenzen, Ineffizienzen oder Störungen erfolgt sein. Grundsätzlich sollten die Agenten versuchen eine Problemlösung aus den resultierenden Fehlern herbeizuführen, damit sich der Fehler nicht permanent auf die Zielerreichung auswirkt [KAS12].

Werden Fehler und Konflikte identifiziert, kann sich nach [KF95, TS10] eine Verhaltensreaktion in drei Qualitätsstufen zeigen. Eine besonders hohe Robustheit liegt vor, wenn alternative Aktionen ausgeführt werden, ohne dass die Performance oder Funktionalität signifikant abnimmt. Andernfalls werden alternative Aktionen durchgeführt, die eine begrenzte Abnahme der Performance und Funktionalität verursachen. Stehen keine alternativen, zielgerichteten Aktionen zur Verfügung, dann wird der Agent in einen Zustand überführt in dem die grundlegenden Funktionen, wie beispielsweise die Kommunikation, aufrecht erhalten werden können.

Grundsätzlich tragen die zuvor in Kapitel 3.2.2 beschriebenen Eigenschaften der Autonomie, Allgemeingültigkeit, Flexibilität, Stabilität, Kohärenz, Selbstorganisation, Dezentralisation dazu bei, dass für ein MA(R)S die Robustheit gesteigert werden kann. Mit Hilfe eines abgestimmten und standardisierten Informationsaustausches sowie der Fähigkeit, das Verhalten anderer Agenten wahrnehmen und analysieren zu können, kann ein robustes Verhalten ausgeprägt werden [TS10]. Des Weiteren können verschiedene Maßnahmen einer Steigerung der Robustheit umgesetzt werden, um einen positiven Einfluss auf die Effektivität und Performance der Agenten zu nehmen. Mögliche Maßnahmen werden in Kapitel 3.3.2 diskutiert.

Zusammenfassend impliziert die Robustheit die Herausforderung, das Systemergebnis unter Unsicherheiten zwischen einem konkreten Zielwert und einer geringen Varianz auszubalancieren [ZFM05, ASM06]. Im nachfolgenden Abschnitt werden im systemtheoretischen Kontext verschiedene Klassen von Unsicherheiten vorgestellt. Hierbei werden Unsicherheiten bezüglich der Systemumwelt, der Designvariablen, des Systemmodells und der Realisierbarkeit unterschieden.

3.3.1. Vier Typen der Unsicherheit

In abstrahierter Form lässt sich ein Systemergebnis $\mathbf{f} = \mathbf{f}(\mathbf{x}, \Xi)$ auf der Menge der Designvariablen $\mathbf{x}$ und den Eingangsgrößen der Systemumwelt Ξ definieren. Hierbei kann das Systemverhalten innerhalb der Designgrenzen mittels der Designparameter kontrolliert werden, um die Performance zu optimieren. Es können verschiedene Unsicherheiten auftreten, die entweder reduzierbar sind oder sich der Kontrolle vollständig entziehen [HS07]. Mittels einer Verbesserung der Genauigkeit der Messverfahren, Modellformulierungen oder Datensätze können die reduzierbaren Unsicherheiten minimiert werden. Nicht reduzierbare Unsicherheiten entziehen sich der Kontrolle und treten inhärent mit realen Systemen auf. Als Beispiel kann die natürliche Variabilität und bedingte Vorhersage der Systemumwelt genannt werden.

Zusammenfassend ist die Identifizierung, Klassifizierung und Quantifizierung von Unsicherheiten unabdingbar, um die Auswirkungen auf das Systemverhalten, die Entscheidungsprozesse und die resultierenden Systemergebnisse bewerten zu können [IRG98, Cho05].

Im systemtheoretischen Kontext können drei grundlegende Klassen von Unsicherheiten vom Typ I bis III unterschieden werden, die sowohl während des Designprozesses beachtet werden müssen als auch direkt auf ein System einwirken können [ASM06]. Des Weiteren wird eine zusätzliche Klasse angefügt, die sich wesentlich von den Typen I bis III unterscheidet [PSP93]. Die Unsicherheiten können mittels verschiedener mathematischer Formulierungen modelliert werden. Generell wird die deterministische, wahrscheinlichkeitstheoretische und Fuzzy Modellierung unterschieden. Für die deterministische Modellierung werden Intervalle definiert, in denen die mit Unsicherheiten behafteten Systemparameter variieren. Gemäß der Wahrscheinlichkeitstheorie werden für die Systemparameter bestimmte Ereignisse definiert. Das Eintreten eines Ereignisses wird durch die Zuordnung von Wahrscheinlichkeiten beschrieben. Hingegen beschreibt die Fuzzy Modellierung den Grad der Zugehörigkeit eines Systemparameters zu einer bestimmten Ausprägung. In Kapitel 4 werden für den Typ I die drei Modellierungen vorgestellt und überprüft.

Nachfolgend werden die Unsicherheiten des Typs I bis III eingeführt. Die Zusammenhänge und Ursprünge der Typen I bis III werden mit Abbildung 3.2 (a) veranschaulicht.

- **Typ I - Systemumwelt:**
  Für die Systemumwelt können Unsicherheiten $\mathbf{u}$ vorliegen, die direkt oder indirekt auf das System mit $\mathbf{f} = \mathbf{f}(\mathbf{x}, \Xi + \mathbf{u})$ einwirken. In der Regel kann das Rauschen dieser Umweltfaktoren nicht kontrolliert werden. Unter Umständen kann selbst schon die Existenz der Unsicherheiten nicht bekannt sein. Aus diesem Grund müssen geeignete Methoden etabliert werden, um das System gegenüber Unsicherheiten des Typs I robust zu gestalten. Mit Hilfe von Anpassungen und Kontrolle der Designparameter kann versucht werden, die Variabilität, verursacht durch das Rauschen, zu minimieren. Der Typ I basiert auf den Arbeiten von Taguchi [ASM06, HS07].

  In Bezug auf die Zielerreichung kann eine optimierte Aufgabenplanung und –verteilung nur unter Unsicherheiten vorgenommen werden. Beispielsweise liegen für bestimmte Regionen noch keine oder unvollständige Informationen vor, welche unter anderem einen Einfluss auf die absolute Entfernung, Energiebedarf oder prognostizierte Zeit zur Aufgabenerfüllung nimmt. Ferner können Ineffizienzen bezüglich der Aufgabenreihenfolge auftreten, welche daher fortlaufend kontrolliert werden sollten. Insbesondere unterliegt die Routenplanung der Agenten einer Vielzahl von Unsicherheiten. Zusätzlich kann sich im Verlauf der Zielerreichung die der Optimierung zugrundeliegende Menge von Heuristikparametern ändern.

- **Typ II - Designvariablen:**
  Die Designparameter können nur bis zu einem bestimmten Grad an Genauigkeit realisiert werden. Zudem können selbst hochtechnologiebasierte Komponenten verschiedenen Unsicherheiten unterliegen. Aus diesem Grund sollten Toleranzen bezüglich der Designvariablen eingeführt werden. Dementsprechend wirken die Unsicherheiten $\mathbf{u}$ zum einen unabhängig auf die Designvariablen mit $\mathbf{f} = \mathbf{f}(\mathbf{x} + \mathbf{u}, \Xi)$ ein. Zum anderen können die Unsicherheiten relativ bezüglich der Designvariablen auftreten [HS07]. Der Typ II geht auf die Ausführungen von Chen et al in [CATM96] zurück.

Beispielsweise können Unsicherheiten aus den einzusetzenden Sensoren resultieren. Die Sensoren liefern in den meisten Fällen nicht exakte Messwerte, die mit einem Rauschen belegt sind. Zudem ist die Verteilung des Rauschens häufig nicht eindeutig bekannt. Aus diesem Grund stellen die Messungen unscharfe Näherungen des Gemessenen dar und im Extremfall etwas ganz anderes [Bro92]. Ein weiteres Problem besteht darin, dass die Sensoren oftmals lediglich Grundeigenschaften wie Form, Dichte oder Farbe messen können. Nur mit Hilfe der Auswertung und Agglomeration der Messwerte können Rückschlüsse auf den eigentlichen Untersuchungsgegenstand getroffen werden. Weiterhin sind die Navigationssensoren bezüglich der Reichweite limitiert. Im gleichen Maße sind die Bewegungssensoren mit Unsicherheiten behaftet. Hierbei können sich die Messfehler mit der Zeit akkumulieren und zu erheblichen Positionsabweichungen und fehlerbehafteten Positionsbestimmungen führen [TB96].

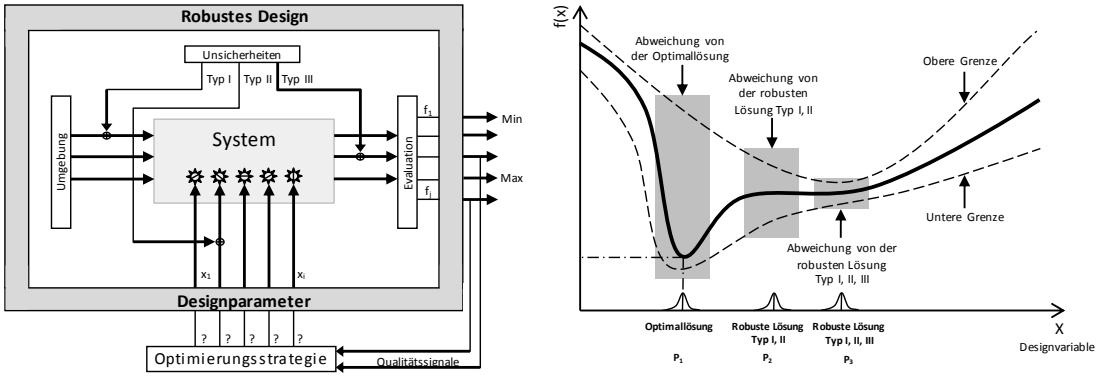
- **Typ III - Systemmodell:**
  Die Unsicherheiten vom Typ III unterscheiden sich von den Unsicherheiten der Systemumwelt und der Designvariablen. Resultierend aus dem Einsatz von Modellen, eingeschlossen der Gesamtheit aller Mess- und Modellierungsfehler, unterliegen die Evaluation des Systemergebnisses und der Systemperformance ebenfalls Unsicherheiten. Zusammenfassend wird das Ziel verfolgt, die (nicht) parametrierbare Variabilität der zugrundeliegenden Unsicherheiten der Modellparameter und ferner des Gesamtmodells zu minimieren [HS07].

  Die Unsicherheiten können unterschiedlichen Ursachen zugrunde liegen. Aufgrund von ungenauen Messdaten oder unzureichenden Erfahrungswerten bezüglich der Modellparameter können Unsicherheiten vorliegen, die sich negativ auf die Variabilität der Systemperformance auswirken. Zum anderen kann eine Modellunsicherheit bestehen, welche beispielsweise aus Modellvereinfachungen und Wahrscheinlichkeitsmodellen resultieren [Kiu89, Cho05].

- **Typ IV - Realisierbarkeit/Nebenbedingung:**
  In [PSP93] wird ein zusätzlicher Typ von Unsicherheiten vorgeschlagen, welcher die Einhaltung der Nebenbedingungen unter Variation der Designvariablen betrachtet. Resultierend werden die Effekte bezüglich des Designumfanges untersucht. Dieser Typ tritt häufig in Verbindung mit den Typen I und II auf [HS07].

  Die Einhaltung von Gleichheitsnebenbedingungen kann mit Hilfe von drei Ansätzen realisiert werden. Zum einen kann die Gleichheitsnebenbedingung weniger scharf formuliert werden [SR97]. Zum anderen besteht die Möglichkeit, die Einhaltung der Nebenbedingung mittels Wahrscheinlichkeiten zu definieren [SIH91]. Ferner kann, wenn möglich, die Nebenbedingung substituiert werden [Das00, ZFM05].

In Abbildung 3.2 (b) werden die Systemunsicherheiten anhand der Minimierung der Zielfunktion des zugehörigen Systemmodells zusammengefasst. Die optimale Lösung der Zielfunktion liegt dementsprechend im Minimum in P_1. Für eine möglichst robuste Lösung sollte eine möglichst geringe Verteilung der Performance vorliegen. In P_2 existiert eine flache Region mit einer geringeren Verteilung der Performance als in P_1. Die robuste Lösung für den Typ I und II ist demnach robuster gegenüber Veränderungen der Designvariablen, weist aber zudem eine geringere Performance auf. Die gestrichelten Linien, um die Zielfunktion, zeigen die Grenzen der Unsicherheiten bezüglich der Modellparameter an. Für den Typ III wird eine geringe Varianz der Lösungen in P_3 erwartet als in P_1 oder P_2 [Cho05].



(a) (b)

Abbildung 3.2.: Systemunsicherheiten Typ I bis III. Unter (a): Typ I Unsicherheiten bezüglich der Systemumwelt, Typ II Toleranzen der Designparameter und Typ III Unsicherheiten bezüglich des Systemmodells (angelehnt an [HS07]). Unter (b): Betrachtung der Systemunsicherheiten anhand des Systemergebnisses in Abhängigkeit einer Designvariable (angelehnt an [Cho05, CATM96]).

Für ein MA(R)S existieren verschiedene Ansätze, um die Robustheit zu steigern und den Umgang mit Unsicherheiten zu verbessern. Aus diesem Grund werden nachfolgend geeignete Maßnahmen diskutiert. Die Maßnahmen reichen von der Planung und Modellierung des MA(R)S über die verteilte Ressourcennutzung und Infrastruktur bis hin zu Kontrollinstanzen und Notfallmaßnahmen. Ferner werden die Maßnahmen, für die in Kapitel 5 vorgestellten MOO TCMs, integriert.

3.3.2. Maßnahmen zur Steigerung der Robustheit

Unvorhersehbare Veränderungen der Systemumwelt sowie Unsicherheiten können Ursachen dafür sein, dass ein System in fehlerhafte oder ineffiziente Zustände übergeht. Um diese Situation verhindern zu können, ist es notwendig, dass Maßnahmen zur Steigerung der Robustheit integriert werden. Hierfür werden vor allem die Überlegungen aus [Gri01, Suh05] erweitert und auf das TCP übertragen.

- **Übergeordneter Planungsprozess:**
  Ein übergeordneter Systemplanungsprozess im Sinne der Steigerung der Zuverlässigkeit und Stabilität kann sich positiv auf die Robustheit auswirken. Im ersten Schritt müssen die FR und Nebenbedingungen definiert werden, um den Designumfang vollständig bestimmen zu können. Hierbei werden im Idealfall nur unabhängige FR ermittelt. Für ein gekoppeltes Design werden die Auswirkungen bei Änderungen der FR identifiziert. Als nächstes wird der Systemumfang und die Designparameter definiert. der Systemumfang sollte nach Möglichkeit vollständig innerhalb des Designumfangs liegen [Suh05].

- **Robustes Design:**
  Das Ziel eines robusten Designs liegt darin, dass das Systemdesign einen größeren Variationsbereich der Designparameter, unter Einhaltung der FR, erfassen kann. Dieses Ziel kann für ein System mit Hilfe der Minimierung des Bias und der Varianz des Systems erreicht werden. Nur für den Idealfall, dass ein entkoppeltes Design vorliegt, kann der Bias für jede FR unab-

hängig der anderen FR durch Anpassung der spezifischen Designparameter gleich Null gesetzt werden. Die Varianz des Systemergebnisses resultiert aus den Unsicherheiten der Typen I bis III und demnach beispielsweise aus Störungen der Systemumwelt, ungenauen Messwerten oder Änderungen der Designparameter. Wird ein FR von verschiedenen Designparametern beeinflusst und bestehen ferner Abhängigkeiten zwischen den FR, dann sollte das Systemdesign derart modelliert werden, dass jede FR nur mittels eines spezifischen Designparameters kontrolliert werden kann und auf Veränderungen der anderen Designparameter nicht reagiert. Eine andere Möglichkeit die Varianz zu minimieren, besteht darin, die Designredundanzen zu minimieren. Für redundante oder weniger relevante Designparameter können statische Werte eingeführt werden, die folglich die Varianz der FR minimieren. Zusammenfassend folgt aus einer Minimierung des Bias und der Varianz des Systems eine Verminderung der Systemkomplexität und Unsicherheit darüber, ob die FR eingehalten werden können und sich das System in einem stabilen Zustand befindet [Suh05].

- **Integriertes Beobachtungssystem:**
  Die Integration eines Mess- und Beobachtungssystems ermöglicht es den Agenten, Änderungen bezüglich der Effizienz, Systemumwelt oder Systemzuständen zu identifizieren. Infolgedessen können Verhaltensanpassungen der Agenten vorgenommen werden. Das Beobachtungssystem kann zusätzlich um Kontrollschleifen erweitert werden, die eine dynamische Anpassung der Designparameter ermöglichen. Mit Hilfe dieser Fähigkeit kann eine diskrete Optimierung der Systemeigenschaften erreicht werden [Gri01].

  Beispielsweise können, bei der Verwendung von Heuristiken für das Routenplanungsproblem, die notwendigen Parameter dynamisch auf Grundlage der Messungen bezüglich der Systemumwelt bestimmt werden. Des Weiteren wird eine Identifikation von kritischen Zuständen ermöglicht, die auf einen bevorstehenden Ausfall eines Agenten oder fehlerbehaftete Informationen hinweisen können.

- **Zeitabhängig periodisches System:**
  Neben der Komplexitätsreduzierung wirken sich zeitabhängig periodische Systeme positiv auf die Robustheit aus. Werden Fehler und instabile Zustände nicht erkannt, dann können diese dauerhaft einen negativen Einfluss auf das System nehmen. Im schlimmsten Fall wird das System handlungsunfähig. Die Einführung von funktionaler Periodizität kann verhindern, dass sich das System außerhalb der Designgrenzen bewegt. Hierbei kann die funktionale Periodizität, beispielsweise auf zeitlichen Perioden oder Informationsprozessen basieren und folglich das System periodisch in einen stabilen Zustand zurücksetzen [Suh05].

  Für das TCP kann eine FR die Minimierung der Abdeckungszeit darstellen, dann ist demnach die FR eine Funktion der Planung der Agentenbewegung, beziehungsweise Reihenfolge der zu besuchenden Teilregionen der Systemumwelt. Die Reihenfolge der zu besuchenden Teilregionen steht in Abhängigkeit der bereits erfüllten Aufgaben. Hierbei nimmt, bei ungeeigneter Auswahl die Wahrscheinlichkeit eine gute Reihenfolge zu finden, kontinuierlich ab [Suh05]. Die Anzahl möglicher Kombinationen ist für das Gesamtproblem hoch und aufgrund von möglichen Unsicherheiten unter Umständen mit Ineffizienzen behaftet. Aus diesem Grund bietet sich die Einführung der Periodizität an. Hierbei ist die Möglichkeit gegeben, zu definierten Ereignissen eine Neuplanung der Aufgabenreihenfolge vorzunehmen. Die bereits vorausgeplante Aufgabenreihenfolge wird nicht weiter betrachtet. Die Periodizität kann bei-

spielsweise auf die einzelne Vergabe von Aufgaben reduziert werden. Resultierend wird das MA(R)S nach jeder Vergabe einer Aufgabe in einen stabilen Zustand zurückgesetzt.

- **Abstrahierte Notfallmaßnahmen:**
  Ungeachtet der Umsetzung von Strategien, die ein System robuster gestalten, kann es vorkommen, dass insbesondere komplexe Systeme aufgrund von Veränderungen der Systemumwelt in fehlerhafte Zustände übergehen. Im ersten Schritt muss das integrierte Beobachtungssystem diesen Fehlzustand erkennen. Darauffolgend können verschiedene Maßnahmen eingeleitet werden, um den Schaden am System zu minimieren. Zum einen können für eine Menge von Zuständen abstrahierte Notfallmaßnahmen eingeleitet werden. Eine drastische Notfallmaßnahme kann beispielsweise das Unterbrechen aller Aufgaben umfassen. Anschließend wird das System in den letzten bekannten stabilen Zustand zurückgesetzt. Für schwach gekoppelte Systeme können sich Fehlzustände weniger stark auswirken, wenn sich die Fehler nicht auf andere Komponenten übertragen [Gri01].

  Ähnlich dem Ansatz zeitabhängig periodischer Systeme kann ein kritischer Zustand, beispielsweise eine redundante Aufgabenbearbeitung oder Ausfall eines Agenten, zu einer Neuplanung der als nächstes zu bearbeiteten Aufgaben führen. Des Weiteren kann eine Kollision mit anderen Agenten oder Hindernisstrukturen durch Sensoren frühzeitig erkannt werden. Um Schäden an den Agenten zu verhindern kann es notwendig sein, dass die Agenten anhalten und eine Konfliktlösung herbeiführen.

- **Verteilte Infrastruktur:**
  Die Umsetzung einer verteilten Infrastruktur bietet mehrere Vorteile. Insbesondere wird eine effektive, effiziente sowie robuste Koordination, Aufgabenverteilung und die resultierende Zielerreichung ermöglicht. Zum einen können, aufgrund der dislozierten Agenten, mehr Informationen zeitgleich über die Systemumwelt gesammelt werden. Zum anderen können Fehler bei einzelnen Agenten besser kompensiert werden. Um die Vorteile der verteilten Infrastruktur sicherzustellen, bedarf das MAS einer angepassten und verteilten Kommunikation. Demnach ist ein einheitlicher Informationsstand bezüglich der Systemumwelt, der Systemzustände und der Aufgaben wesentlich, um die Vorteile einer verteilten Infrastruktur nutzen zu können. [GSKP07, Vel10].

  Für ein MAS wirkt sich eine verteilte Infrastruktur positiv auf eine robuste Kommunikation und Koordination sowie Aufgabenerfüllung aus. Informationen können innerhalb eines MANET über unterschiedliche Agenten übermittelt werden. Demnach können Informationen im MAS auch verteilt werden, wenn einzelne Agenten gestört sind oder ausfallen. Zusätzlich kann die Infrastruktur eines MAS durch weitere Agenten flexibel angepasst und erweitert werden.

- **Integrierte Selbstregulierung:**
  Insbesondere für dynamische und komplexe Systeme können Änderungen der Systemumwelt oder Störungen einen nachhaltigen Einfluss auf die Effizienz und Effektivität eines System nehmen. Aus diesem Grund sollten Mechanismen integriert sein, um das Verhalten der Agenten regulieren zu können. Neben der vollständig autonomen Selbstregulierung der Agenten kann die Regulierung der Agenten über eine gegenseitige Einflussnahme realisiert werden. Die Selbstregulierung kann Elemente der Selbstorganisation, der Selbstkonfiguration, der Selbst-

optimierung, des Selbstschutzes sowie der Selbstwiederherstellung umfassen [Fer94, SZVM11]. Um die Systemzustände und Effizienz der Agenten bewerten zu können, setzt die Selbstregulierung ein Mess- und Beobachtungssystem voraus.

Für ein MAS kann die Systemstruktur, insbesondere durch die zielgerichtete Zuweisung der verschiedenen Rollen und Aufgaben, organisiert werden. Mit Hilfe der Selbstkonfiguration können beispielsweise zuvor definierte Rollen, Sensoren und Effektoren derart geändert werden, dass die Agenten befähigt sind in der Systemumwelt agieren zu können. Die Selbstoptimierung hingegen hat zum Ziel, die Systemparameter hinsichtlich der Effizienz und Robustheit anzupassen. Der Selbstschutz ist die Fähigkeit, auftretende Fehler in einem MAS zu isolieren und anschließend zu beheben, um eine effektive Ausführungen der Agenten nicht zu gefährden. Sollten die Agenten oder das gesamte MAS in einen instabilen Systemzustand übergehen, kann mittels der Selbstwiederherstellung eine Rückführung in einen stabilen Systemzustand durchgeführt werden.

- **Flexibles Planungssystem:**
  Basierend auf dem Beobachtungssystem kann ein flexibles Planungssystem integriert werden. Vor allem aufgrund der unvollständigen Informationen oder der rauschenden Sensormessungen liegen Unsicherheiten bezüglich der Zielerreichung und der daraus resultierenden Optimierungsprobleme vor. Mit Hilfe eines flexiblen Planungssystem kann die Aufgabenplanung der Agenten situationsabhängig erfolgen. Ferner können zu einem späteren Zeitpunkt zuvor getroffene Entscheidungen als suboptimal identifiziert werden. Den Agenten muss es daher möglich sein, eine wiederholte Planung vorzunehmen [SB05].

  Für das TCP sollten wenn möglich spezifische Situationen und Ereignisse definiert werden, die eine Neuplanung von Routen sowie Aufgabenverteilung auslösen. Die Definition von Ereignissen ist den diskreten Zeitintervallen zu bevorzugen. Zum einen werden Ressourcen für die Neuplanung gebunden und zum anderen können Redundanzen resultieren, die ohne eine qualitative Verbesserung der Lösung einhergehen.

- **Ausreichende Ressourcenversorgung:**
  Systeme neigen dazu, in einen instabilen Zustand überzugehen, wenn Sie einen gewissen Schwellenwert bei der Nutzung der zur Verfügung stehenden Ressourcen überschreiten. Die Lösung von Problemen bezüglich der Aufgabenverteilung und Routenplanung können unter Umständen nicht zeitgerecht bearbeitet werden. In Folge von Fehlentscheidungen und Verzögerungen können negative Auswirkungen bezüglich der Effizienz und Robustheit auftreten, welche sich für zeitabhängige Systeme zudem akkumulieren können [Gri01].

  Für das TCP wirkt sich eine gleichmäßige Ressourcenverteilung über alle Agenten positiv aus. Hierbei sollten alle Agenten über die notwendigen Sensoren und Effektoren verfügen. Resultierend können zum einen die Aufgaben von allen Agenten im gleichen Ausmaß bearbeitet werden und es entstehen keine beziehungsweise weniger Verzögerungen aufgrund fehlender Ressourcen. Zum anderen wirkt sich ein Ausfall einzelner Agenten weniger stark auf die Effizienz und Robustheit aus.

- **Übergeordnete Ressourcenkontrollinstanz:**
  Mit Hilfe einer übergeordneten Kontrollinstanz kann der Übergang in einen instabilen Systemzustand verhindert werden. Vor allem das Überschreiten der vorhandenen Ressourcenversorgung kann zu einem instabilen Systemzustand führen. Aus diesem Grund sollte vor der Ausführung von Aufgaben geprüft werden, wieviele Ressourcen gebunden sein werden. Dadurch können Aufgaben, die zum aktuellen Zeitpunkt die vorhanden Ressourcen übersteigen würden, zurückgestellt werden. Die Kontrollinstanz kann somit auch zur Effizienzsteigerung beitragen [Gri01].

  Stehen aus einem Grund nicht genügend Ressourcen für die Aufgabenbearbeitung zur Verfügung, muss es den Agenten möglich sein, diesen Zustand frühzeitig erkennen zu können, um Verzögerungen bei der Abdeckung zu minimieren. Wird die Ressourcenkontrollinstanz in Verbindung mit dem Planungssystem eingesetzt, können die Agenten mittels der Position in der Systemumwelt für nachfolgende Aufgaben geeigneter koordiniert werden.

- **Verteilte Ressourcennutzung:**
  Für ein dezentrales und hybrides System kann die Gesamtheit der verteilten Agentenressourcen genutzt werden, um für ein Problem möglichst schnell eine geeignete Lösung zu ermitteln. Der Vorteil liegt darin begründet, dass ein Optimierungsproblem in Teilprobleme zerlegt werden kann und diese parallel bearbeitet werden.

  Für das TCP kann beispielsweise die Routenplanungen durch jeden Agenten zeitgleich ermittelt werden. Trotz Fehlern oder des Ausfalls einzelner Ressourcen kann durch dieses Vorgehen eine Lösung ermittelt werden.

Bei der Abdeckung einer unbekannten Systemumgebung mit Hilfe von einem MA(R)S können Unsicherheiten und Fehler auftreten, die sich entweder auf die Effizienz der Agenten auswirken oder gar zum Totalausfall führen können. Bei der Implementierung von einem MA(R)S sollte daher auf die Ausprägung von einem robusten Verhalten geachtet werden. Hierfür können verschiedene Maßnahmen im Rahmen des TCP umgesetzt werden. Darüber hinaus sollten Unsicherheiten bezüglich der unbekannten Systemumgebung in der Routenplanung geeignet berücksichtigen werden. In dem sich anschließenden Kapitel 4 wird das SSPP in ein mehrkriterielles Optimierungsproblem transformiert. Entsprechend der Annahmen für die Routenplanung, im Rahmen des TCP, wird eine bestehende Heuristik adaptiert sowie ein geeignetes Heuristik-Parameterset ermittelt. In Kapitel 5 dieser Dissertation werden die Erkenntnisse bezüglich des SSPP auf das TCP übertragen und zusammen mit den Maßnahmen zur Steigerung der Robustheit in einem neuen TCM integriert. Hierbei wird erstmals eine Routenplanung unter Unsicherheiten für das TCP untersucht.

4. Das Shortest Path Problem

Viele sind hartnäckig in Bezug auf den einmal eingeschlagenen Weg,
wenige in Bezug auf das Ziel.
Friedrich Wilhelm Nietzsche (1844–1900)

In [Kle64] stellt Klee eine anschauliche Lösung für das Problem der Suche nach dem kürzesten Pfad dar. In einem Modell werden die Knoten und Kanten eines Graphen mittels Perlen und Fäden dargestellt. Die Länge der Fäden sind proportional bezüglich der jeweiligen Kosten auf den Kanten. Wird nun das Modell an zwei verschiedenen Perlen auseinandergezogen, stellt die Kette, welche ein weiteres Auseinanderziehen verhindert, zugleich den kürzesten Pfad dar.

In diesem Kapitel wird zuerst das Problem des kürzesten Pfades unter Unsicherheiten (SSPP[1]) eingeführt und verschiedene Definitionen für einen robusten Pfad angegeben. Daran anschließend wird eine Transformation des SSPP in ein Mehrkriterielles Optimierungsproblem vorgenommen, um eine alternative Lösung zu beschreiben. Der neue Lösungsansatz kann für Anwendungsgebiete verschiedene Vorteile bieten. Hierbei werden die Grundlagen der Mehrkriteriellen Optimierung eingeführt und die Metaheuristik der Ant Colony Optimization beschrieben. Anhand der BiCriterionAnt Heuristik wird eine experimentelle Untersuchung des neuen Lösungsansatzes durchgeführt.

Für die nachfolgenden Probleme sei stets ein gerichteter Graph $G = (V, E, C)$, mit einer endlichen Menge von Knoten $V = \{1, \ldots, n\}$ und einer endlichen Menge von Kanten mit $E \subseteq \{(i, j) | i, j = 1, \ldots, n; i \neq j; i, j \in V\}$ gegeben. Die Kostenmatrix C wird auf den Kanten $(i, j) \in E$ definiert.

Ferner ist das klassische, deterministische Optimierungsproblem durch deterministische Übergänge von einem zulässigen Zustand $\mathfrak{z} \in \mathfrak{Z}$ in einen Folgezustand $f(\mathfrak{z}, \mathfrak{a})$, für eine Aktion $\mathfrak{a} \in \mathfrak{A}(\mathfrak{z})$ gekennzeichnet. Der Entscheidungsträger verfolgt das Ziel, eine Sequenz $(\mathfrak{a}_0, \mathfrak{a}_1, \ldots, \mathfrak{a}_n)$ von zulässigen Aktionen zu ermitteln, welche einen Anfangszustand $\mathfrak{z}_S$ in einen Zielzustand $\mathfrak{z}_G$ mit $(\mathfrak{z}_0 = \mathfrak{z}_S, \mathfrak{a}_0, \mathfrak{z}_1, \mathfrak{a}_1, \ldots, \mathfrak{a}_n, \mathfrak{z}_{n+1} = \mathfrak{z}_G)$, $\forall \mathfrak{a}_i \in \mathfrak{A}(\mathfrak{z}_i) \wedge \mathfrak{z}_{i+1} = f(\mathfrak{z}_i, \mathfrak{a}_i)$ überführt. In der Regel wird ein optimaler Plan unter allen zulässigen Sequenzen gesucht, für die die Kosten $\sum_{i=0}^{n} c(\mathfrak{z}_i, \mathfrak{a}_i)$ minimiert werden [BH10].

Entsprechend wird für das SPP auf dem Graphen G eine Sequenz von Knoten und Kanten $(v_S, e_{S,i}, v_i, e_{i,n}, \ldots, e_{m,G}, v_G)$, $\forall e_{i,j} \in E(v_i)$ mit einem Startknoten v_S sowie Zielknoten v_G ermittelt derart, dass die Kosten für die resultierende Kantenfolge beziehungsweise den Pfad minimiert werden.

Das SPP gehört zu den am Besten untersuchten kombinatorischen Optimierungsproblemen. Effiziente Algorithmen zum Lösen des SPP werden bereits von Dijkstra in [Dij59] mit einer Laufzeit

[1] Eng. Stochastic Shortest Path Problem

von $O(|V|^2)$, Hart et al. [HNR68] ebenfalls mit $O(|V|^2)$, Bellmann und Ford in [For56, Bel58] mit $O(|V||E|)$ oder Floyd und Warshall in [Flo62, War62] mit $O(|V|^3)$ angegeben.

4.1. Das Single Objective Shortest Path Problem unter Unsicherheiten

Eine anschauliche Darstellung für das SSPP anzugeben ist ungleich schwerer als die Anschauung des Problems ohne Unsicherheiten. Für das SSPP sind die Kosten mit Unsicherheiten behaftet. Eine mögliche Darstellung kann wie folgt beschrieben werden: Für jede Kante wird ein Intervall mit einer zugehörigen Wahrscheinlichkeitsverteilung definiert. Die Länge der Fäden wird individuell für jedes Intervall per Zufall bestimmt und anschließend wird der kürzeste Pfad ermittelt. Der Versuchsaufbau wird n-mal wiederholt und ein Entscheidungsträger wählt einen der ermittelten Pfade aus. Hierbei treten zugleich zwei Probleme auf. Zum einen müsste n sehr groß gewählt werden, um möglichst viele verschiedene Kombinationen erfassen zu können. Zum anderen muss eine Definition vorliegen, unter welchen Kriterien die Auswahl des geeignetsten Pfades erfolgt.

Für Probleme der realen Welt liegen in der Regel Unsicherheiten bezüglich der Modelldaten vor, die negative Auswirkungen auf die Lösung eines Optimierungsproblems nehmen können [BBC11]. Hierbei können verschiedene Gründe, wie zum Beispiel die Prozesse der Datenmessung, der Datensammlung oder der Datenvorverarbeitung, für die Ausprägung von Unsicherheiten vorliegen. Aufgrund von unzureichendem Wissen und der unkontrollierbaren Systemumwelt können Annahmen und Nebenbedingungen mit Unsicherheiten behaftet sein. Zudem können die Verfahren der Komplexitätsreduzierung zu Unsicherheiten der Modelldaten führen. Einen wesentlichen Einfluss nimmt zusätzlich die Ungewissheit bezüglich der Auswirkungen zukünftiger Entscheidungen ein [HS07]. In Kapitel 3.3 wurden die Ursachen von möglichen Unsicherheiten für ein MARS dargestellt, welche ebenfalls für eine Betrachtung des SSPP auftreten können.

Die Unsicherheiten können auf unterschiedliche Weise modelliert werden. Grundsätzlich ist eine Definition der Unsicherheiten auf den Knoten V und den Kanten E im Graphen G möglich. Typische Modellierungen beschreiben die Unsicherheiten auf den Zustandsübergangsfunktionen. Resultierend liegt ein nicht deterministischer Übergang von einem zulässigen Zustand $\mathfrak{z} \in \mathfrak{Z}$ in einen Folgezustand $f(\mathfrak{z}'|\mathfrak{z}, \mathfrak{a})$, für eine Aktion $\mathfrak{a} \in \mathfrak{A}(\mathfrak{z})$ vor. Mit Hilfe einer stochastischen Zustandsübergangsfunktion $f(\cdot|\mathfrak{z}, \mathfrak{a})$ kann eine Wahrscheinlichkeit für den Übergang in einen Folgezustand $\mathfrak{z}'$ in Abhängigkeit der Aktion $\mathfrak{a}$ angegeben werden. Entsprechend sind die Anzahl der benötigten Zustandsübergange a priori nicht eindeutig ermittelbar. Hingegen werden bei einer Modellierung der Unsicherheiten mittels einer stochastischen Kostenfunktion die erwarteten Kosten für einen Zustandsübergang angegeben. Im Nachfolgenden werden die Unsicherheiten stets über die erwarteten Kosten $c(\mathfrak{z}, \mathfrak{a})$ betrachtet [BH10].

Die allgemeine Formulierung für das robuste Optimierungsproblem kann wie folgt angegeben werden [BBC11]:

$$\min f_0(\mathbf{x}) \tag{4.1}$$

Unter der Nebenbindung:

$$f_i(\mathbf{x},\mathbf{u}_i) \leq 0,\ \forall \mathbf{u}_i \in U_i, i = 1, \ldots, m \tag{4.2}$$

Für dieses Optimierungsproblem ist $\mathbf{x} \in \mathbb{R}$ der Vektor für die Entscheidungsvariablen, $\mathbf{u}_i \in \mathbb{R}$ sind die Störvektoren beziehungsweise die Parameterunsicherheiten und $U_i \subseteq \mathbb{R}$ die zugehörigen $i = 1, \ldots, m$ verschiedenen m-Mengen der definierten Unsicherheiten. Für (4.1) wird das Ziel verfolgt eine Minimierung von f_0 durchzuführen, unter der $\mathbf{x}^*$ zulässig für alle Realisierungen der Unsicherheiten $\mathbf{u}_i \in U_i$ ist. Demnach wird eine Worst-Case Optimierung betrachtet, für die gilt, dass $\mathbf{x}^*$ auch für den ungünstigsten Fall am geeignetsten ist. Unsicherheiten können sowohl in der Zielfunktion als auch den Nebenbedingungen auftreten. Gilt für $|U_i| = 1$, dass heißt es liegt genau ein eindeutiges Szenario vor, dann unterliegt die zugehörige Nebenbedingung keinen Unsicherheiten. Die Modellierung der Unsicherheitsmengen U_i kann verschiedene Ausprägungen annehmen. In der Literatur werden allgemein eine szenariobasierte, intervallbasierte, ellipsoidale und polyhedrale Modellierungen vorgeschlagen.

In der Literatur werden für das SSPP verschiedene Modellierungen für die Unsicherheiten untersucht [Sah04, GM07]. Für das Szenariodesign-Problem wird für jede Unsicherheit eine endliche Menge von diskreten Szenarien definiert. Eine zulässige Lösung des Problems liegt demnach in der endlichen Lösungsmenge der Kombination von unabhängigen Szenarien. Eine weitere Möglichkeit besteht in der Beschreibung der Unsicherheiten durch Intervalle. Hierbei werden Intervalle dazu genutzt, um die maximale und minimale Ausprägung von Eigenschaften zu definieren. Für das stochastische Optimierungsproblem hingegen werden die Unsicherheiten mittels diskreter oder kontinuierlicher Zufallsvariablen mit zugehöriger Wahrscheinlichkeitsverteilung beschrieben. Des Weiteren können die Unsicherheiten für das Soft-Constraint Optimierungsproblem mit Hilfe von Fuzzy Mengen modelliert werden.

Ein der robusten Optimierung unter Unsicherheiten ähnliches Problem stellt die Unsicherheitsanalyse dar. Anders als bei der robusten Optimierung, für die die Unsicherheiten der Einflussfaktoren bereits vor der Problemlösung betrachtet werden, wird bei der Unsicherheitsanalyse das Ausmaß der Unsicherheiten für den Zielfunktionswert in Abhängigkeit der verschiedenen Unsicherheiten, welche bei den Eingabewerten und Einflussfaktoren auftreten können, quantifiziert. Eine Methode der Unsicherheitsanalyse stellt die Sensitivitätsanalyse dar, welche die Variation des Zielfunktionswertes unter systematischer Veränderung der Eingabewerte misst [STCR04]. Bei der Monte-Carlo-Simulation werden zufallsbasierte Stichproben auf Grundlage von Wahrscheinlichkeitsverteilungen für die Eingabewerte gezogen, um die Unsicherheiten zu quantifizieren [Hel93].

Für das generalisierte SSPP können die unsicheren Kosten mittels verschiedener Beschreibungen vorgenommen werden. In dieser Dissertation werden die unsicheren Kosten gemäß [BT91, Jam06] mit Hilfe einer Wahrscheinlichkeitsverteilung stochastisch beschrieben werden. Demzufolge sind die Kosten $c_{i,j}$ auf der Grundlage einer Zufallsgröße definiert und werden für einen gerichteten Graphen $G = (V, E, C)$ mittels einer $|V| \times |V|$ Matrix C über die Wahrscheinlichkeitsdichtefunktion $c_{i,j}$ einer jeden Kante $e_{i,j}$ modelliert. Hierbei spiegelt $c_{i,j}$ die wahrscheinlichsten Kosten wider und basiert auf a priori Expertenwissen oder Beobachtungen. Es wird der kürzeste Pfad $p*$, zwischen einem Startknoten v_s und einem Zielknoten v_t, mit den erwarteten Kosten $\widetilde{c}_{p*} = \min_{p \in P} \widetilde{c}_p$ gesucht. Die Menge P beinhaltet alle voneinander verschiedenen Pfade zwischen v_S und v_G. Das SSPP lässt sich für den Fall, dass für jede Kante eine zugehörige Zufallsvariable vorliegt, die genau einen Wert mit der Wahrscheinlichkeit gleich eins annimmt, auf das deterministische SPP Problem reduzieren.

In [KY97, GM07, MG08] werden verschiedene Maße für die Robustheit auf das SSPP übertragen, um eine Bewertung der ermittelten Pfade durchführen zu können. Einfach gesprochen liegt

ein robuster Pfad genau dann vor, wenn dieser für möglichst viele verschiedene Realisierungen des untersuchten Graphen G einen geeigneten Pfad darstellt. Die Realisierungen beziehungsweise Szenarien können wie folgt definiert werden.

Definition 4.1 (Szenario)**.** : *Ein Szenario $\mathfrak{s} \in \mathfrak{S}$ ist eine zulässige Realisierung der Kosten $c(e_{i,j})$ über alle Kanten $e_{i,j} \in E$ im Graphen G.*

Um einen Vergleich von Alternativen durchführen zu können, kann für die robuste Optimierung der maximal, zu erwartende Verlust untersucht werden. Entsprechend wird der ungünstigste Fall, für den der größte Verlust einer Alternative vorliegt, wie nachfolgend beschrieben.

Definition 4.2 (Kriterium - ungünstigster Fall)**.** : *Für einen Pfad p liegt der ungünstigste Fall $\mathfrak{f}_{wor}(p)$ genau für das Szenario $\mathfrak{s} \in \mathfrak{S}$ vor, für den die Kosten $c(p)$ maximiert werden.*

$$\mathfrak{f}_{wor}(p) = \max_{\mathfrak{s} \in \mathfrak{S}} c^{\mathfrak{s}}(p) \tag{4.3}$$

Resultierend wird der absolut robuste Pfad bestimmt.

Definition 4.3 (Absolut robuster Pfad)**.** : *Ein Pfad $p^* \in P_{S,G} \subseteq P$ von v_S zu v_G wird genau dann als absolut robuster Pfad bezeichnet, wenn dieser das kleinste (über alle Pfade von v_S zu v_G) Maximum (über alle zulässigen Szenarien $\mathfrak{s} \in \mathfrak{S}$) bezüglich der Kosten aufweist.*

$$p^* = \min_{p \in P_{S,G}} \mathfrak{f}_{wor}(p) \tag{4.4}$$

Alternativ kann der geringste Opportunitätsverlust untersucht werden. Das heißt, es wird die Alternative gesucht, die den maximalen Verlust gegenüber der jeweils besten Lösung über alle Szenarien minimiert. Hierfür wird die robuste Abweichung zugrunde gelegt.

Definition 4.4 (Kriterium - maximal/relativ robuste Abweichung)**.** : *Für ein konkretes Szenario $\mathfrak{s}$ wird die robuste Abweichung für einen Pfad p von v_S zu v_G durch die Differenz zwischen den Kosten für einen Pfad p in $\mathfrak{s}$ und den Kosten für den besten Pfad p' von v_S zu v_G in $\mathfrak{s}$ ermittelt.*

Die maximal robuste Abweichung für einen Pfad p über alle Szenarien $\mathfrak{s} \in \mathfrak{S}$ definiert sich zu:

$$\mathfrak{f}_{reg}(p) = \max_{\mathfrak{s} \in \mathfrak{S}} (c^{\mathfrak{s}}(p) - c^{\mathfrak{s}}(p')) \tag{4.5}$$

Die relativ robuste Abweichung für einen Pfad p über alle Szenarien $\mathfrak{s} \in \mathfrak{S}$ definiert sich über die Normierung der maximal robusten Abweichung:

$$\mathfrak{f}_{relreg}(p) = \max_{\mathfrak{s} \in \mathfrak{S}} \frac{c^{\mathfrak{s}}(p) - c^{\mathfrak{s}}(p')}{c^{\mathfrak{s}}(p')} \tag{4.6}$$

Es folgt die Definition für den robust abweichenden Pfad.

Definition 4.5 ((Relativ) robust abweichender Pfad)**.** : *Ein Pfad p^* von v_S zu v_G wird genau dann als robust abweichender Pfad bezeichnet, wenn dieser das kleinste (über alle Pfade von v_S zu v_G) Maximum (über alle zulässigen Szenarien $\mathfrak{s} \in \mathfrak{S}$) bezüglich der robusten Abweichung aufweist.*

Entsprechend wird mit diesem Pfad auch das geringste Bedauern, beziehungsweise der geringste Opportunitätsverlust assoziiert.

$$p^* = \min_{p \in P_{S,G}} \mathfrak{f}_{reg}(p) \tag{4.7}$$

Alternativ kann der relativ robust abweichende Pfad p^* angegeben werden.

$$p^* = \min_{p \in P_{S,G}} \mathfrak{f}_{relreg}(p) \tag{4.8}$$

Im nachfolgenden wird eine neue Methode vorgestellt, um eine geeignete Lösung des SSPP zu ermitteln. Hierbei wird das SSPP auf Grundlage einer mehrkriteriellen Betrachtung untersucht. Nach einer allgemeinen Einführung in die MOO wird das mehrkriterielle SPP (MOSPP[2]) definiert. Daran anschließend wird eine mögliche Lösung vorgestellt und eine Rückführung auf das SSPP durchgeführt. Abschließend wird eine Bewertung der neuen Methode diskutiert.

4.2. Eine Transformation in das Multiobjective Shortest Path Problem (unter Unsicherheiten)

Für das SSPP werden in der Regel die Unsicherheiten in den Nebenbedingungen modelliert. Entsprechend wird unter Betrachtung der Zielfunktion genau eine zulässige und robuste Lösung für das untersuchte Problem ermittelt. Es wird versucht, die robusteste Lösung unter Beachtung aller möglichen Szenarien zu bestimmen. Hierbei ist es oftmals nur schwer möglich, die Zusammenhänge verstehen und Rückschlüsse auf die Auswahl der ermittelten Lösung ziehen zu können. Des Weiteren erhält der DM keine Informationen darüber, wie sich die Robustheit der ermittelten Lösung zu anderen zulässigen Lösungen verhält.

Nachfolgend wird eine neue Herangehensweise zur Bestimmung von robusten Lösungen vorgestellt. Statt die Unsicherheiten in den Nebenbedingungen zu modellieren, werden die Unsicherheiten in einer zusätzlichen Zielfunktion definiert. Daraus folgt für das einkriterielle Optimierungsproblem unter Unsicherheiten das bikriterielle Optimierungsproblem. Entsprechend wird neben der problemspezifischen Zielfunktion des SSPP die Minimierung der Unsicherheiten als eine weitere Zielfunktion betrachtet. Aufgrund dieser Transformation kann der DM die Vorteile der mehrkriteriellen Optimierung nutzen. Hierbei nimmt die Bestimmung der Menge von Pareto-optimalen Kompromisslösungen einen positiven Einfluss. Resultierend erhält der DM eine Menge an Kompromisslösungen, die in den Extremfällen jeweils Lösungen unter Betrachtung genau einer Zielfunktion beinhaltet. Des Weiteren enthält die Menge an Kompromisslösungen auch jene, die unter Betrachtung aller Zielfunktionen optimierte Kompromisslösungen darstellen. Derart entwickelt der DM ein besseres Verständnis für den Lösungsraum und ferner für den Entscheidungsraum. Zudem ist die Lösung des mehrkriteriellen Optimierungsproblems mit Hilfe von etablierten und effizienten Heuristiken möglichen.

[2]Engl. Multiobjective Shortest Path Problem

4.2.1. Eine Einführung in das Mehrkriterielle Optimierungsproblem

Ein Vielzahl von Problemen in den unterschiedlichsten Wissenschaftsbereichen lassen sich nicht mit Hilfe einer singulären Zielfunktion beschreiben. Aufgrund von konfliktären Zielen die gleichzeitig betrachtet werden müssen, existiert oftmals keine eindeutig beste Lösung. Beispielsweise können für autonome Fahrzeuge geeignete Routen gesucht werden, die eine möglichst kurze Fahrtdauer aufweisen, zudem aber auch einen geringen Energieverbrauch verursachen. Resultierend können die Extremlösungen für die Routen mit der kürzesten Fahrtdauer als auch mit dem geringsten Energieverbrauch bestimmt werden. Zwischen den Extremlösungen existieren weitere Kompromisslösungen, für die es keine anderen eindeutig besseren Lösungen gibt. Aus diesem Grund sollten sie für das Problem ebenfalls betrachtet werden. Des Weiteren müssen die Präferenzen des DM in die Lösungsfindung einbezogen werden, insbesondere wenn mehrere Lösung mit gleicher Güte vorliegen.

Bezüglich der Einbindung des DM können können drei grundlegende Verfahrensweisen unterschieden werden, um eine geeignete Lösung für Probleme der MOO zu ermitteln [HM79].

- **Entscheidungsfindung vor der Lösungssuche:**
  Ein intuitiver Ansatz zur Lösung des mehrkriteriellen Optimierungproblems (MOOP[3]) besteht in der Transformation in das einkriterielle Optimierungsproblem (SOOP[4]). Über eine Gewichtung der Zielfunktionen wird eine gemeinsame Zielfunktion erzeugt. Anschließend kann das SOOP mit unterschiedlichen Gewichtungen untersucht werden, um verschiedene Lösungen zu approximieren. Des Weiteren besteht die Möglichkeit, eine Transformation von Zielfunktionen in Nebenbedingungen vorzunehmen. Entsprechend wird für eine ausgewählte Zielfunktion und der transformierten Nebenbedingung das SOOP untersucht.

- **Entscheidungsfindung während der Lösungssuche:**
  Die Suche nach Lösungen stellt einen interaktiven Prozess dar. Hierbei werden dem DM nach jedem Optimierungsschritt alternative Lösungen aus verschiedenen Regionen des MOOP zugehörigen Lösungsraumes präsentiert. Die Präferenzen des DM werden kontinuierlich für die Suche nach weiteren Lösungen genutzt.

- **Entscheidungsfindung nach der Lösungssuche:**
  Die Suche nach Lösungen wird unabhängig von den Präferenzen des DM durchgeführt. Hierbei werden Pareto-optimale Lösungen gesucht, die gute Kompromisslösungen für das MOOP darstellen. Im Anschluss an die Lösungssuche, muss der DM eine Lösung aus der Menge der Pareto-optimalen Lösungen auswählen.

Für den Ansatz einer aggregierten Zielfunktion heraus resultieren Nachteile. Grundsätzlich bestehen, aufgrund der Unvergleichbarkeit von konfliktären Zielfunktionen, Schwierigkeiten eine gemeinsame, zusammengesetzte Zielfunktion zu definieren. Des Weiteren muss der DM bereits im Vorfeld der Optimierung seine Präferenzen bezüglich der Zielfunktionen kennen. Zusätzlich verändert eine Transformation des MOOP in das SOOP die Eigenschaften, Besonderheiten und Struktur des zugrundeliegenden Optimierungsproblem derart, dass der Zielfunktionsraum verändert wird. Folglich können zulässige Lösungen durch die Transformation des MOOP in das SOOP verloren

[3] Engl. Multiobjective Optimization Problem
[4] Engl. Singleobjective Optimization Problem

gehen [AB11, Zit99]. Sowohl für die Transformation in das SOOP als auch für den Ansatz der Transformation von Zielfunktionen in Nebenbedingungen können nur Lösungen in nicht konvexen Regionen der zugehörigen Kurve von Kompromisslösungen ermittelt werden [DD97]. Aufgrund der resultierenden Nachteile einer aggregierten Zielfunktion wird dieses Vorgehen in dieser Dissertation nicht weiter betrachtet.

Des Weiteren ist der interaktive Ansatz für autonome Agenten im Rahmen des Terrain Covering eher ungeeignet. Aufgrund zusätzlicher Entscheidungsprozesse und dynamischer Anpassungen sowie eines erhöhten Koordinierungs- und Kooperationsaufwandes zwischen den autonomen Agenten nimmt die Komplexität zu. In dieser Dissertation wird das Vorgehen mit einer Entscheidungsfindung im Anschluss an die Lösungssuche untersucht. Vorteilhaft wirkt sich hierbei der ergebnisoffene Ansatz der Lösungsfindung aus. Mit Hilfe von Lösungsverfahren wird beabsichtigt eine Untersuchung des gesamten Lösungsraumes vorzunehmen. Jeder autonome Agent wählt situationsbedingt die geeignetste Kompromisslösung aus. Das weitere Vorgehen wird erst auf Grundlage der Lösungsfindung mit den anderen Agenten abgestimmt. Entsprechend basiert die übergeordnete Entscheidungsfindung zwischen den Agenten auf bereits geeigneten Lösungen der individuellen Teilprobleme.

Das in dieser Dissertation untersuchte MOOP kann als Quintupel $(\mathbf{x}, \mathbf{d}, \Lambda, \mathbf{f}, \mathbf{R})$ mit

- einem n-dimensionalen Vektor $\mathbf{x}$ von Entscheidungsvariablen $\mathbf{x} = (x_1, \ldots, x_n)^T$,
- einem k-dimensionalen Vektor $\mathbf{d}$ von zugehörigen Definitionsbereichen der Entscheidungsvariablen $\mathbf{d} = (d_1, \ldots, d_k)^T$,
- einer Menge von Nebenbedingungen Λ, die den zulässigen Bereich der Entscheidungsvariablen $\mathbf{x}$ im Definitionsbereich einschränkt,
- einem k-dimensionalen Vektor $\mathbf{f}$ von Zielfunktionen $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x}))^T$ mit $k \geq 2$
- einer Ordnungsrelation $\mathbf{R} \subseteq \mathbb{N}^k \times \mathbb{N}^k$ für die, für zwei Ausprägungen des Vektors der Entscheidungsvariablen $\mathbf{x'}, \mathbf{x''}$, nachfolgend $(f_1(\mathbf{x'}), \ldots, f_k(\mathbf{x'}))\mathbf{R}(f_1(\mathbf{x''}), \ldots, f_k(\mathbf{x''})) \Leftrightarrow f_1(\mathbf{x'})\mathbf{R}^1 f_1(\mathbf{x''}) \wedge \ldots \wedge f_k(\mathbf{x'})\mathbf{R}^k f_k(\mathbf{x''})$ gilt. Für das Minimierungsproblem (Maximierungsproblem) ist $\mathbf{R}^i$ mit $\leq$ $(\geq)$ definiert.

beschrieben werden.

Die Pareto Optimierung basiert grundlegend auf dem Konzept der Ordnungsrelation, um Lösungen miteinander vergleichen zu können [Neh03, Beu06, Beu09, LGCM10]. Die nachfolgenden Ausführungen sind für das Minimierungsproblem dargestellt.

Definition 4.6 (Schwache/starke Ordnungsrelation)**.** : *Die schwache und starke Ordnung $\boldsymbol{R}$ einer Menge B beschreiben binäre Ordnungsrelationen $\boldsymbol{R} \subseteq B \times B$. Hierbei gilt für die schwache Ordnungsrelation $\boldsymbol{R}$ mit $b \leq b'$ und stellt eine Partialordnung dar. Ferner liegen die Eigenschaften der Reflexivität $(b, b) \in \boldsymbol{R}$ für $\forall b \in B$, der Antisymmetrie $(b, b') \in \boldsymbol{R} \wedge (b', b) \in \boldsymbol{R} \Rightarrow b = b'$ für $\forall b, b' \in A$ und Transitivität $(b, b') \in \boldsymbol{R} \wedge (b', b'') \in \boldsymbol{R} \Rightarrow (b, b'') \in \boldsymbol{R}$ für $\forall b, b', b'' \in B$ vor. Hingegen stellt die starke Ordnungsrelation $\boldsymbol{R}$ mit $b < b'$ eine Striktordnung auf der Menge B dar. Hierfür entfällt die Eigenschaft der Antisymmetrie und statt der Reflexivität gilt die Irreflexivität mit $(b, b) \notin \boldsymbol{R}$ für $\forall b \in B$.*

Die schwache und starke Ordnungsrelation muss für die mehrkriterielle Optimierung um einen komponentenweisen Vergleich der Zielfunktionsvektoren erweitert werden. Entsprechend kann für die k-dimensionalen Vektoren **b** und **b'** eine komponentenweise, schwache Ordnungsrelation **R** mit $\mathbf{b} \leq \mathbf{b'} \Leftrightarrow (b_1, b_2, \ldots, b_k)^T \leq^k (b'_1, b'_2, \ldots, b'_k)^T$ definiert werden. Für eine komponentenweise, starke Ordnungsrelation **R** gilt $\mathbf{b} < \mathbf{b'} \Leftrightarrow (b_1, b_2, \ldots, b_k)^T \leq^k (b'_1, b'_2, \ldots, b'_k)^T \wedge \exists i \in \{1, \ldots, k\} | b_i < b'_i$. Eine komponentenweise, strikte Ordnungsrelation **R** gilt, genau dann wenn $\mathbf{b} < \mathbf{b'} \Leftrightarrow (b_1, b_2, \ldots, b_k)^T <<^k (b'_1, b'_2, \ldots, b'_k)^T$.

Resultierend kann für das mehrkriterielle Optimierungsproblem, mit Hilfe von komponentenweisen Ordnungsrelationen, eine Ordnung über die Präferenzen des DM im Zielfunktionsraum angegeben werden. Die Dominanzrelation der (schwachen und starken) Pareto-optimalität für den Entscheidungsraum X und äquivalent für den Zielfunktionsraum $Y = f(X)$ wird wie folgt definiert.

Definition 4.7 (Schwach dominierte Lösung)**.** : *Für zwei k-dimensionale Vektoren $\boldsymbol{x}, \boldsymbol{x'} \in X$ gilt für $\boldsymbol{x}$ die schwache Dominanz (Schreibweise: $\boldsymbol{x} \preceq \boldsymbol{x'}$), genau dann wenn $\forall i \in \{1, \ldots, k\} | x_i \leq x'_i$.*

Definition 4.8 (Stark dominierte Lösung)**.** : *Für zwei k-dimensionale Vektoren $\boldsymbol{x}, \boldsymbol{x'} \in X$ gilt für $\boldsymbol{x}$ die Dominanz (Schreibweise: $\boldsymbol{x} \prec \boldsymbol{x'}$), genau dann wenn $\forall i \in \{1, \ldots, k\} | \boldsymbol{x}_i \leq \boldsymbol{x'}_i \wedge \exists i \in \{1, \ldots, k\} | x_i < x'_i$. Es folgt $\boldsymbol{x} < \boldsymbol{x'}$.*

Definition 4.9 (Strikt dominierte Lösung)**.** : *Für zwei k-dimensionale Vektoren $\boldsymbol{x}, \boldsymbol{x'} \in X$ gilt für $\boldsymbol{x}$ die strikte Dominanz (Schreibweise: $\boldsymbol{x} \prec\prec \boldsymbol{x'}$), genau dann wenn $\forall i \in \{1, \ldots, k\} | x_i < x'_i$. Es folgt $\boldsymbol{x} << \boldsymbol{x'}$.*

Mit Hilfe der Definition einer (schwach) dominierten Lösung kann nachfolgende Beschreibung einer Pareto-optimalen Lösung angegeben werden.

Definition 4.10 (Pareto-optimale Lösung)**.** : *Der Vektor $\boldsymbol{x'} \in X$ stellt eine Pareto-optimale Lösung und zugleich nicht dominierte Lösung für das Problem **P** dar, genau dann wenn es kein anderes $\boldsymbol{x} \in X$ gibt für das $f(\boldsymbol{x}) \leq f(\boldsymbol{x'})$ gilt.*

Eine schwache Pareto-optimale Lösung kann hingegen wie folgt beschrieben werden.

Definition 4.11 (schwache Pareto-optimale Lösung)**.** : *Der Vektor $\boldsymbol{x'} \in X$ stellt eine schwache Pareto-optimale Lösung und zugleich schwach dominierte Lösung für das Problem **P** dar, genau dann wenn es kein anderes $\boldsymbol{x} \in X$ gibt für das $f(\boldsymbol{x}) < f(\boldsymbol{x'})$ gilt.*

Das Konzept der Pareto-Optimalität wurde erstmals 1896 von Vilfredo Pareto eingeführt. Aufgrund von wirtschaftlichen Untersuchungen entwickelte er die Aussage, dass sich, nur in einer nicht Pareto effizienten (optimalen) Gesellschaft, ein Individuum verbessern kann ohne ein anderes dabei schlechter zu stellen [Par96]. Dementsprechend ist eine Pareto-optimale Lösung dadurch gekennzeichnet, dass keine Verbesserung in einer Dimension erzielt werden kann, ohne dass gleichzeitig eine Verschlechterung in einer anderen Dimension herbeigeführt wird. Resultierend stellt eine Pareto-optimale Lösung immer auch gleichzeitig eine Kompromisslösung für das Problem **P** dar [Beu09]. Eine *nicht*/schwach dominierte Lösung grenzt sich von einer *(schwach)* Pareto-optimalen Lösung dahingehend ab, dass sie in einer Teilmenge $S \subseteq X$ vorliegen kann, es aber dominierende *(schwach)* Pareto-optimale Lösungen in der Obermenge X geben kann.

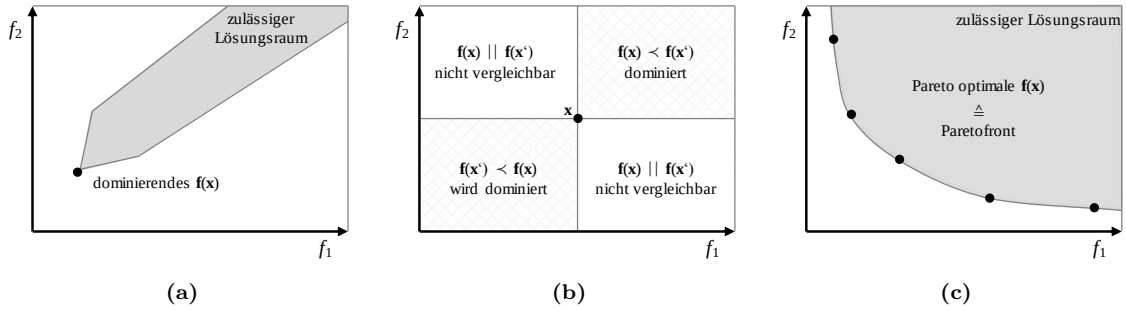


Abbildung 4.1.: Dominanz und Pareto-Optimalität. Unter (a) Eindeutig dominierendes $\mathbf{f}(\mathbf{x})$, (b) Dominanz und nicht Vergleichbarkeit von $\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{x}')$, (c) Pareto-Front der Pareto-optimalen $\mathbf{f}(\mathbf{x})$ (angelehnt an [LGCM10], Seite 72 und [Ste05], Seite 25).

Aufgrund der fehlenden Totalordnung in $\mathbb{R}^k$, ist für das mehrkriterielle Optimierungsproblem zu beachten, dass für beliebige $\mathbf{x} \neq \mathbf{x'} \in \mathbf{R}$ nicht immer $\mathbf{x} \preceq \mathbf{x'}$ oder $\mathbf{x'} \prec \mathbf{x}$ gilt. Insbesondere für konfliktäre Zielfunktionen f_k besteht die Möglichkeit, dass es keine eindeutig dominierende Lösung im Sinne der Dominanzrelation gibt. Dementsprechend muss die Unvergleichbarkeit von zwei Lösungen eingeführt werden [Ste05].

Definition 4.12 (Unvergleichbarkeit)**.** : *Die Vektoren $\boldsymbol{x}, \boldsymbol{x'} \in X$ sind unvergleichbar (Schreibweise: $\boldsymbol{x} || \boldsymbol{x'}$), genau dann wenn $\boldsymbol{x} \not\preceq \boldsymbol{x'} \wedge \boldsymbol{x'} \not\preceq \boldsymbol{x}$.*

Die Menge aller Pareto-optimaler Lösungen kann im Entscheidungsraum X wie folgt beschrieben werden.

Definition 4.13 (Paretomenge)**.** : *Die Paretomenge (PM) enthält alle nicht dominierten Lösungen im Entscheidungsraum X, das heißt nur solche Lösungen, welche sich schwach dominieren oder nicht vergleichbar sind.*

Hingegen bezieht sich die Paretofront (PF[5]) auf die zugehörigen Lösungen der Urbilder im Zielfunktionsraum $Y = f(X)$.

Definition 4.14 (Paretofront)**.** : *Die PF enthält alle nicht dominierten Lösungen im Zielfunktionsraum Y, das heißt nur solche Lösungen, welche sich schwach dominieren oder nicht vergleichbar sind.*

Ferner folgt, dass die Menge der nicht dominierten Lösungen nur solche enthält, die sich schwach dominieren oder nicht vergleichbar sind. In der Abbildung 4.1 werden die Zusammenhänge der Dominanzrelation für die Pareto Optimierung veranschaulicht.

Grundsätzlich werden im Rahmen der Ermittlung einer PF verschiedene Ziele verfolgt, um die Qualität und Güte der Menge nicht dominierter Lösungen zu gewährleisten. Zitzler et al. unterschieden in [ZDT00] drei Kriterien, die bei er Ermittlung einer PF beachtet werden sollten:

[5] Engl. Pareto Frontier

- Die Distanz zwischen der ermittelten (approximierten) PF und der wahren PF wird minimiert.
- Die Lösungen entlang der ermittelten (approximierten) PF sind gleichmäßig verteilt.
- Die Größe/Ausdehnung der ermittelten (approximierten) PF wird maximiert. Entsprechend weist die PF in allen Zieldimensionen eine hohe Diversität von Lösungen auf.

In der Regel muss der DM eine konkrete Lösung ermitteln. Der Vorteil der mehrkriteriellen Optimierung liegt in der Ermittlung einer Paretomenge. Entsprechend muss der DM die Prioritäten der Zielfunktionen a priori nicht zwangsläufig kennen. Ferner wird dem DM eine Menge an geeigneten Lösungen dargeboten, aus der er wählen kann [NZES05, Ste05]. Das Einbringen von Expertenwissen kann sich jedoch positiv auf die Suche im Lösungsraum auswirken. Hierbei werden verschiedene Ansätze unterschieden [Mie99, AB11]. Zum einen kann der DM seine Präferenzen vor der Lösungsermittlung einbringen. Somit kann der Lösungsraum weiter eingeschränkt werden. Des Weiteren besteht die Möglichkeit, dass die Präferenzen des DM während der Suche nach Lösungen kontinuierlich angepasst werden und interaktiv im Suchprozess genutzt werden.

Das MOSPP in dieser Dissertation wird als diskretes, mehrkriterielles Entscheidungsproblem betrachtet. Aus einer endlichen Menge von Alternativen $\mathcal{A} = \{a_1, \ldots, a_m\}$ soll die geeignetste Alternative a_i ausgewählt werden. Die Alternativen werden anhand von k Zielfunktionen $f_q : \mathcal{A} \to \mathbb{R}$ mit $q = 1, \ldots, k$ bewertet. Entsprechend kann jeder Alternative a_i ein konkreter Vektor mit entsprechenden Zielfunktionswerten, definiert über die Kosten $\mathbf{c}_{a_i} = f_q(a_i)$, zugeordnet werden. Ferner ist das MOSPP eine Erweiterung des SSPP [Shi55] und definiert sich wie folgt [Thu79, Lou83, HQF07]:

Für einen gerichteten Graphen $G = (V, E, C)$ werden die Kosten $\mathbf{c}$ einer jeden Kante $e_{i,j} \in E \subseteq V \times V$ mit Hilfe eines k-dimensionalen Vektor $\mathbf{c}(e_{i,j}) = (c_1(e_{i,j}), \ldots, c_k(e_{i,j}))^T$ mit $c : E \to [\mathbb{R}^+]$ beschrieben. Unter der Bedingung, dass eine additive Netzwerkmetrik vorliegt, definieren sich die Kosten für einen Pfad, ausgehend von einem Startknoten v_S zu einem Zielknoten v_G, zu $\mathbf{c}(p) = (\mathbf{c}(e_{S,i}), \ldots, \mathbf{c}(e_{j,G}))$. Jede Komponente q des Kostenvektors $\mathbf{c}$ setzt sich zu

$$c_q(p) = \sum_{(i,j) \in p} c_q(e_{i,j}), \text{ mit } q = 1, \ldots, k \tag{4.9}$$

zusammen. Für das MOSPP wird ein Pfad p gesucht für den die Kosten $\mathbf{c}(p)$ optimiert werden. Entsprechend dem allgemeinem MOOP existiert jedoch keine eindeutige Lösung und ferner auch kein eindeutiger Pfad, der über alle $c_q(p)$, $q = 1, \ldots, k$ optimiert. Resultierend wird eine Menge von nicht dominierten Pfaden gesucht. Das zugehörige lineare Programm kann wie folgt beschrieben werden [Mar84, KPY01, MMN11]:

$$\min \mathbf{c}(p) = \sum_{(i,j) \in E} c_q x_{i,j}, \ \forall q = 1, \ldots, k \tag{4.10}$$

Unter den Nebenbindungen:

$$\sum_j x_{i,j} - \sum_j x_{j,i} = \begin{cases} 1, & \text{wenn } i = v_S, \\ 0, & \text{wenn } i \neq v_S, v_G (i = 1, \ldots, n), \\ -1 & \text{wenn } i = v_G, \end{cases} \tag{4.11}$$

$$x_{i,j} \in \{0, 1\}, \ \forall (i,j) \in E \tag{4.12}$$

Die Menge X beschreibt den zulässigen Definitionsbereich der Entscheidungsvariablen auf den Kanten, eingeschränkt durch die Nebenbedingungen (4.11) und (4.12). Die Entscheidungsvariable $x_{i,j}$ entspricht eins, genau dann wenn die Kante als Bestandteil für einen kürzesten Pfad aufgenommen wird. Die Nebenbedingung (4.11) sichert, dass ein kürzester Pfad den Startknoten $v_S \in V$ mit dem Zielknoten $v_G \in V$ verbindet.

Ein naiver Ansatz zur Lösung des MOSPP kennzeichnet sich durch die Ermittlung aller zulässigen Lösungen und anschließenden Bestimmung der PF beziehungsweise PM. Für Probleme der Klasse $\mathcal{NP}$-vollständig steigt generell die Anzahl der zulässigen Lösungen exponentiell mit der Größe einer Probleminstanz. Mit Hilfe von sehr problemspezifischen, exakten Algorithmen wird die Laufzeit zur Lösung des Problems reduziert. Oftmals führen derartige Algorithmen nur bedingt zu einer schnelleren Problemlösung. Für die Klasse $\mathcal{NP}$-vollständigen Probleme weisen exakte Algorithmen eine exponentielle Laufzeit auf und sind entsprechend nur eingeschränkt anwendbar. Um eine geeignete, zulässige Lösung zu ermitteln, kann die Anwendung von approximativen Algorithmen zielführend sein. Hierbei werden gute Lösungen effizient ermitteln. Hingegen kann die optimale Lösung nicht mit Sicherheit bestimmt werden [DS04].

Für das MOOP verfolgen approximative Algorithmen beziehungsweise Heuristiken grundsätzlich das Ziel, eine möglichst gute Approximation der wahren PF zu ermitteln. Dementsprechend wird die Distanz der approximativen PF zur wahren PF minimiert. Ferner wird das Ziel verfolgt, möglichst viele Pareto Lösungen gleichmäßig über die PF zu ermitteln [Zit99]. In der Literatur werden für das MOOP verschiedene Heuristiken eingeführt: Simulated Annealing [CL94], Tabu Search [PMA05], Agent-Based Approach [CJ07], Particle Swarm Optimization [CPL04] und evolutionäre Algorithmen [Zit99]. In der Klasse der evolutionären Heuristiken stellt die ACO Metaheuristik insbesondere für das MOSPP einen geeigneten Ansatz dar. Hierbei wird die natürliche Suche von Ameisen nach einem kürzesten Pfad auf das MOSPP übertragen. Im Nachfolgenden wir die ACO Metaheuristik eingeführt, um zulässige Lösungen für das MOSPP zu ermitteln.

4.2.2. Die Ant Colony Optimization Metaheuristik – Eine Metaheuristik zur Lösung des Multiobjective Shortest Path Problem unter Unsicherheiten

Die ACO Metaheuristik ist ein stochastischer, populationsbasierter Ansatz für kombinatorische Optimierungsprobleme, die von Dorigo et al. [DS03b] mit dem Algorithmus Ant System eingeführt wurde. Die ACO Metaheuristik beschreibt allgemein ein Konzept und Rahmenwerk für die Anwendung bei unterschiedlichen Problemen. Hierbei basiert die ACO Metaheuristik auf der Grundlage von Verhaltensbeobachtungen realer Ameisenkolonien. Bei der Futtersuche hinterlassen Ameisen fortlaufend einen Botenstoff. Nachfolgende Ameisen werden von diesen Botenstoffen beziehungsweise Pheromonen angezogen. Aufgrund der additiven Eigenschaft der Pheromonmenge und höheren Frequentierung von kürzeren Pfaden erhöht sich die absolute Pheromonmenge gegenüber längeren Pfaden schneller. Resultierend folgen die Ameisen einem kürzeren Pfad mit einer höheren Wahrscheinlichkeit und verstärken selbst die Auswahlwahrscheinlichkeit zunehmend. Aufgrund der stochastischen, lokalen Entscheidungen folgen Ameisen vereinzelt auch scheinbar längeren Pfaden. Hierdurch können jedoch neue Pfade explorativ erschlossen werden, die sich anschließend global betrachtet als besser geeignet herausstellen. Die Exploration von neuen Pfaden wird durch eine natürliche Evaporation der Pheromone unterstützt, um die Möglichkeit einer Etablierung von neu-

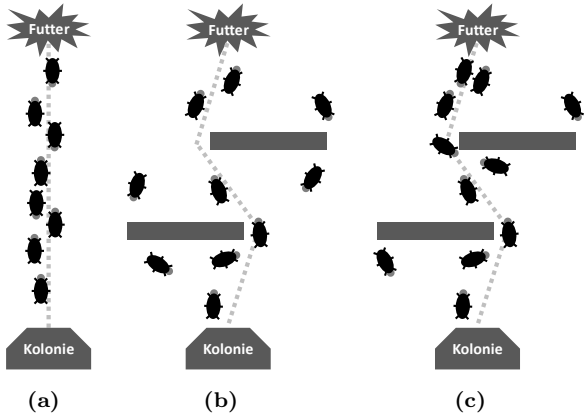


Abbildung 4.2.: Grundidee der ACO Metaheuristik. Unter (a) Direkter, kürzester Pfad zwischen zwei Punkten, (b) Zu Beginn der Suche nach dem kürzesten Pfad, (c) Bestimmung des kürzesten Pfades im weiteren Verlauf der Suche basiert auf einem selbstverstärkenden Prozess. Der kürzere Pfad ermöglicht eine höhere Frequentierung, welche zu einer höheren Pheromonkonzentrationen führt und mehr Ameisen anzieht (angelehnt an [DCG99], Seite 139).

en Pfaden gewährleisten zu können. Zusammenfassend fungieren die Pheromone als gemeinsames „Gedächtnis“ der Ameisenkolonie, um Informationen über den gesamten Verlauf des Suchprozesses, einschließlich der Systemumwelt, zu kodieren.

Die Nutzung von Pheromonen basiert grundlegend auf dem Konzept der Stigmergie. Informationen werden lokal zur Verfügung gestellt und in die Entscheidungsprozesse einbezogen. Entsprechend liegt eine indirekte Kommunikation vor. Das emergente Systemverhalten der gesamten Kolonie beruht auf der dezentralen Koordination individueller Aktivitäten der Ameisen. Die Entscheidungen werden autonom auf Grundlage der Pheromone getroffen. Hinzukommend werden zusätzliche Informationen über die lokale Umgebung einbezogen. Zusammenfassend kann mittels der Stigmergie der Informationsaustausch und die Entscheidungsfindung auch in großen Gruppen effizient gestalten werden [Ell07]. In Anlehnung an Beobachtungen aus der Natur, ist in der Abbildung 4.2 ein Beispiel für die Suche nach dem kürzesten Pfad veranschaulicht.

Die Beobachtungen und Eigenschaften einer Ameisenkolonie werden für die ACO Metaheuristik genutzt, um für Probleme geeignete Lösungen zu ermitteln. Die erste Anwendung wurde von Dorigo [Dor92, DG97] anhand des TSP gezeigt. Darauffolgend wurden weitere Implementierungen erfolgreich, beispielsweise für Instanzen des Vehicle Routing Problems, des Scheduling Problems, des Knapsack Problems oder des Graph Colouring [SLID11]. Nachfolgend werden das Grundmodell der ACO Metaheuristik eingeführt und die Erweiterungen hinsichtlich der MOO erläutert.

4.2.2.1. Das Grundmodell

Im Folgenden wird das Grundmodell der ACO Metaheuristik beschrieben und auf die verschiedenen Designkomponenten eingegangen. Die Erläuterungen orientieren sich an den Ausführungen in [DS04, AB11, LIS12]. Für jede Designkomponente werden die frei definierbaren Elemente hervorgehoben und mögliche Implementierungen aufgezeigt.

Die Ausprägung der Elemente nimmt einen unmittelbaren Einfluss auf die Qualität der gefundenen Lösungen und muss problemspezifisch angepasst werden.

Das ACO Grundmodell wird anhand des SPP erläutert. Hierbei wird das SPP auf den zugehörigen Graphen $G = (V, E, C)$, mit einer endlichen Menge von Knoten $V = \{1, \ldots, n\}$ und einer endlichen Menge von Kanten mit $E \subseteq \{(i, j) | i \neq j; i, j \in V\}$ abgebildet. Auf den Kanten werden die Kosten (beispielsweise Entfernung) zwischen zwei benachbarten Knoten definiert. Die Ameisen bewegen sich auf den Knoten und Kanten jeweils von einem Startknoten v_S zu einem Zielknoten v_G.

In Anlehnung an das TCP und die interne Abbildung der Systemumwelt Ξ', kann die Menge der Knoten V mit den Zellen assoziiert werden. Aufbauend auf E beschreibt die Menge C die Kosten, die von den Agenten aufgebracht werden müssen, um sich zwischen zwei Zellen zu bewegen.

- **Bestimme Lösungen für Ameisen:**
  Mit dieser Designkomponente wird wesentlich das Verhalten der individuellen Ameisen $a_k \in A$ und ferner das emergente Gesamtverhalten der Ameisenkolonie bestimmt. Hierbei bewegen sich die Ameisen auf dem zum Problem zugrundeliegenden Graphen und ermitteln inkrementell eine zulässige Lösung. Neben der einzusetzenden Anzahl von Ameisen $|A|$ muss der Startknoten der Ameisen definiert werden. Es ist möglich, alle Ameisen ausgehend von einem gemeinsamen Startknoten v_S mit der Suche zum Zielknoten v_G beginnen zu lassen. Eine weitere Möglichkeit besteht darin, alle Knoten $v_i \in V$ mit Ameisen zu belegen. Ausgehend von dem jeweiligen Startknoten $v_{S_{a_k}}$ werden Lösungen zu dem Zielknoten v_G ermittelt. Zum Problem relevante Lösungen tragen nur die Ameisen bei, die auf dem tatsächlichen Startknoten initialisiert werden. Jedoch wird die Suche über den gesamten Lösungsraum verteilt.

  Entscheidungen zur Aufnahme von Knoten/Lösungskomponenten werden von den Ameisen auf der Grundlage der Informationen aus der Pheromonmatrix $\mathcal{P}$ und Heuristikmatrix $\mathcal{H}$ getroffen. Beide Matrizen sind auf den Kanten $e_{i,j}$ zwischen zwei benachbarten Knoten v_i, v_j definiert.

  - *Pheromoninformation und heuristische Informationen:*
    Die Pheromonmatrix $\mathcal{P}$ kann mit unterschiedlichen Startwerten initialisiert werden und verschiedene Eigenschaften aufweisen. Grundsätzlich trägt die Pheromonmatrix $\mathcal{P}$ dazu bei, die Suche effizienter und zielgerichteter zu gestalten. Hierbei wird die Suche vor allem auf Regionen im Suchraum gerichtet, die gute Lösungen vermuten lassen. Kanten die geringe Kosten aufweisen, werden für das SSP tendenziell einen höheren Pheromonwert aufweisen. Die Pheromonmatrix $\mathcal{P}$ kann mit einem gleichen Startwert für alle Einträge $\rho_{i,j}$ vordefiniert werden. Des Weiteren ist es möglich, eine obere und untere Schranke für jeden Eintrag in der Pheromonmatrix festzulegen. Mit dem Intervall $[\rho_{\min}, \rho_{\max}]$ kann einer Stagnation in der Suche nach optimierten Lösungen, beziehungsweise einer schnellen Konvergenz gegen eine Lösung reduziert werden. Maßgeblich ist hierfür eine begrenzte Dominanz gegenüber Knoten mit $\rho_{\min}$ [SH00]. Der Startwert kann mit $\rho_{i,j} \geq 0$ oder auch gemäß $\rho_{\min}, \rho_{\max}$ gewählt werden. Zudem ist es auch denkbar, dass Einträge unterschiedliche Startwerte aufweisen, um die Suche zuerst auf eine bestimmte Region im Suchraum zu orientieren.

Die Heuristikmatrix $\mathcal{H}$ kodiert a priori bekannte Beobachtungen, Eigenschaften oder Expertenwissen über das Problem. Für das SSP entsprechen die Einträge $\eta_{i,j}$ der Heuristikmatrix $\mathcal{H}$ den Kosten oder genauer der Länge der Kanten $e_{i,j}$ zwischen zwei Knoten v_i, v_j.

– *Pheromon- und Heuristikaggregation:*
Jede Ameise a_k ermittelt inkrementell eine zulässige Lösung und für das SPP entsprechend einen Pfad zwischen $v_{S_{a_k}}$ und v_G. Zu jedem Zeitschritt t wird die Güte einer möglichen Lösungskomponente (nächster Knoten) v_j mittels einer Funktion abgebildet. Die Gütefunktion $f : e_{i,j} \in E \rightarrow \mathbb{R}$ ist auf den Kanten definiert, die zwei Knoten v_i, v_j verbinden. Hierzu werden die zugehörigen numerischen Pheromoninformationen $\rho_{i,j}$ und heuristischen Informationen $\eta_{i,j}$ herangezogen. Für eine Ameise a bestimmt sich die Auswahlwahrscheinlichkeit p einer konkreten Lösungskomponente v_j aus einer Nachbarschaft $\Phi_{a_k}(v_i, t)$ für den aktuellen Knoten v_i über alle Kanten $e_{i,u} \in \Phi_{a_k}$ zum Zeitschritt t zu

$$p_a(v_j) = \begin{cases} \frac{[\rho_{i,j}]^\alpha [\eta_{i,j})]^\beta}{\sum_{e_{i,u} \in \Phi_{a_k}(v_i,t)} [\rho_{i,u}]^\alpha [\eta_{i,u}]^\beta}, & \text{wenn } v_j \in \Phi_{a_k}(v_i, t) \\ 0, & \text{sonst.} \end{cases} \tag{4.13}$$

Mit Hilfe von α und β wird eine relative Gewichtung der Pheromone und heuristischen Informationen und deren Einfluss auf die Auswahlwahrscheinlichkeit vorgenommen. Entsprechend betrachtet die Auswahlwahrscheinlichkeit für $\beta = 0$ ausschließlich die Informationen aus der Pheromonmatrix. Hingegen verhält sich die Auswahlwahrscheinlichkeit für $\alpha = 0$ proportional zu den heuristischen Informationen. Resultierend werden für das SPP eher Knoten ausgewählt, die geringere Kosten/Längen aufweisen. Eine geeignete, systematische Variation von α und β unterstützt die Exploration und Exploitation des gesamten Lösungsraumes. Ferner wird einer schnellen Konvergenz entgegengewirkt [DMC96].

Die Aktualisierung der Pheromonmatrix $\mathcal{P}$ kann entweder bereits während der Lösungsermittlung oder zentral vor Beendigung einer Generation durchgeführt werden. In einer Generation versuchen alle Ameisen zulässige Lösungen (Pfade) zu bestimmen, die über weitere Generationen unter Verwendung der Pheromoninformationen verbessert werden. Die Pheromoninformationen werden über eine Pheromonaktualisierung und zusätzliche Dämonenaktionen angepasst.

- **Pheromonaktualisierung:**
Die Pheromonaktualisierung wird für die gesamte Pheromonmatrix einer Generation durchgeführt. Es besteht die Möglichkeit, dass Ameisen, während oder nach der individuellen Lösungsermittlung, auf jede besuchte Kante eine bestimmte Anzahl von Pheromonen hinzufügen. Entsprechend werden bei einer hohen Frequentierung von Lösungskomponenten (Knoten) die nachfolgenden Ameisen angezogen. Des Weiteren können alle Einträge einer Pheromonmatrix mittels der Pheromonevaporation verringert werden. Zudem kann die Pheromondiffusion implementiert werden. Hierbei findet eine gezielte Hinzugabe/Umverteilung von Pheromonen statt. In der Regel werden die Pheromonevaporation und -diffusion für alle Einträge der Pheromonmatrix $\mathcal{P}$ vorgenommen.

– *Pheromonhinzugabe:*
Die Auswahlwahrscheinlichkeiten besuchter Kanten und Knoten, die Bestandteil einer zulässigen Lösung des SPP $s_{a_k} \in \mathcal{S}$ sind, werden durch die Ameisen a_k mit einer Hinzugabe von Pheromonen erhöht. Die Einträge $\rho_{i,j}$ der Pheromonmatrix $\mathcal{P}$ definieren sich zu

$$\rho_{i,j} \leftarrow \rho_{i,j} + \sum_{a_k=1}^{|A|} \Delta\rho_{i,j}^{a_k}, \qquad \forall e_{i,j} \in \mathcal{S}. \tag{4.14}$$

Hierbei beschreibt $\Delta\rho_{i,j}^{a_k}$ die Anzahl aller Pheromonhinzugaben der Ameisen $a_k \in A$. Es existieren unterschiedliche Definition bezüglich der Pheromonhinzugabe. Eine übliche Definition erfolgt über die Güte einer Lösung s_{a_k}. Für das SPP kann die Güte allgemein mit Hilfe der Kosten $c : s_{a_k} \in \mathcal{S} \rightarrow \mathbb{R}$ einer Route s_{a_k} angegeben werden. Die Kosten $c(s_{a_k})$ entsprechen der Summe über alle befahrenen Kanten. Es folgt:

$$\Delta\rho_{i,j}^{a_k} = \begin{cases} \frac{1}{c(s_{a_k})}, & \text{wenn } e_{i,j} \in s_{a_k} \\ 0, & \text{sonst} \end{cases} \tag{4.15}$$

Entsprechend werden die Pheromone proportional zur Güte einer Lösung auf die zugehörigen Lösungskomponenten hinzugegeben.

– *Pheromonevaporation:*
Mit Hilfe der Evaporation wird eine zu schnelle Konvergenz gegen eine Lösung verhindert. Resultierend können lokal gute Lösungen besser überwunden werden, um eine global bessere Lösung zu finden. Ferner wird die Exploration im gesamten Lösungsraum unterstützt. Formal wird für die Pheromonevaporation eine Evaporationsrate ϱ_E definiert, die alle Pheromoneinträge reduziert. Resultierend bestimmen sich die Pheromoneinträge $\rho_{i,j} \in \mathcal{P}$ zu

$$\rho_{i,j} \leftarrow (1 - \varrho_E)\rho_{i,j}, \forall (i,j) \in \mathcal{P}. \tag{4.16}$$

mit $0 < \varrho_E \leq 1$. Entsprechend nehmen die Pheromonwerte exponentiell ab.

– *Pheromondiffusion:*
Mit der Pheromondiffusion kann die lokale Suche in den Nachbarschaften guter Lösung unterstützt werden. Gute Lösungskomponenten mit einer hohen Pheromonanzahl erhöhen die Auswahlwahrscheinlichkeit und somit die Anziehungskraft von benachbarten Lösungskomponenten. Ein definierter Anteil der Pheromone auf den Kanten $e_{i,j}$ diffundiert zusätzlich auf alle Kanten $e_{u,q}$ innerhalb der Nachbarschaft $\Phi(e_{i,j})$. Die Diffusionsrate ϱ_D gibt den prozentualen Anteil der diffundierenden Pheromone an. Der Pheromonwert für einen Eintrag $\rho_{u,q}$ beträgt nach der Pheromondiffusion

$$\rho_{u,q} \leftarrow \rho_{u,q} + f(e_{i,j}, e_{u,q})\varrho_D\rho_{i,j}, \forall (i,j) \in \mathcal{P} | e_{u,q} \in \Phi(e_{i,j}). \tag{4.17}$$

In der Regel wird eine exponentielle Funktion $f : e_{i,j}, e_{u,q} \in E \rightarrow \mathbb{R}$ gewählt, die mit zunehmender Entfernung zwischen zwei Kanten $e_{i,j}, e_{u,q}$ abnimmt. Resultierend diffundieren weniger Pheromone. Die Entfernung von zwei Kanten kann beispielsweise problemspezifisch über die Heuristikmatrix $\mathcal{H}$ oder über die Länge der benötigten Folge von Kanten zwischen $e_{i,j}$ und $e_{u,q}$ definiert werden.

- **Dämonaktionen:**
  Mit Hilfe von Dämonaktionen können zentrale, übergeordnete Aktionen implementiert werden, die einen positiven Einfluss auf die Suche der Ameisen im Lösungsraum nehmen. Beispielsweise kann die lokale Suche dazu genutzt werden, um in den Nachbarschaften von Pareto-optimalen Lösungen weiter zu suchen. Die Auswertung von globalen Informationen kann dazu genutzt werden, um zusätzliche Pheromone auf einzelne Lösungskomponenten hinzufügen zu können.

  Nachdem alle Ameisen eine Lösung ermittelt haben, können zusätzliche Pheromone auf einzelnen Einträgen der Pheromonmatrix hinzugefügt werden. Auf der Grundlage von verschiedenen Definitionen werden die Ameisen bestimmt, die in einer Generation zusätzliche Pheromone auf die zugehörigen Lösungen hinzugeben. Beispielsweise kann nur der besten Lösung über alle Generationen oder der jeweiligen Generation zusätzliche Pheromone hinzugefügt werden. Es ist auch möglich, im Verhältnis zur besten Lösung über alle Generationen die Menge der zusätzlichen Pheromone zu bestimmen.

  Grundsätzlich führt die zusätzliche Hinzugabe von Pheromonen zu einer erhöhten Auswahlwahrscheinlichkeit der Lösungskomponenten und ferner zu einer stärkeren Anziehung der Ameisen in den nachfolgenden Generationen.

- **Abbruchkriterium:**
  Grundlegend sind zwei Abbruchkriterien zu unterscheiden. Zum einen ist die Definition eines Abbruchkriteriums für die individuelle Suche der Ameisen nach einer Lösung erforderlich. Hierbei können Ameisen für einen zyklischen oder ungerichteten Graphen in Zyklen geraten. Zudem kann eine ungünstige Wahl des Startknotens dazu führen, dass keine zulässige Lösung ermittelt werden kann. Demzufolge ist eine garantierte Terminierung der Ameisen innerhalb der Generationen notwendig.

  Zum anderen ermitteln die Ameisen solange Lösungen in aufeinanderfolgenden Generationen, bis ein definiertes Abbruchkriterium erreicht wird. Es gibt mehrere Möglichkeiten einer Definition. Ein intuitiver Ansatz setzt eine maximale Anzahl von Generationen. Mit dem Erreichen der maximalen Generation wird die Suche abgebrochen. Eine weitere Möglichkeit besteht darin, die Suche nach neuen Lösungen zu beenden, wenn eine Konvergenz der Lösungsgüte eintritt, beziehungsweise für eine vordefinierte Anzahl von Generationen keine bessere Lösung ermittelt wird. Ebenfalls kann der DM eine zu erreichende Lösungsgüte vorgeben, für die ein Abbruch der Suche folgt.

Die Metaheuristik gibt keine Spezifikationen vor, wie die Gestaltung der Designkomponenten vorgenommen werden soll. Des Weiteren werden keine Vorgaben bezüglich der Ablaufplanung vorgenommen. Die Designkomponenten können parallelisiert und synchronisiert werden oder unabhängig voneinander durchgeführt werden. Die Interaktion der Designkomponenten muss den Zielen der Optimierung und den Problemeigenschaften angepasst werden. In dem Algorithmus 4.1 ist der ACO Grundalgorithmus angelehnt an [DS04] zusammengefasst.

Die Implementierung der ACO Metaheuristik für das MOSPP bedarf Erweiterungen und Anpassungen des Grundalgorithmus. Im nachfolgenden Abschnitt werden die Besonderheiten und Möglichkeiten sowie die notwendigen Anpassungen an die Designkomponenten erläutert.

Algorithmus 4.1 : Ant Colony Optimization – Grundalgorithmus

Prozedur *ACO(#Ameisen, Abbruchkriterium, Pheromonabgabe, Pheromonevaporationsrate)*:

Eingabe : *#Ameisen, Abbruchkriterium, Pheromonabgabe, Pheromonevaporationsrate*
Ausgabe : Lösung

Initialisiere: Ameisen, Pheromonmatrix, Heuristikmatrix
repeat
  Ablaufplanung
    Bestimme Lösungen für Ameisen
    Aktualisiere Pheromonmatrix
    Führe Dämonaktionen aus {optional}
  end Ablaufplanung
until *Abbruchkriterium*;
return Lösung s

4.2.2.2. Erweiterungen für das Multiobjective Optimization Problem

Auf der Grundlage von erfolgreichen Implementierungen der ACO Metaheuristik für das Singleobjective Shortest Path Problem (SOSPP), sind verschiedene Erweiterungen und Anpassungen bezüglich des MOOP entwickelt worden. Im nachfolgenden werden die notwendigen Anpassungen an die mehrkriteriellen ACO Heuristiken (MACO[6]) eingeführt. Insbesondere eine geeignete Kolonie- und Pheromonstruktur sowie zusätzlich eine Verwaltung Pareto-optimaler Lösungen müssen eingeführt werden. Die nachfolgenden Ausführungen richten sich nach [DS04, LIS12] falls nicht anders angegeben.

- **Kolonie:**
  Es besteht die Möglichkeit, ebenfalls für das MOOP eine Ameisenkolonie zu verwenden. Hierbei werden heterogene Ameisen verwendet, die mittels unterschiedlicher Gewichtungen der Pheromon-, und Heuristikmatrizen die Suche in verschiedenen Regionen des mehrdimensionalen Lösungsraumes konzentrieren.

  Des Weiteren können mehrere, homogene und heterogene Ameisenkolonien genutzt werden. Jede Ameisenkolonie konzentriert die Suche nach Pareto-optimalen Lösungen auf verschiedene Regionen des Suchraumes, entsprechend der zugehörigen Zielfunktion. In der Regel entspricht die Anzahl der Zielfunktionen der Anzahl der Ameisenkolonien. Hierzu werden für jede Ameisenkolonie individuelle Pheromon- und Heuristikmatrizen definiert. Um die Suche zu verbessern, können die Ameisenkolonien miteinander kooperieren. Grundsätzlich können Informationen bezüglich Pareto-optimaler Lösungen ausgetauscht werden, um dominierte Lösungen frühzeitig zu identifizieren. Zudem können Pareto-optimale Lösungen zur Pheromonaktualisierung aller Ameisenkolonien genutzt werden.

  In [ASG07] wird eine zusätzliche Ameisenkolonie eingeführt, um geeignete Pareto-optimale Kompromisslösungen zu ermitteln, die zwischen den Extremlösungen der betrachteten Zielfunktionen liegen.

[6] Eng. Multiple Ant Colony Optimization

- **Bestimme Lösungen für Ameisen:**
  In Abhängigkeit der realisierten Ameisenkolonien werden problemspezifische Definitionen der Pheromon- und Heuristikmatrix vorgenommen. Es können Ansätze mit einer gemeinsamen oder mehreren Pheromon- und Heuristikmatrizen unterschieden werden. Insofern mehrere Matrizen, entsprechend der Anzahl der Zielfunktionen, verwendet werden, kann entweder eine einkriterielle Betrachtung der zugehörigen Kolonie vorgenommen werden oder eine Aggregation der Pheromoninformationen und heuristischen Informationen. Bei einer Aggregation können die Ansätze der gewichteten Summe und des gewichteten Produkts unterschieden werden. Ebenso ist eine zufallsbasierte Auswahl und Betrachtung der Pheromon- und Heuristikmatrix möglich.

  - *Pheromoninformation und heuristische Informationen:*
    Es kann zwei verschiedene Implementierungen der Pheromoninformationen und heuristischen Informationen unterschieden werden. Bei der Verwendung von einer jeweils gemeinsamen Pheromon- und Heuristikmatrix folgt der Ansatz den einkriteriellen ACO Heuristiken. Mittels des Modells der gewichteten Summe werden die unterschiedlichen Informationen über die Zielfunktionen zu $f_{\text{gesamt}}(x) = \alpha f(x) + \ldots + \alpha' f'(x)$ aggregiert und entsprechende heuristische Informationen abgebildet. Nachteilig wirkt sich hierbei die eingeschränkte Flexibilität der Aggregation aus. Zudem muss Expertenwissen vorhanden sein, um eine a priori Gewichtung der Zielfunktionen vornehmen zu können. Ähnlich spiegelt die Pheromonmatrix die gesammelten Informationen über den gesamten Lösungsraum aggregiert wider.

    Des Weiteren existieren Ansätze, die in der Regel für jede Zielfunktion jeweils eine Pheromon- und Heuristikmatrix verwenden. Mittels einer geeigneten Strategie bezüglich der Pheromonaktualisierung werden Informationen über das Problem und den Lösungskomponenten zwischen den Matrizen ausgetauscht. Eine a priori Gewichtung der Zielfunktionen ist nicht notwendig.

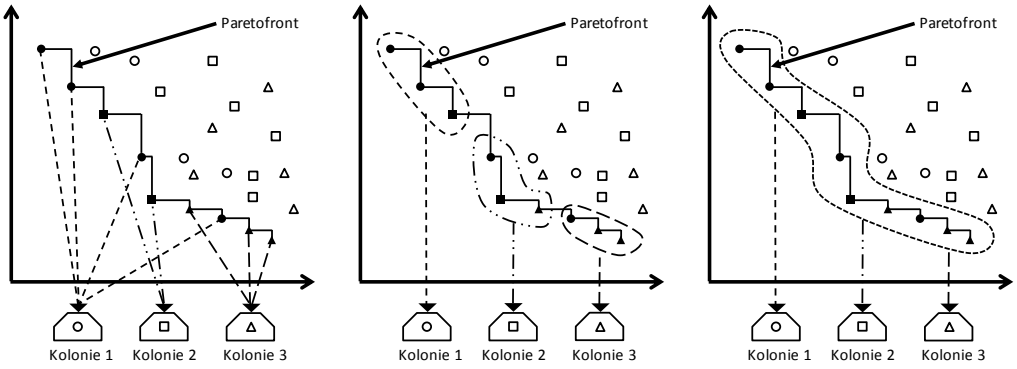
  - *Pheromon- und Heuristikaggregation:*
    Werden heterogene Ameisen in einer Kolonie implementiert, dann führt die Verwendung von Gewichtungsfaktoren bei jeder Ameise zu einer unterschiedlichen Gewichtung der Zielfunktionen. In den Extremfällen werden von Ameisen ausschließlich eine Zielfunktion betrachtet. Wiederrum versucht ein Großteil der Ameisen geeignete Pareto-optimale Kompromisslösungen entsprechend der jeweiligen Gewichtung der Zielfunktionen zu finden. Um die Auswahlwahrscheinlichkeit einer Lösungskomponente bestimmen zu können, werden die Pheromoninformationen und heuristischen Informationen geeignet aggregiert. Es werden hinsichtlich einer Aggregation zwei Ansätze unterschieden. Eine Aggregation kann über ein gewichtetes Produkt $\rho_{i,j} = \prod_{q=1}^{k} \rho_{i,j}^{q} \lambda^q$ und $\eta_{i,j} = \prod_{q=1}^{k} \eta_{i,j}^{q} \lambda^q$ vorgenommen werden [SH00]. Analog kann ebenfalls eine Aggregation mittels einer gewichteten Summe durchgeführt werden [DHR03, ASG07]. Entsprechend setzen sich die betrachteten m Pheromon- und Heuristikmatrizen zu $\rho_{i,j} = \sum_{q=1}^{k} \rho_{i,j}^{q} \lambda^q$ und $\eta_{i,j} = \sum_{q=1}^{k} \eta_{i,j}^{q} \lambda^q$ zusammen. Für beide Ansätze dient eine mögliche Gewichtung λ der Pheromoninformationen und heuristischen Informationen dazu, eine Suche nach Lösungen gezielt auf unterschiedliche Regionen des Lösungsraumes zu konzentrieren.

Ansätze, die für jede Zielfunktion eine Ameisenkolonie mit zugehöriger Pheromon- und Heuristikmatrix definieren, ermitteln die Auswahlwahrscheinlichkeit für die nächste Lösungskomponente entsprechend den einkriteriellen ACO Heuristiken. Wird eine zusätzliche Ameisenkolonie wie in [ASG07] eingeführt, dann kann zufallsbasiert eine der Pheromon- und Heuristikmatrizen betrachtet werden.

- **Pheromonaktualisierung:**
  Grundsätzlich können die Modellierungen der Pheromonaktualisierung der ACO Heuristiken auf MACO Heuristiken übertragen werden. Entsprechend kann die geeignetste Lösung der aktuellen Iteration oder die geeignetste Lösung über alle Iterationen zur Bestimmung der zusätzlichen Hinzugabe von Pheromonen herangezogen. Werden mehrere Pheromonmatrizen in Abhängigkeit der Zielfunktionen definiert, können für jede Pheromonmatrix jeweils die Pareto-optimalen Lösungen entsprechend der zugehörigen Zielfunktion betrachtet werden, die zudem Bestandteil der globalen PF sind. In [IMM01] wird die Annahme aufgestellt, dass diese Alternative zu bevorzugen ist, weil die Kolonien gegenseitig dominierte Lösungen in den lokalen PFs identifizieren. Zudem wird beobachtet, dass eine Suche in unterschiedlichen Regionen des Lösungsraumes unterstützt wird. Begründet wird diese Beobachtung dadurch, dass die Wahrscheinlichkeit eine Pareto-optimale Lösung zu finden in Regionen des Lösungsraumes, die zum Zeitpunkt der Suche kaum Lösungen aufweisen, größer ist. Des Weiteren kann die Aktualisierung der Pheromonmatrizen auf der Grundlage der Pareto-optimalen Lösungen erfolgen. Hierbei können die Pareto-optimalen Lösungen zu einer Aktualisierung der Ameisen zugehörigen Pheromonmatrix, einer spezifischen Auswahl oder allen Pheromonmatrizen führen. Entsprechend kann die Suche im Lösungsraum gezielt durchgeführt werden. Eine Aktualisierung aller Pheromonmatrizen entspricht dem MACO Ansatz mit einer Kolonie. Grundsätzlich führen problemspezifische Anpassungen zu einer Verbesserung der Lösungsgüte. Hierbei können beispielsweise die Anzahl der ermittelten Lösungen einer Generation und die Güte der individuellen Lösungen sowie weitere Metriken genutzt werden, um die Pheromonaktualisierung geeignet und problemspezifisch durchzuführen. Die unterschiedlichen Designmöglichkeiten der Pheromonaktualisierung sind in Abbildung 4.3 veranschaulicht.

- **Paretofront:**
  Wird die PF nicht zur Aktualisierung der Pheromonmatrizen herangezogen, ist eine Speicherung der Lösungen einer jeden Generation hinreichend. Nachdem alle Generationen durchgeführt sind, wird die zugehörige Paretomenge bestimmt und als Lösung des Problems ausgegeben.

  Insofern die PF genutzt wird, um eine Aktualisierung der Pheromonmatrizen durchzuführen, muss die zugehörige Paretomenge kontinuierlich über die Generationen bestimmt werden. Grundsätzlich können zwei verschiedene Implementierungen der Paretomenge umgesetzt werden. Zum einen kann eine offline Verwaltung der Paretomenge vorgenommen werden. Hierbei wird die Paretomenge extern gespeichert. Für jede Generation werden die neuen Lösungen anhand der bisherigen PF bewertet und die zugehörige Paretomenge neu bestimmt. Nach der letzten Generation wird die Paretomenge als Lösung des Problems ausgegeben. Zum anderen kann die Verwaltung der Paretomenge online durchgeführt werden. Die Pareto-optimalen Lösungen nehmen einen direkten Einfluss auf die Pheromonaktualisierung. Eine Verbesserung der aktuellen PF führt unmittelbar zu einer Aktualisierung der Pheromonmatrizen.



(a) **(b)** **(c)**

Abbildung 4.3.: Designmöglichkeiten der Pheromonaktualisierung für MACO Heuristiken, am Beispiel eines bikriteriellen Minimierungsproblems. Unter (a) Ameisen aktualisieren ausschließlich die eigene Pheromonmatrix, wenn eine Pareto-optimale Lösung gefunden wurde, (b) Ameisen aktualisieren die Pheromonmatrix, welche der Pareto-optimalen Lösung, aufgrund der Lage im Lösungsraum, zugeordnet werden kann, unabhängig der Zugehörigkeit zu den Kolonien, (c) Ameisen aktualisieren alle Pheromonmatrizen, wenn eine Pareto-optimale Lösungen gefunden wurde, unabhängig der Zugehörigkeit zu den Kolonien (angelehnt und erweitert nach [IMM01], Seite 365).

Entsprechend spiegeln die Pheromonmatrizen zu jeder Zeit die Ausprägung der aktuellen Paretomenge wider.

In dem Algorithmus 4.2 sind die notwendigen Erweiterungen für das MOOP zusammengefasst.

Algorithmus 4.2 : Multiple Ant Colony Optimization – Grundalgorithmus

Prozedur *MACO(#Kolonien, #Ameisen, #Pheromonmatrizen, #Heuristikmatrizen, Abbruchkriterium, Pheromonabgabe, Pheromonevaporationsrate)***:**

Eingabe : *#Kolonien, #Ameisen, #Pheromonmatrizen, #Heuristikmatrizen, Abbruchkriterium, Pheromonabgabe, Pheromonevaporationsrate*
Ausgabe : Paretomenge

Initialisiere: Kolonien, Ameisen, Pheromonmatrizen, Heuristikmatrizen, Paretomenge
repeat
  Ablaufplanung
    Bestimme Lösungen für Ameisen
    Bestimme Paretofront und zugehörige Paretomenge
    Aktualisiere alle Pheromonmatrizen
    Führe Dämonaktionen aus {optional}
  end Ablaufplanung
until *Abbruchkriterium*;
return Paretomenge

4.2.2.3. Die BiCriterionAnt Heuristik

Im Nachfolgenden wird die BiCriterionAnt Heuristik vorgestellt, welche zur Lösungsermittlung des MOSPP in dieser Dissertation genutzt wird. Untersuchungen von López-Ibáñez und Stützle in [LIS12] haben für ein bikriterielles Problem gezeigt, dass die BiCriterianAnt Heuristik eine der geeignetsten Heuristiken der ACO Metaheuristik ist.

Die nachfolgenden Ausführungen basieren auf [IMM01], in der Iredi, Merkle und Middendorf die BiCriterianAnt Heuristik erstmals einführen. Es werden die Implementierungen bezüglich des Ansatzes einer Ameisenkolonie erläutert. Hierbei werden zwei Pheromonmatrizen genutzt, um Informationen über die Zielfunktionen für das bikriterielle Optimierungsproblem zu repräsentieren.

- **Kolonie:**
  Grundsätzlich wird die Suche nach geeigneten Lösungen verbessert, wenn der Lösungsraum in verschiedenen Regionen parallel untersucht wird. Aus diesem Grund wird für den Ansatz unter Verwendung einer einzigen Kolonie vorgeschlagen, dass sich diese aus heterogenen Ameisen zusammensetzt. Entsprechend wird die Suche nach Lösungen im zweidimensionalen Suchraum gleichmäßig unter den Ameisen a_k mit $\lambda_k = \frac{k-1}{|A|-1}$ aufgeteilt. Hinsichtlich der heterogenen Kolonie wird λ_k als Gewichtungsfaktor für die Pheromon- und Heuristikaggregation eingeführt, um eine unterschiedlich gewichtete Auswahl der Lösungskomponenten vornehmen zu können.

- **Bestimme Lösungen für Ameisen:**
  In jeder Generation starten alle Ameisen zur Ermittlung einer Lösung von dem gemeinsamen Startknoten v_S. In jedem Zeitschritt wird eine Pheromon- und Heuristikaggregation für alle Kanten in der Nachbarschaft durchgeführt. Die stochastische Auswahl eines der direkten Nachbarknoten wird für jede Ameise solange wiederholt, bis die Ameise einen Pfad zwischen v_S und v_G ermittelt hat.

  - *Pheromoninformation und heuristische Informationen:*
    Es werden jeweils zwei Pheromon- und Heuristikmatrizen definiert, um zum einen Informationen bezüglich der zwei Zielfunktionen zu speichern und zum anderen die Suche im Lösungsraum geeignet steuern zu können. Im weiteren Verlauf dieser Dissertation werden die Pheromonmatrizen im Sinne der $\mathcal{MAX} - \mathcal{MIN}$ Heuristik [SH00] umgesetzt. Entsprechend wird ein Intervall $[\rho_{\min}, \rho_{\max}]$ angegeben, für das die Einträge der Pheromonmatrizen definiert sind.

  - *Pheromon- und Heuristikaggregation:*
    Der Parameter λ_k führt in den Extremfällen mit λ_k gleich null und eins zu einer Gewichtung und Betrachtung jeweils einer Zielfunktion. Pareto-optimale Kompromisslösungen werden tendenziell von Ameisen a_k mit $0 < \lambda_k < 1$ ermittelt. Die Auswahlwahrscheinlichkeit für ein bikriterielles Optimierungsproblem, mit jeweils zwei Pheromon- und Heuristikmatrizen, kann für eine Ameise a_k wie folgt

$$p_{a_k}(v_j) = \begin{cases} \frac{[\rho_{i,j}]^{\lambda\alpha}[\rho'_{i,j}]^{(1-\lambda)\alpha}[\eta_{i,j}]^{\lambda\beta}[\eta'_{i,j}]^{(1-\lambda)\beta}}{\sum_{e_{i,u}\in\Phi_{a_k}(v_i,t)}[\rho_{i,u}]^{\lambda\alpha}[\rho'_{i,u}]^{(1-\lambda)\alpha}[\eta_{i,u}]^{\lambda\beta}[\eta'_{i,u}]^{(1-\lambda)\beta}}, & \text{wenn } v_j \in \Phi_{a_k}(v_i,t) \\ 0, & \text{sonst} \end{cases} \tag{4.18}$$

definiert werden. Mit $\rho_{i,j}$, $\rho'_{i,j}$ werden die Einträge beider Pheromonmatrizen bezeichnet. Hingegen bilden $\eta_{i,j}$, $\eta'_{i,j}$ die heuristischen Informationen auf den Lösungskomponenten ab. Mit $\Phi_a(v_i,t)$ wird die Nachbarschaft von v_i zum Zeitpunkt t definiert und α sowie β gewichten allgemein die Pheromon- und Heuristikeinträge.

- **Pheromonaktualisierung:**
  Nach jeder Generation wird eine Aktualisierung der Pheromonmatrizen durchgeführt, um die Suche nach Pareto-optimalen Lösungen zu unterstützen. Zuerst werden die gefundenen Lösungen geeignet bewertet. Hierbei können zum einen die nicht dominierten Lösungen, welche in der aktuellen Generation neu hinzugekommen sind, ermittelt werden. Zum anderen können die jeweiligen Zielfunktionen als Grundlage zur Aktualisierung der Pheromonmatrizen $\mathcal{P}, \mathcal{P}'$ herangezogen werden. Nach der Aktualisierung wird eine Evaporation der Pheromonmatrizen durchgeführt.

  - *Pheromonhinzugabe:*
    Iredi et al. empfehlen in [IMM01] für eine heterogene Kolonie eine Pheromonhinzugabe basierend auf der Menge nicht dominierter Lösungen und bezugnehmend auf die aktuelle Generation vorzunehmen. Die zugehörigen Ameisen a_k fügen entsprechend

$$\Delta\rho_{i,j}^{a_k} = \begin{cases} \frac{\varrho_{ANZ}}{|A_{S_{ND}}|}, & \text{wenn } e_{i,j} \text{ Bestandteil der Lösung } s_{a_k} \text{ ist} \\ 0, & \text{sonst} \end{cases} \tag{4.19}$$

    zusätzliche Pheromone $\Delta\rho_{i,j}$ auf die jeweiligen Lösungskomponenten der Pheromonmatrizen hinzu. Mit $|A_{S_{ND}}|$ wird die Anzahl der Ameisen angegeben, die eine nicht dominierte Lösung gefunden haben und somit zur Pheromonhinzugabe berechtigt sind. Zusätzlich kann ein Faktor ϱ_{ANZ} eingeführt werden, welcher die Anzahl der zu verteilenden Pheromone definiert, um den Einfluss von nicht dominierten Lösungen auf die Suche nach neuen Lösungen zu erhöhen. Die Pheromonhinzugabe wird stets für beide Pheromonmatrizen $\mathcal{P}, \mathcal{P}'$ vorgenommen. Hierbei ist die Gesamtmenge der verteilten Pheromone in jeder Generation identisch, um den Ameisen einer jeden Generation den gleichen Einfluss zu ermöglichen. Anzumerken ist, dass dieser Ansatz nur für heterogene Pheromonmatrizen Anwendung finden kann. Pheromonmatrizen sind genau dann heterogen, wenn die zugehörigen Einträge auf unterschiedliche Lösungskomponenten abgebildet werden. Für das TCP ist diese Annahme nicht zutreffend, weil beide Pheromonmatrizen auf den gleichen Kanten abgebildet werden. Resultierend würde für eine Pheromonaktualisierung eine identische Ausprägung der Pheromonmatrizen folgen [LIS12].

Eine Pheromonhinzugabe für homogene Pheromonmatrizen $\mathcal{P}, \mathcal{P}'$ kann bezugnehmend auf die jeweilige Zielfunktion $f(x), f'(x)$ wie folgt

$$\Delta\rho_{i,j}^{a_k} = \begin{cases} \frac{\varrho ANZ}{|f(s_{a_k})|}, & \text{wenn } \rho_{i,j} \text{ Bestandteil der Lösung } s_{a_k} \text{ ist} \\ 0, & \text{sonst} \end{cases} \tag{4.20}$$

$$\Delta\rho_{i,j}'^{a_k} = \begin{cases} \frac{\varrho ANZ}{|f'(s_{a_k})|}, & \text{wenn } \rho'_{i,j} \text{ Bestandteil der Lösung } s_{a_k} \text{ ist} \\ 0, & \text{sonst} \end{cases} \tag{4.21}$$

angegeben werden.

Für beide Ansätze können die Pheromoneinträge $\rho_{i,j} \in \mathcal{P}$, entsprechend der Pheromonhinzugaben auf die Lösungskomponenten für nicht dominierte Lösungen nach einer jeden Generation wie folgt angegeben werden

$$\rho_{i,j} = \begin{cases} \rho_{i,j} + \Delta\rho_{i,j}^{a_k}, & \text{wenn } \rho_{i,j} + \Delta\rho_{i,j}^{a_k} \leq p_{\max} \\ p_{\max}, & \text{sonst.} \end{cases} \tag{4.22}$$

Hierbei stellt $p_{\max}$ eine obere Grenze für die Pheromone auf jeden Pheromoneintrag dar, um einer zu schnellen Konvergenz der Heuristik entgegenwirken zu können. Die gleiche Verfahrensweise wird für die Pheromoneinträge $\rho'_{i,j} \in \mathcal{P}'$ vorgenommen.

Es ist zu beachten, dass für beide beschriebenen Ansätze die Kolonie genügend groß sein muss, damit sich die Ameisen bezüglich λ nicht zu sehr unterscheiden. Bei zu wenigen Ameisen in einer Kolonie besteht die Gefahr, dass die Ameisen in zu großen und voneinander verschiedenen Regionen des Lösungsraumes suchen. Ferner konkurrieren die Ameisen nicht stark genug um Pareto-optimale Lösungen. Resultierend würden ebenfalls weniger gute Lösungen zu einer Verstärkung führen.

– *Pheromonevaporation:*
  Die Pheromonevaporation wird für beide Pheromonmatrizen nach der Pheromonhinzugabe durchgeführt. Hierbei richtet sich die Evaporation nach dem ACO Grundalgorithmus und wird für

$$\rho_{i,j} \leftarrow (1 - \varrho_E)\rho_{i,j}, \forall (i,j) \in \mathcal{P}. \tag{4.23}$$

  mit $0 < \varrho_E \leq 1$.

sowie analog für $\rho'_{i,j}$ angegeben.

- **Paretofront:**
  In Abhängigkeit der realisierten Pheromonhinzugabe muss abschließend eine Verwaltung der PF und zugehörigen Paretomenge vorgenommen werden. Zum einen können in jeder Generation die aktuelle PF und die neu hinzugekommenen nicht dominierten Lösungen bestimmt werden. Die Menge neu hinzugekommener nicht dominierter Lösungen wird genutzt, um kontinuierlich die Pheromonaktualisierung durchzuführen. Der Ansatz kann ebenfalls dazu dienen, die maximale Generationenanzahl als Abbruchkriterium dynamisch zu definieren. Falls für eine bestimmte Anzahl von aufeinanderfolgenden Generationen keine neue nicht

dominierte Lösung gefunden wird, ist der ACO Algorithmus beendet. Zum anderen kann es genügen, die PF zum Ende der Generationen zu ermitteln.

4.2.3. Zusammenfassung

Die Transformation des SSPP in das MOSPP bietet dem DM die Möglichkeit, ein besseres Verständnis über den Lösungsraum zu gewinnen. Vor allem können zulässige Lösungen, hinsichtlich der Ausprägungen der Robustheit, in ein Verhältnis zueinander gesetzt werden. Dem DM ist es dadurch möglich, Pareto-optimale Lösungen identifizieren und eine konkrete Lösung, angepasst an das untersuchte Problem, auswählen zu können.

Zur Bestimmung der Pareto-optimalen Lösungen können Heuristiken angewendet werden, die eine mehrkriterielle Optimierung unterstützen. Untersuchungen haben für die ACO Metaheuristik gezeigt, dass sich die Ameisen eigenständig an eine konkrete Problemstruktur mit deren spezifischen Nebenbedingungen und Zielfunktionen anpassen können. Die Anpassung wird durch einen kontinuierlichen Austausch aller gesammelten Informationen in der Kolonie begünstigt. Hierbei werden Informationen bezüglich einzelner Lösungskomponenten auf der Grundlage der Pheromone bereitgestellt. Zudem können Lösungen, die sich im Verlauf der Optimierung als ungeeignet herausgestellt haben, mit Hilfe einer stochastischen Auswahl von Lösungskomponenten überwunden werden. Grundsätzlich bietet die Metaheuristik den Vorteil, dass der Zielfunktionsraum beziehungsweise der Entscheidungsraum von den Ameisen disloziert und parallel untersucht wird. Resultierend können geeignete Lösungen oftmals schnell gefunden werden.

Neben den positiven Aspekten sind ebenfalls Nachteile der ACO Metaheuristik bekannt. Eine Implementierung und Umsetzung von ACO Algorithmen bedarf einer Vielzahl von unterschiedlichen, aufeinander abgestimmten Komponenten. Hierbei müssen allgemeine Regeln und Annahmen bezüglich der Verhaltensweisen von Ameisen bestimmt werden, die sich zu einer Kolonie zusammensetzen. Zusätzlich muss die Pheromonstruktur implementiert werden, um gesammelte Informationen, bezüglich der Suche nach geeigneten Lösungen, allen Ameisen zur Verfügung zu stellen. Die Umsetzung einer stochastischen Auswahlentscheidung von Lösungskomponenten und die sich kontinuierlich ändernde Pheromonstruktur erschwert eine theoretische Analyse bezüglich der Komplexität und Berechenbarkeit sowie Laufzeit. Zusätzlich ist die Lösungsgüte abhängig von der initialisierten Heuristik-Parameterkonfiguration. Des Weiteren können nur bedingt Aussagen über die Laufzeit bezüglich der Konvergenz getroffen werden.

Für das TCP stellt die starke Abhängigkeit zwischen der Heuristik-Parameterkonfiguration und der Lösungsgüte eine Herausforderung dar. Im Verlauf der Abdeckung werden kontinuierlich neue Informationen bezüglich der Systemumwelt hinzugewonnen. Entsprechend verändert sich die Ausprägung der den Agenten bekannten Systemumwelt dynamisch. Es können zwei Ansätze verfolgt werden, um die Lösungsgüte zu verbessern. Hierbei ist anzumerken, dass eine manuelle Anpassung während der autonomen Abdeckung nicht möglich ist. Zum einen kann vor der Abdeckung eine möglichst robuste Heuristik-Parameterkonfiguration bestimmt werden, welches für eine Vielzahl von unterschiedlichen Ausprägungen der Systemumwelt gute Lösungen liefert. Zum anderen kann eine autonome Anpassung der Heuristik-Parameterkonfiguration durch die Agenten vorgenommen werden. Hierbei kann eine quantitative Analyse der Systemumgebung erfolgen und die Ergebnisse

für eine Klassifizierung genutzt werden, welche eine Zuordnung von a priori definierten Heuristik-Parameterkonfiguration ermöglicht. Eine Diskussion beider Ansätze und die Bestimmung einer robusten Heuristik-Parameterkonfiguration wird im nachfolgenden Kapitel 4.3 vorgenommen. Daran anschließend wird eine experimentelle Analyse des implementieren ACO Algorithmus durchgeführt.

Ungleich stärker wirken sich die Vorteile für die Suche nach geeigneten Pfaden mittels autonomen Agenten im Rahmen des TCP positiv auf die Abdeckung aus. Demnach besteht die Möglichkeit, dass Anpassungen und Erweiterungen der Zielfunktionen und Problemstruktur in einer dynamischen Systemumwelt vorgenommen werden. Hierbei kann keine Lösungsgüte garantiert werden, aber es wird in jedem Fall eine zulässige Lösung gefunden. Des Weiteren ist es von Vorteil, dass die Suche im Lösungsraum oftmals nach kurzer Zeit zu möglichst geeigneten Lösungen führt. Resultierend können die Agenten mit der Aufgabenbearbeitung kontinuierlich fortfahren und eine zeitliche Unterbrechung kann vermieden werden.

4.3. Eine experimentelle Untersuchung des transformierten Single Objective Shortest Path Problem unter Unsicherheiten

Die Unsicherheiten neben den Kosten in Form einer zusätzlichen Zielfunktion zu definieren, ermöglicht die Suche nach Pareto-optimalen Lösungen. Entsprechend beschreibt die PF verschiedene Alternativen entlang von Lösungen mit niedriger Unsicherheit, die somit eine hohe Eintrittswahrscheinlichkeit der Kosten aufweisen, und Lösungen mit niedrigen Kosten. Mit Hilfe einer Ordnungsrelation ist eine Vergleichbarkeit von Routen gegeben.

In einer experimentellen Untersuchung wird die ermittelte Lösungsmenge des neuen mehrkriteriellen Lösungsansatzes mit der des ursprünglichen SSPP bewertet. Diesbezüglich werden zuerst Annahmen getroffen, die sich an realistischen Anwendungen zur Suche nach kürzesten Routen mit mobilen Robotern orientieren. Daran anschließend wird die angepasste Umsetzung der BiCriterion-Ant Heuristik beschrieben, um eine Lösung für das Mehrkriterielle Routenoptimierungsproblem zu ermitteln. Zur Analyse und quantitativen Bewertung der ermittelten PF werden geeignete Metriken eingeführt. Die Anwendung der BiCriterionAnt Heuristik bedarf der Bestimmung der Heuristik-Parameterkonfiguration. Diesbezüglich werden verschiedene Ansätze diskutiert und ein geeigneter Ansatz umgesetzt. Auf der Grundlage von Experimenten werden abschließend die Vor- und Nachteile des neuen Lösungsansatzes für das SSPP diskutiert. Ferner werden die Auswirkungen auf das TCP erörtert.

4.3.1. Annahmen

Eine reale Systemumgebung Ω wird in ein abstrahiertes Abbild Ω', in Form einer zweidimensionalen Darstellung, überführt. Des Weiteren wird eine approximierte Zerlegung in gleich große quadratische Zellen z_i mit $\Omega' \subseteq \cup_{i \in I} z_i$ vorgenommen. Die maximale horizontale und vertikale Ausdehnung der Systemumgebung ist durch die Anzahl der Zellen in jeder Dimension $\Omega'_{x,y}$ definiert. Eine Zelle

z_i wird eindeutig durch die horizontale und vertikale Lage $z_i(x, y)$ in Ω' beschrieben. Ferner kann eine Zelle z_i zwei Zustände [frei|blockiert] annehmen. Eine Zelle ist genau dann blockiert und kann nicht befahren werden, wenn sie teilweise oder vollständig durch ein Hindernis belegt wird. Andernfalls ist eine Zelle frei und kann durch einen Agenten abgedeckt werden. Zusammenfassend kann die Umgebung als ein planarer Graph beschrieben werden. Die Zellen entsprechen den Knoten und die Zellübergänge den Kanten.

Ein Agent kann sich von einer Zelle in alle acht direkt angrenzenden Zellen bewegen, insofern diese frei sind. Zwei Zellen sind direkt miteinander verbunden, wenn sie eine gleiche Systemgrenze teilen. Das Befahren einer Zelle verursacht Kosten, die abhängig von der Strömungsrichtung und der Fahrtrichtung des Agenten sind. Die Kosten sind auf den Zellen definiert. Die Strömungsrichtung ist für jede freie Zelle definiert und kann aus allen acht Richtungen kommen. Die Strömungsrichtungen orientieren sich dabei an den acht möglichen Fahrtrichtungen/Zellübergängen. Die Informationen bezüglich der Strömungsrichtung werden für jede Zelle z_i mit einem Einheitsvektor $\mathbf{d}_R(z_i)$ beschrieben. Hierbei werden die Eintrittswahrscheinlichkeiten für die acht möglichen Strömungsrichtungen angegeben. Liegen keine Informationen über die Strömungsrichtung vor, dann wird die Strömungsrichtung mit $\mathbf{d}_R(z_i) = (.125\ \ .125\ \ .125\ \ .125\ \ .125\ \ .125\ \ .125\ \ .125)^T$ definiert. Sind hingegen vollständige Informationen bezüglich der Strömungsrichtung verfügbar, dann kann für eine Strömungsrichtung aus Süden kommend nach Norden mit $\mathbf{d}_R(z_i) = (1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0)^T$ angegeben werden. Ein Beispiel für unterschiedliche Strömungsinformationen in einer Systemumgebung können der Abbildung 4.4 entnommen werden.

Die Zusammenhänge der Strömungsrichtung und der resultierenden Kosten für die Fahrtrichtungen sind in der Abbildung 4.5 dargestellt. Die Agenten befahren jede Zelle in gleicher Zeit, unabhängig der Strömungsrichtung. Demnach muss ein Agent mehr Energie aufwenden, um in gleichbleibender Geschwindigkeit eine benachbarte Zelle zu besuchen, für die eine frontale Gegenströmung vorliegt. Der erhöhte Energiebedarf verursacht höhere Kosten. Hingegen verursacht die Befahrung einer benachbarten Zelle mit einer Strömungsrichtung in Fahrtrichtung einen geringeren Energieverbrauch. Die Kosten $c(z_i)$ für das Befahren einer Zelle z_i definieren sich zu:

$$c(z_i) = \sum_{q=1}^{8} d_{Rq}(z_i) \ \ w \tag{4.24}$$

mit

$$w = \begin{cases} 1, & \text{wenn } \sphericalangle \ \ (\text{Fahrtrichtung, Strömungsrichtung}) = 0^\circ \\ 2, & \text{wenn } \sphericalangle \ \ (\text{Fahrtrichtung, Strömungsrichtung}) = 45^\circ \\ 3, & \text{wenn } \sphericalangle \ \ (\text{Fahrtrichtung, Strömungsrichtung}) = 90^\circ \\ 4, & \text{wenn } \sphericalangle \ \ (\text{Fahrtrichtung, Strömungsrichtung}) = 135^\circ \\ 5, & \text{wenn } \sphericalangle \ \ (\text{Fahrtrichtung, Strömungsrichtung}) = 180^\circ, \end{cases} \tag{4.25}$$

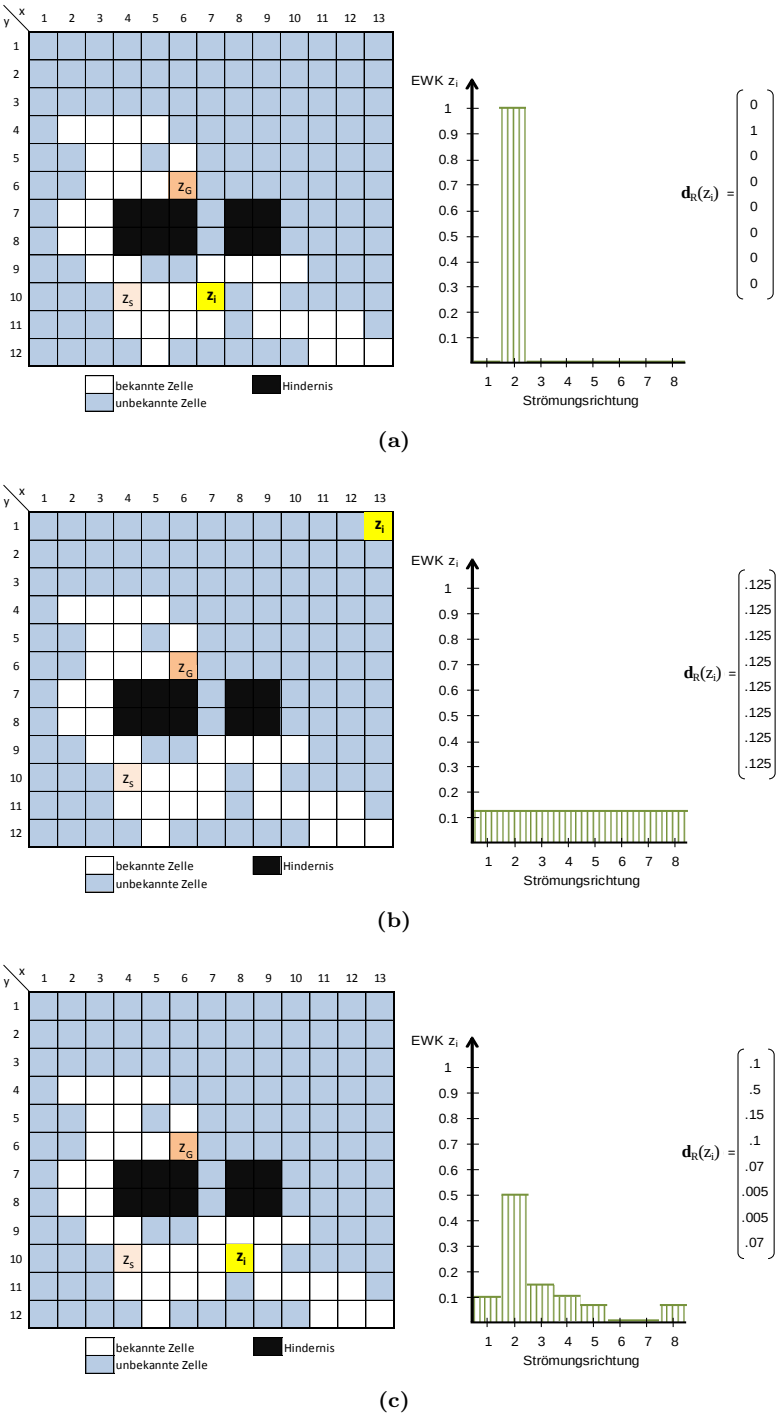


Abbildung 4.4.: Unterschiedliche Charakteristika von möglichen Strömungsinformation in einer Systemumgebung. Unter (a) Bekannte Zelle: es liegen vollständige Strömungsinformationen vor, (b) Unbekannte Zelle: es liegen keine Strömungsinformation vor, (c) Unbekannte Zelle: es liegen approximierte Strömungsinformationen vor.

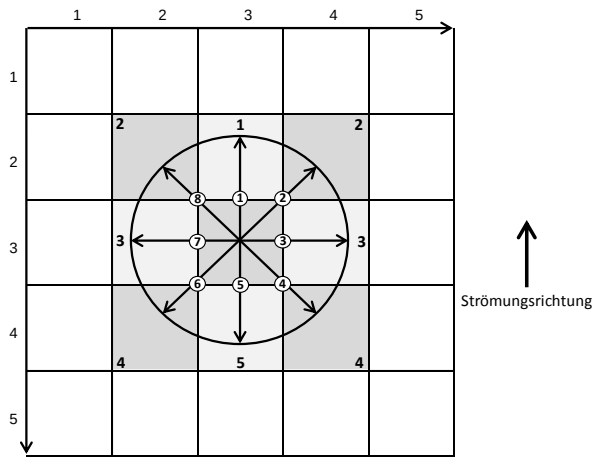


Abbildung 4.5.: Modell der Strömungskosten für eine approximierte Umgebungszerlegung. Bewegungsmöglichkeiten eines Agenten in alle acht angrenzenden Zellen. Bei einer Strömungsrichtung aus Süden kommend, resultiert für die Bewegungsrichtungen nachfolgendes Kostenprofil (Fahrtrichtung → Kosten): 1→1; 2→2; 3→3; 4→4; 5→5; 6→4; 7→3; 8→2;.

Bezugnehmend auf die Abbildung 4.5 resultieren für die nachfolgenden Beispiele, unter der Annahme, dass die Zelle z_i mit der Fahrtrichtung 1 (von Süden nach Norden) befahren wird, nachfolgend beschriebene Kosten. Für die Befahrung der Zelle z_i, unter vollständigen Informationen $\mathbf{d}_R(z_i) = (1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0)^T$, ergeben sich Kosten in Höhe von einer Einheit. Unter der Annahme, dass keine Informationen bezüglich der Strömungsrichtung $\mathbf{d}_R(z_i) = (.125\ \ .125\ \ .125\ \ .125\ \ .125\ \ .125\ \ .125\ \ .125)^T$ vorliegen, folgen Kosten von drei Einheiten. Für eine approximierte Strömungsrichtung gemäß $\mathbf{d}_R(z_i) = (.1\ \ .5\ \ .15\ \ .1\ \ .07\ \ .005\ \ .005\ \ .07)^T$ folgen Kosten in Höhe von 2.475 Einheiten.

4.3.2. Funktionsweise

Die Ameisenkolonie A setzt sich aus heterogenen Ameisen a_k zusammen, denen jeweils eine eindeutige Kennung k mit $k \in \{1, \ldots, |A|\}$ zugewiesen ist. Um eine parallele und dislozierte Suche nach Lösungen im Lösungsraum zu ermöglichen, wird für jedes a_k ein Gewichtungsfaktor λ_k mit $\frac{k-1}{|A|-1}$ bestimmt. Die Informationen aus der Systemumgebung werden auf der Grundlage von zwei Heuristikmatrizen repräsentiert. Die Fahrtkosten werden gemäß der Strömungsrichtung und jeweiligen Fahrtrichtung ermittelt. Aus diesem Grund spiegelt die Heuristikmatrix $\mathcal{H}^K$ die (approximierten) Fahrtkosten wider. Die Bestimmung der $\mathcal{H}^K$ für eine Zelle z_i erfolgt auf der Grundlage der aktuellen Strömungsrichtungsvektoren. Des Weiteren wird die notwendige Fahrtrichtung einbezogen, welche aus der aktuellen Zelle der Ameise zur zu besuchenden Zelle z_i ermittelt wird. Die (approximierte) Strömungsrichtung kann für jede freie Zelle angegeben werden. Entsprechend sind die heuristischen Informationen $\mathcal{H}^K$ wie folgt definiert:

$$\mathcal{H}^K(z_i) = \begin{cases} c(z_i), & \text{wenn } z_i \in \Omega' \backslash \{z | z \text{ ist blockiert}\} \\ \infty, & \text{sonst.} \end{cases} \tag{4.26}$$

Die heuristischen Informationen $\mathcal{H}^{EWK}$, bezüglich der Eintrittswahrscheinlichkeit der Strömungsrichtung für eine Zelle, sind durch den zugehörigen Strömungsrichtungsvektor $\mathbf{d}_R(z_i)$ definiert.

Des Weiteren werden zwei Pheromonmatrizen $\mathcal{P}^K$, $\mathcal{P}^{EWK}$ eingeführt. Die Pheromonmatrizen kodieren jeweils in den Generationen gesammelte Informationen bezüglich der Suche nach geeigneten Routen. Hierbei wird die Minimierung der Kosten und Maximierung der Eintrittswahrscheinlichkeit betrachtet. Beide Pheromonmatrizen werden im Sinne eines $\mathcal{MAX}-\mathcal{MIN}$ Ant System modelliert. Resultiert sind die Einträge der Pheromonmatrizen mit $\rho_{\min}$ vorinitialisiert. Die Gewichtung der heuristischen Informationen und der Pheromoninformationen wird über die Gewichtungsfaktoren α und β definiert. Zusätzlich wird die Anzahl der Pheromone ϱ_{ANZ} festgelegt, die auf gute Lösungskomponenten im Rahmen der Pheromonaktualisierung abgelegt werden. Ebenfalls wird die Pheromonevaporationsrate ϱ_E und die maximale Anzahl an Generationen $\max_{Generation}$ initialisiert. Zusammenfassend wird eine Suche nach geeigneten Routen zwischen der Startzelle z_S sowie Zielzelle s_G und ferner der Bestimmung einer PF sowie der zugehörigen Paretomenge durchgeführt. Im Anschluss an die heuristische Suche wird aus der Menge nicht dominierter Lösungen $\mathcal{S}_{ND}$ eine konkrete Route ausgewählt und diese fungiert als Rückgabewert. Der Gesamtablauf der implementierten BiCriterionAnt Heuristik ist als Algorithmus 4.3 beigefügt und wird nachfolgend erläutert.

Die heuristische Suche nach Routen wird solange durchgeführt, bis das Abbruchkriterium in Form einer maximalen Anzahl von Generationen
$\max_{Generation}$ erreicht ist. In jeder Generation ermitteln alle Ameisen $a_k \in A$ eine zulässige Route r_{a_k} zwischen z_S und z_G. Hierbei ist eine Ameise genau solange lebendig bis sie an der Zielzelle z_G angekommen ist. Für alle lebenden Ameisen einer Generation wird eine stochastische Auswahl einer zulässigen Lösungskomponente, sprich Zelle in der aktuellen Nachbarschaft $\Phi_{a_k}(z_{Pos}, t)$ der Ameise a_k vorgenommen. Grundsätzlich ist die Nachbarschaft $\Phi_{a_k}(z_{Pos}, t)$ über alle Zellen definiert, die direkt an die aktuelle Zelle der Ameise z_{Pos} angrenzen. Darüber hinaus wird für die Nachbarschaft genau die Teilmenge ermittelt, die den kleinsten Zählerstand auf den Zellen aufweist. Der Zählerstand beschreibt die Anzahl der Besuche auf einer Zelle, die eine Ameise auf der Suche nach einer Route zwischen z_S und z_G vornimmt. Resultierend wird das Node Counting (NC) Modell integriert. Mit Hilfe des NC Modells kann eine vollständige Abdeckung einer Systemumgebung garantiert werden. Entsprechend kann in jedem Fall eine Route zwischen Start und Ziel gefunden werden. Die Integration des NC Modells ist insofern notwendig, da die Suche auf einem ungerichteten zyklischen Graphen vorgenommen wird. Ohne eine zielgerichtete Suche nach der Zielzelle z_G, bestimmt ausschließlich der Zufall, ob eine Route zwischen z_S und z_G gefunden wird. Neben der Integration des NC Modells und somit indirekten Ausrichtung auf die Zelle z_G, kann eine direkte Suche nach der Zielzelle z_G integriert werden. Hierbei wird zusätzlich die heuristische Information bezüglich der Entfernung zu z_G für die Bestimmung der Auswahlwahrscheinlichkeit einer Zelle berücksichtigt. In dieser Dissertation wird der direkte Ansatz nicht weiter berücksichtigt. Zum einen geht mit dem direkten Ansatz in die Optimierung der in dieser Dissertation betrachteten Zielfunktionen zusätzlich die Optimierung der Routenlänge ein. Resultierend werden kürzere Routen tendenziell eher bevorzugt. In dieser Dissertation sollen jedoch Routen ausschließlich gemäß den Kosten und der zugehörigen Eintrittswahrscheinlichkeit optimiert werden. Zum anderen ist das integrierte NC Modell innerhalb der ACO Heuristik bisher in der Literatur noch nicht beschrieben.

Die Auswahlwahrscheinlichkeit der Zellen $p_{a_k}(z_i)$ mit $z_i \in \Phi_{a_k}(z_{Pos}, t)$ wird zusammenfassend mit

$$p_{a_k}(z_i) = \frac{\rho^{EWK}(z_i)^{\lambda_{a_k}*\alpha} * \rho^K(z_i)^{(1-\lambda_{a_k})*\alpha} * \eta^{EWK}(z_i)^{\lambda_{a_k}*\beta} * \eta^K(z_i)^{(1-\lambda_{a_k})*\beta}}{\sum_{z_n \in \Phi_{a_k}(z_i,t)} \rho^{EWK}(z_n)^{\lambda_{a_k}*\alpha} * \rho^K(z_n)^{(1-\lambda_{a_k})*\alpha} * \eta^{EWK}(z_n)^{\lambda_{a_k}*\beta} * \eta^K(z_n)^{(1-\lambda_{a_k})*\beta}} \tag{4.27}$$

bestimmt. Der Gewichtungsfaktor λ_k bestimmt die Suche nach Lösungen gemäß der Minimierung der Kosten und Maximierung der Eintrittswahrscheinlichkeit. Für die Extremfälle ($\lambda_k = 0 \wedge 1$) existiert jeweils eine Ameise, die ausschließlich eines der beiden Ziele betrachtet. Die anderen Ameisen suchen nach Kompromisslösungen entsprechend der Gewichtungen. Die Auswahl einer konkreten Lösungskomponente erfolgt über eine Monte-Carlo-Auswahl. Hierbei werden die Auswahlwahrscheinlichkeiten kumuliert und anschließend eine Zufallszahl für das Intervall $[0, 1]$ gezogen. Entsprechend liegt die Zufallszahl mit einer höheren Wahrscheinlichkeit in einem Bereich, der durch eine Zelle mit höherer Auswahlwahrscheinlichkeit im Intervall $[0, 1]$ beschrieben ist. Nachdem eine konkrete Zelle aus $\Phi_{a_k}(z_{Pos}, t)$ bestimmt ist, bewegt sich die Ameise zu dieser. Hat die Ameise im Verlauf der aktuellen Suche nach Routen diese Zelle bereits besucht, dann wird der resultierende Zyklus gelöscht. Die Suche nach einer zulässigen Route wird solange durchgeführt, bis die Ameise an der Zelle z_G angekommen ist. Anschließend stirbt die Ameise. Hat jede Ameise eine Route r_{a_k} bestimmt, werden für jede Route die Gesamtkosten, die zugehörige Eintrittswahrscheinlichkeit und die Anzahl der Richtungsänderungen ermittelt. Hierbei werden die Gesamtkosten entsprechend der konkreten Route, einschließlich der angenommenen Strömungskosten auf jeder Zelle, für jede Ameise ermittelt. Die Eintrittswahrscheinlichkeit wird mittels einer zufälligen Stichprobe auf jeder Route im Umfang von 10.000 Ziehungen approximiert. Hierbei gibt das Verhältnis zwischen den Ziehungen mit Kosten kleiner gleich der Kosten der betrachteten Route und der Gesamtanzahl aller 10.000 Ziehungen die Eintrittswahrscheinlichkeit an. Darauffolgend wird die PF über alle Generationen aktualisiert. Um die Suche nach geeigneten Lösungen unterstützen zu können, wird eine Aktualisierung der Pheromonmatrizen auf der Grundlage der in der letzten Generation neu hinzugekommenen Pareto-optimalen Routen ermittelt.

Die Güte einer Pareto-optimalen Route wird jeweils anhand der beiden Ziele bewertet. Für die Minimierung der Kosten wird für das betrachtete Optimierungsproblem O die theoretisch kostengünstigste Route $\min_{\text{Kosten}}(O)$ ermittelt. Für jede unbekannte Zelle wird angenommen, dass die resultierenden Kosten stets minimal sind in Abhängigkeit der Fahrtrichtung. Es folgt das SPP, für das alle Informationen vorliegen, welches beispielsweise mittels des Dijkstra-Algorithmus effizient gelöst werden kann. Die kostengünstigste Route $\min_{\text{Kosten}}(O)$ dient als Referenzroute zur Bestimmung der Güte aller Pareto-optimalen Lösungen $s_{a_k} \in S_{ND}$. Des Weiteren kann für jede Pareto-optimalen Lösungen $s_{a_k} \in S_{ND}$ die Eintrittswahrscheinlichkeit bestimmt werden. Es wird die Maximierung der Eintrittswahrscheinlichkeit verfolgt.

Resultierend können die Pheromonmengen, die zusätzlich auf den Lösungskomponenten der Pheromonmatrizen $\mathcal{P}^K$ und $\mathcal{P}^{EWK}$ abgelegt werden, wie folgt angegeben werden:

$$\Delta\rho_{i,j}^{K}, = \begin{cases} \varrho_{ANZ} \frac{min_{\text{Kosten}}(O)}{\text{Kosten}(r_{a_k})}, & \text{wenn } \rho_{i,j}^{K} \text{ Bestandteil der Lösung } s_{a_k} \text{ ist} \\ 0, & \text{sonst} \end{cases} \tag{4.28}$$

$$\Delta\rho_{i,j}^{EWK} = \begin{cases} \varrho_{ANZ}\text{EWK}(r_{a_k}), & \text{wenn } \rho_{i,j}^{EWK} \text{ Bestandteil der Lösung } s_{a_k} \text{ ist} \\ 0, & \text{sonst} \end{cases} \tag{4.29}$$

Der Ablauf zur Bestimmung einer Route, nach der BiCriterionAnt Heuristik, ist zusammenfassend in dem Algorithmus 4.3 angegeben. Einen wesentlichen Einfluss auf die Lösungsgüte nimmt die

Ausprägung der Heuristik-Parameterkonfiguration. Dazu werden im nächsten Kapitel verschiedene Metriken eingeführt, um die Lösungsgüte in Form einer PF quantifizieren zu können. Daran anschließend werden Ansätze zur Bestimmung einer optimierten Parameterkonfiguration diskutiert. Abschließend wird unter der Annahme, dass die Routenplanung von autonomen Agenten durchgeführt werden soll, ein geeigneter Ansatz realisiert.

4.3.3. Der IRace Algorithmus - Eine offline Parameterkonfiguration

Die BiCriterionAnt Heuristik gehört zur Klasse der ACO Meta-Heuristik und kann für eine Vielzahl von verschiedenartigen Problemen der kombinatorischen Optimierung zur Anwendung kommen. Die generalisierte Verwendung der Heuristiken wird vor allem auf der Grundlage von einer iterativen, stochastischen Verbesserung der Problemlösung und der Definition einer Parameterkonfiguration ermöglicht. Im Rahmen der Instantiierung der konkreten Algorithmen werden die Parameterkonfiguration und resultierend die verschiedenen Komponenten der Heuristik auf das konkrete Problem angepasst. Ferner ermöglicht die Parameterkonfiguration eine gezielte Steuerung der Suche nach Lösungen. Hierbei ist die Lösungsgüte maßgeblich von der Heuristik-Parameterkonfiguration abhängig. In [Bir02, AL06] wird des Weiteren gezeigt, dass die Ermittlung einer optimierten Parameterkonfiguration für Heuristiken zur Klasse der $\mathcal{NP}$-schweren Probleme gehört.

Die Parameterkonfiguration für die BiCriterionAnt Heuristik ist eine Zusammensetzung aus den Angaben hinsichtlich der maximalen Anzahl von Generationen $\max_{Generation}$, der Anzahl der Ameisen $|A|$, den Gewichtungsfaktoren α, β sowie der Evaporationsrate der Pheromone ϱ_E und der Anzahl von Pheromonen ϱ_{ANZ}, die in jeder Generation auf die Komponenten einer geeigneten Problemlösungen abgelegt werden. Im gleichen Ausmaß ist jedoch auch die Konfiguration der Parameterkonfiguration abhängig von der Ausprägung der Probleminstanz. Für das in dieser Dissertation betrachtete MOSPP unterscheiden sich Probleminstanzen hinsichtlich der räumlichen Ausdehnung einer Systemumgebung, der Entfernung zwischen Start und Ziel, der Lage von Start und Ziel innerhalb der Systemumgebung, der Lage und Ausprägung von nicht befahrbaren Hinderniszellen und den Informationen bezüglich der zu erwartenden Kosten auf allen Zellen. Allein aufgrund der Definition der Strömungsrichtungsvektoren existiert eine nicht endliche Menge von Probleminstanzen für das übergeordnete TCP. Aus diesem Grund liegt eine Besonderheit und zugleich Herausforderung in der Anwendung von autonomen Agenten vor. Hierbei müssen die autonomen Agenten für eine vollständige Abdeckung wiederholt die BiCriterionAnt Heuristik zur Suche nach geeigneten Routen im Rahmen des MOSPP ausführen. Aufgrund der Abhängigkeit zwischen Lösungsgüte und der Parameterkonfiguration auf Basis der konkret vorliegenden Probleminstanz müsste idealerweise für jede auftretende Probleminstanz eine optimierte Parameterkonfiguration bestimmt werden.

Ein intuitiver Ansatz besteht daher darin, vor der autonomen Abdeckung für jede mögliche Probleminstanz eine optimierte Parameterkonfiguration zu ermitteln. Anschließend werden die optimierten Parameterkonfigurationen entsprechend der auftretenden Probleminstanzen zur Problemlösung des MOSPP genutzt. Für das in dieser Dissertation zu betrachtende TCP und ferner die wiederholte Anwendung des MOSPP wird dieser Ansatz, aufgrund der sehr hohen Anzahl von möglichen Probleminstanzen, nicht weiter verfolgt. Des Weiteren besteht jedoch grundsätzlich die Möglichkeit, dass die autonomen Agenten parallel zur Abdeckung für jedes zu betrachtende MOSPP vor der

Algorithmus 4.3 : BiCriterionAnt – Routenplanung

Eingabe : z_S, z_G, $|A|$, $\max_{Generation}$, α, β, $\mathcal{P}^{EWK}$, $\mathcal{P}^K$, ϱ_{ANZ}, ϱ_E, $\rho_{\min}$, $\rho_{\max}$, ρ_{ini}
Ausgabe : $r \in \mathcal{R}$

Initialisiere: $a_k \in A$ für $k = 1, \ldots, |A|$ mit $\lambda_{a_k} = \frac{k-1}{\|A|-1}$, $A_{lebend} = A$, z_S, z_G, α, β, $\mathcal{P}^{EWK}$ und $\mathcal{P}^K$ mit $\rho_{\min}$, $\rho_{\max}$, ρ_{ini}, $\mathcal{S}_{ND} = \emptyset$, $g_j = 1$
while $g_j <= \max_{Generation}$ **do**
  while $A_{lebend}! = \emptyset$ **do**
    foreach $a_k \in A_{lebend}$ **do**
      Bestimme $p_{a_k}(z_i) \forall z_i \in \Phi_{a_k}(z_{Pos})$ mit

$$p_{a_k}(z_i) = \frac{\rho^{EWK}(z_i)^{\lambda_{a_k}*\alpha} * \rho^K(z_i)^{(1-\lambda_{a_k})*\alpha} * \eta^{EWK}(z_i)^{\lambda_{a_k}*\beta} * \eta^K(z_i)^{(1-\lambda_{a_k})*\beta}}{\sum_{z_n \in \Phi_{a_k}(z_i,t)} \rho^{EWK}(z_n)^{\lambda_{a_k}*\alpha} * \rho^K(z_n)^{(1-\lambda_{a_k})*\alpha} * \eta^{EWK}(z_n)^{\lambda_{a_k}*\beta} * \eta^K(z_n)^{(1-\lambda_{a_k})*\beta}}$$

      Ermittle z_i mit Monte-Carlo-Auswahl
      Gehe zu z_i
      if $z_i \in r_{a_k}$ **then**
        Lösche Zyklus aus r_{a_k}
      end
      if $z_i == z_G$ **then**
        a_k stirbt, d.h. $A_{lebend} = A_{lebend} \setminus a_k$
      end
    end
  end
  alle Ameisen haben z_G erreicht
  foreach $a_k \in A$ **do**
    Ermittle für r_{a_k} (Kosten, Eintrittswahrscheinlichkeit, #Richtungsänderungen)
    Ermittle $\mathcal{S}_{ND}$ der aktuellen Generation
  end
  foreach $s_j \in \mathcal{S}_{ND}$ *und in aktueller Generation gefunden* **do**
    Aktualisiere $\mathcal{P}^K$

$$\Delta\rho^K_{i,j} = \begin{cases} \varrho_{ANZ} \frac{min_{\text{Kosten}}(O)}{\text{Kosten}(r_{a_k})}, & \text{wenn } \rho^K_{i,j} \text{ Bestandteil der Lösung } s_j \text{ ist} \\ 0, & \text{sonst} \end{cases}$$

    Aktualisiere $\mathcal{P}^{EWK}$

$$\Delta\rho^{EWK}_{i,j} = \begin{cases} \varrho_{ANZ}{}^{EWK}(r_{a_k}), & \text{wenn } \rho^{EWK}_{i,j} \text{ Bestandteil der Lösung } s_j \text{ ist} \\ 0, & \text{sonst} \end{cases}$$

  end
  Evaporiere $\mathcal{P}^K$

$$\rho^K_{i,j} = \max(0, \min(100, 1 - \varrho_E * \rho^K_{i,j}))$$

  Evaporiere $\mathcal{P}^{EWK}$

$$\rho^{EWK}_{i,j} = \max(0, \min(100, 1 - \varrho_E * \rho^{EWK}_{i,j}))$$

  foreach $a_k \in A$ **do**
    Rekonfiguriere a_k mit ($z_{Pos} = z_S$, $r_{a_k} = \emptyset$, $\mathcal{P}^{EWK}$, $\mathcal{P}^K$)
  end
  Rekonfiguriere $A_{lebend} = A$
  Erhöhe aktuelle Generation $g_j = g_j + +$
end
Wähle $s \in \mathcal{S}_{ND}$ aus
return zugehörige $r \in \mathcal{R}$ Route zu s

Routenplanung eine Optimierung der Heuristikparameter des BiCriterionAnt Algorithmus selbstständig durchführen. Das Vorgehen wird ebenfalls als nicht zielführend erachtet, weil in der Regel nicht genügend Zeit zur Bestimmung einer Parameterkonfiguration zur Verfügung steht. Resultierend müssten die Agenten ihre aktuelle Position halten und könnten erst verzögert mit der Abdeckung der Systemumgebung fortfahren.

Zum anderen besteht die Möglichkeit, vor der Abdeckung der Systemumgebung die Probleminstanzen anhand von charakteristischen Eigenschaften der Systemumgebung in disjunkte Klassen einzuteilen. Anschließend wird für jede Klasse von ähnlichen Probleminstanzen eine geeignete Parameterkonfiguration bestimmt. Mit diesem Vorgehen wird die Anzahl der zu bestimmenden Parameterkonfigurationen auf die Anzahl der Klassen reduziert. Ferner ist mit diesem Vorgehen ebenfalls eine Abnahme der Lösungsgüte hinsichtlich des MOSPP verbunden, weil keine individuell optimierte Parameterkonfiguration angegeben wird. Für eine Bestimmung der Klassen von Probleminstanzen und für Zuordnung der auftretenden Probleminstanzen kann unter anderem die Systemumgebung in einen Graphen transformiert werden. Anschließend kann mittels der Quantitativen Graphentheorie die aktuell vorliegende Systemumgebung analysiert und einer Klasse zugeordnet werden. Der Ansatz wird in [PDPH14] beschrieben. Beispielsweise kann eine Zuordnung von Probleminstanzen zu den zuvor definierten Klassen von gleichartigen Probleminstanzen mit Hilfe von Künstlich Neuronalen Netzen erlernt werden. Der beschriebene Ansatz kann grundsätzlich verfolgt werden, aber aufgrund der eingeschränkt verfügbaren Informationen bezüglich der Systemumgebung vor einer Abdeckung, ist die Ermittlung von Klassen gleichartiger Probleminstanzen nur bedingt möglich. Mitunter kann während der Abdeckung keine eindeutige Zuordnung, der aktuell vorliegenden Probleminstanz, zu den zuvor definierten Klassen bestimmt werden. Des Weiteren kann die Klassifizierung der Probleminstanzen parallel zur Abdeckung zum Halten der aktuellen Position und resultierend zu Verzögerungen führen.

In dieser Dissertation wird der IRace Algorithmus zur Bestimmung einer optimierten Parameterkonfiguration umgesetzt. Der IRace Algorithmus ermöglicht eine offline Parameterkonfiguration. Ausgehend von einer Menge repräsentativer Probleminstanzen wird in einer Trainingsphase eine geeignete Parameterkonfiguration für die BiCriterionAnt Heuristik ermittelt. Hierbei führt die Parameterkonfiguration für eine Vielzahl von Probleminstanzen zu einer optimierten Performance der BiCriterionAnt Heuristik und resultierend zu guten Ergebnissen. Anschließend wird die Parameterkonfiguration in einer Produktionsphase genutzt, um zuvor in der Trainingsphase nicht betrachtete Probleminstanzen ebenfalls lösen zu können. Resultierend liegt ein Generalisierungsproblem vor, bei dem eine geeignete Parameterkonfiguration für eine (nicht endliche) Menge von Probleminstanzen gesucht wird [BYBS09, LIDLSB11]. Der allgemeine Aufbau des IRace Algorithmus ist in dem Algorithmus 4.4 nach [LIDLSB11] beigefügt.

Zu Beginn der Suche nach einer optimierten Parameterkonfiguration wird das maximale Budget angegeben, welches bei Erreichen zum Abbruch der Suche führt. Mit Hilfe des Budgets wird festgelegt, wieviele Parameterkonfigurationen in jeder Iteration überprüft werden. Des Weiteren wird der Parameterraum der Parameterkonfiguration angegeben und für jeden Parameter eine unabhängige Wahrscheinlichkeitsverteilung definiert. Ferner wird eine Sequenz Ψ an repräsentativen Probleminstanzen ψ_j festgelegt. Zu Beginn der Suche wird eine Ausgangsmenge an Parameterkonfigurationen entsprechend der Wahrscheinlichkeitsverteilung auf den Parametern erzeugt. Daran anschließend wird eine Bewertung der Parameterkonfigurationen anhand einer Probleminstanz

Algorithmus 4.4 : Grundalgorithmus für IRace

Eingabe : $\text{Budget}_{\max}$, Probleminstanzen Ψ, Definition einer Parameterkonfiguration $k_i \in K$
Bewertungsmaß $f : K \times \Psi \to \mathbb{R}$
Ausgabe : K^{elite}

Initialisiere: Erzeuge Sequenz von repräsentativen Probleminstanzen $\psi_j \in \Psi$, Bestimme Definitionsbereich der Parameter für die Parameterkonfigurationen $k_i \in K$, Laufvariable $j = 1$

Erzeuge Menge von Parameterkonfigurationen $K_j \subseteq K$ anhand von Wahrscheinlichkeitsverteilung auf Parametern
Bestimme K^{elite} und führe dazu $\text{Race}(K_j, \text{Budget}_j)$ durch mit
    Evaluiere jede Parameterkonfiguration $k_i \in K_j$ anhand ψ_j
    Führe statistischen Test durch zur Bestimmung unterlegener Parameterkonfigurationen
    Eliminiere gegebenenfalls schlechte Parameterkonfigurationen
j=j+1
while $Budget_{verbraucht} \leq Budget_{\max}$ **do**
    Bestimme K^{neu} anhand angepasster Wahrscheinlichkeitsverteilung auf Parametern
    Bestimme $K_j = K^{neu} \cup K^{\text{elite}}$
    Bestimme K^{elite} und führe dazu $\text{Race}(K_j, \text{Budget}_j)$ durch
    j=j+1
end
return Menge von Parameterkonfigurationen K^{elite}

durchgeführt. Mit Hilfe von statistischen Tests kann überprüft werden, ob Parameterkonfigurationen gegenüber anderen unterlegen sind und für eine weitere Betrachtung verworfen werden können. In [Bir02, BYBS09] werden der Friedman-Test und ein t-Test vorgeschlagen. Auf der Grundlage der weiter betrachteten Parameterkonfigurationen und angepassten Wahrscheinlichkeitsverteilungen auf den Parametern wird eine neue Menge von Parameterkonfigurationen ermittelt. Resultierend wird mit der Aktualisierung der Wahrscheinlichkeitsverteilungen auf den Parametern auf die Ermittlung von zukünftigen Parameterkonfigurationen Einfluss genommen. Hierbei kann eine trunkierte Verteilung entsprechend der Parameterkonfigurationen in K^{elite} verwendet werden. Dementsprechend liegt eine höhere Wahrscheinlichkeit vor, dass Parameter ähnlich der im Verlauf besten Parameterkonfigurationen erzeugt werden. Basierend auf der Menge der neu ermittelten Parameterkonfigurationen K^{neu} und den Parameterkonfigurationen der vorhergehenden Iteration K^{elite} wird wiederholt eine Bewertung und Überprüfung der Parameterkonfigurationen K_j durchgeführt. Jede Überprüfung einer Parameterkonfiguration anhand einer Probleminstanz reduziert das noch zur Verfügung stehende Budget. Ist das maximale Budget aufgebraucht, werden die besten beziehungsweise überlegenen Parameterkonfigurationen ausgegeben. Die Rangfolge der Parameterkonfigurationen wird gemäß der gemittelten Performance über alle Probleminstanzen bestimmt.

Die Vorteile gegenüber dem intuitiven Brute-Force Ansatz bestehen vor allem darin, dass unterlegene Parameterkonfigurationen verworfen werden. Das heißt, die systematische und frühzeitige Eliminierung von nachweislich unterlegenen Parameterkonfigurationen führt zu einer effizienteren Verwendung der Rechenkapazitäten. Resultierend werden die verfügbaren Rechenkapazitäten für die Evaluation guter Parameterkonfigurationen beziehungsweise zur Suche nach neuen, potentiellen Parameterkonfigurationen verwendet. Entsprechend können mit gleichem Rechenaufwand im Racing Ansatz mehr Probleminstanzen betrachtet werden als mit dem Brute-Force Ansatz [BYBS09]. In der Abbildung 4.6 wird der Unterschied zwischen dem Brute-Force Ansatz und dem in dieser Dissertation implementieren Racing Ansatz dargestellt.

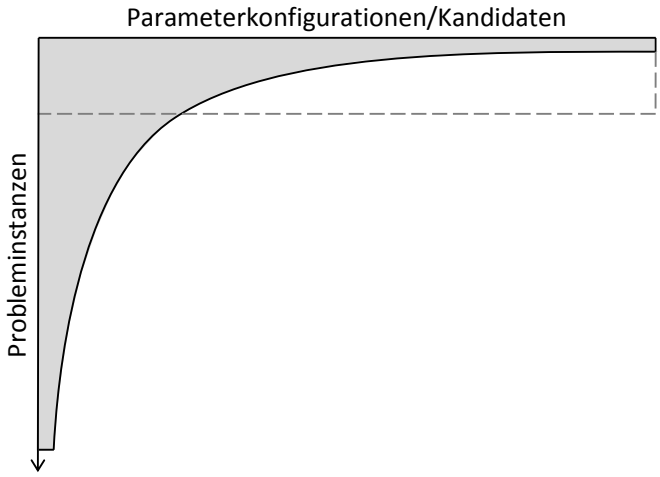


Abbildung 4.6.: Veranschaulichung des Racing Ansatz zur Ermittlung einer geeigneten Parameterkonfiguration (graue Fläche). Im Verhältnis dazu ist das Vorgehen im Brute-Force Ansatz dargestellt (gestricheltes Rechteck). Die Flächen beider Ansätze sind gleich groß und demnach auch die Anzahl der durchgeführten Experimente (nach [BYBS09], Seite 6).

4.3.3.1. Untersuchungsgegenstand und Metriken

Den Untersuchungsgegenstand im Rahmen der offline Parameterkonfiguration stellt die, dem BiCriterionAnt Algorithmus zugrundeliegende, Heuristik-Parameterkonfiguration dar. Die Heuristik-Parameterkonfiguration beschreibt hierbei die wesentlichen Komponenten der übergeordneten ACO Metaheuristik. Abhängig von der Problemdefinition müssen die Komponenten geeignet definiert und implementiert werden. Der BiCriterionAnt Algorithmus soll im weiteren Verlauf der Dissertation in ein TCP integriert werden und autonomen Agenten dazu dienen, ein Ergebnis für das MOSPP zu liefern. Im Rahmen der Offline Parameterkonfiguration soll mittels des IRace Algorithmus eine automatisierte Bestimmung der nachfolgenden Parameter vorgenommen werden:

- α - Gewichtungsfaktor zur Berücksichtigung der Pheromoninformationen hinsichtlich der Auswahlwahrscheinlichkeit einer Lösungskomponente
- β - Gewichtungsfaktor zur Berücksichtigung der heuristischen Informationen hinsichtlich der Auswahlwahrscheinlichkeit einer Lösungskomponente
- ϱ_E - Evaporationsrate der Pheromone in jeder Generation
- ϱ_{ANZ} - Anzahl von Pheromonen, die auf geeignete Lösungskomponenten einer Generation abgelegt werden

Die Anzahl der Generationen und die Anzahl der Ameisen wird vorgegeben. Zum einen kann gezeigt werden, dass mit steigender Anzahl von Generationen und Ameisen eine Verbesserung der Lösungsgüte einhergeht. Des Weiteren sind ϱ_E und ϱ_{ANZ} selbst abhängig von der Anzahl der Generationen und Ameisen. Zum anderen ist durch die Integration des BiCriterionAnt Algorithmus in das TCP der Rechenaufwand ein limitierender Faktor. Mit steigender Anzahl von Generationen und Ameisen ist ein erhöhter Rechenaufwand zu verzeichnen.

Eine Aussage über die Güte einer approximierten PF ist unabdingbar, um einen Vergleich von Algorithmen hinsichtlich der Problemeignung und Lösungsgüte durchführen zu können. Ferner kann

eine Bewertung der PF für die Analyse und den Einfluss der Heuristik-Parameterkonfiguration herangezogen werden. Hierbei ist vor allem die Güte hinsichtlich verschiedener Eigenschaften einer PF von Interesse, um eine Quantifizierung der Lösungsgüte zu ermitteln. Eine Quantifizierung ist in dieser Dissertation vor allem aus zwei Gründen notwendig. Zum einen wird ein Verfahren zur automatisierten Algorithmenkonfiguration durchgeführt, um eine geeignete Parameterkonfiguration für die BiCriterionAnt ACO Heuristik bestimmen zu können. Zum anderen wird die SPP Optimierung im Rahmen von einem autonomen Ansatz zur Bestimmung einer geeigneten Lösung des TCP genutzt. Dementsprechend müssen die Agenten autonom eine Bewertung der PFs durchführen, um eine Auswahl der geeignetsten Route ermitteln zu können.

In der Literatur werden verschiedene Metriken aufgeführt, um eine quantitative und qualitative Bewertung einer approximierten PF durchzuführen. Gemäß Deb [Deb01] können drei grundlegende Kategorien von Metriken unterschieden werden: Metriken der Konvergenz, Metriken der Diversität und Metriken, die beide Eigenschaften gleichzeitig abbilden. Des Weiteren wird in [JOZF14] eine vierte Kategorie vorgeschlagen, welche die Kapazität beziehungsweise Gehalt einer PF angeben kann. Eine detaillierte Beschreibungen von bestehenden Metriken ist dem Anhang A beigefügt. Zudem wird im Anhang A eine Einführung der Überlegenheitsrelation sowie die Eigenschaften der Kompatibilität und Vollständigkeit von Metriken angeführt.

In [ZTL+03] wird gezeigt, dass keine allgemeingültige unäre[7] Metrik existiert, die eine Aussage darüber treffen kann, ob eine Paretofront PF_A besser ist als eine Paretofront PF_B. Ferner existiert ebenfalls keine endliche Kombination von unären Metriken, die eine solche Aussage treffen kann. In der Regel können unäre Metriken eine Aussage darüber treffen, dass eine Paretofront PF_A gegenüber PF_B nicht schlechter oder unvergleichbar ist. Grundsätzlich wird eine Quantifizierung und somit eine Abbildung von qualitativen Eigenschaften einer PF auf $\mathbb{R}$ vorgenommen. Hingegen existieren binäre Metriken, die für ein Paar von PFs eine Aussage darüber ermöglichen, welche von beiden besser ist. Hierbei wird oftmals ein direkter Vergleich der Pareto-optimalen Lösungen vorgenommen. Der Unterschied beider PFs kann jedoch nicht quantifiziert werden derart, dass ein Vergleich mit weiteren PFs möglich ist [ZTL+03].

In dieser Dissertation wird eine Kombination von Metriken aus allen vier Kategorien genutzt, um die Güte einer PF messen zu können. Um die Integration in ein Verfahren der automatisierten Algorithmenkonfiguration zu ermöglichen, müssen verschiedene Eigenschaften einer PF analysiert werden. Die Lösungsgüte für eine konkrete Ausprägung einer Parameterkonfiguration für den BiCriterionAnt Algorithmus bestimmt maßgeblich die Suche nach besseren Parameterkonfigurationen. Für die Konstruktion einer Kombination von Metriken ist entscheidend, dass die Metriken einen Vergleich mehrerer PFs zulässt. Des Weiteren ist es von Vorteil, wenn die Metriken die Eigenschaft der Monotonie aufweisen und ferner ordnungserhaltende Abbildungen darstellen. Somit können eindeutige Rückschlüsse von der Lösungsgüte auf die Parameterkonfiguration gezogen und eine zielgerichtetere Optimierung der unterschiedlichen Parameter des BiCriterionAnt Algorithmus vorgenommen werden. Es werden nachfolgende unäre Metriken zur Bewertung der Lösungsgüte einer Parameterkonfiguration herangezogen. Unäre Metriken bieten den Vorteil, dass die Ergebnisse anschließend auf Grundlage einer Ordnungsrelation einen intuitiven Vergleich erlauben.

[7] Eine unäre Metrik evaluiert eine Menge A (PF) mit $M_u(A) : \mathfrak{A} \rightarrow \mathbb{R}$. Die Menge $\mathfrak{A}$ beinhaltet alle zu evaluierende Mengen.

Grundsätzlich ist es von Vorteil, wenn die PF eine hohe Anzahl von Pareto-optimalen Lösungen enthält. Der DM kann im Anschluss daran eine geeignete Lösung auswählen. Die ONVG[8]-Metrik [VL00] misst für eine Paretofront PF_A die Anzahl der Pareto-optimalen Lösungen und gehört der Kategorie der Kapazitätsmetriken an.

$$\text{ONVG}(PF_A) = |PF_A| \tag{4.30}$$

Die Metrik ist monoton, dass heißt eine Hinzunahme einer Pareto-optimalen Lösung führt in keinem Fall zu einer Verschlechterung. Allein genommen kann die Metrik jedoch zu missverständlichen Ergebnissen führen. Angenommen es gilt $\text{ONVG}(PF_A) = 1$ und PF_A dominiert alle Lösungen einer Paretofront PF_B und ist somit „besser“, obwohl $\text{ONVG}(PF_A) < \text{ONVG}(PF_B)$ gilt [KC02]. Resultierend bildet die Metrik nicht die Überlegenheit einer PF ab. Aus diesem Grund muss die ONVG-Metrik mit Metriken der Konvergenz und Diversität kombiniert werden, um eine umfassende Bewertung der Lösungsgüte vornehmen zu können.

Die Konvergenz, dass heißt die Annäherung einer PF an die wahre PF kann mit der $\mathfrak{S}$-Metrik nach [ZT98] bestimmt werden. Hierbei wird für jede Lösung innerhalb einer Paretofront PF_A das dominierte Hypervolumen bestimmt. Das Hypervolumen wird durch einen Referenzpunkt $\mathbf{r}$ beschränkt und definiert sich zu

$$\begin{aligned} \mathfrak{S}(PF_A, \mathbf{r}) &= \iota(\{\mathbf{u} \in \mathbb{R}^k | \exists f(\mathbf{x}) \in PF_A : f(\mathbf{x}) \preceq \mathbf{u} \preceq \mathbf{r}\}) \\ &= \iota(\bigcup_{i=1}^{|PF_A|} [f(\mathbf{x}_i), \mathbf{r}]) \end{aligned} \tag{4.31}$$

Es gilt für $\mathbb{R}^k$, dass $k \geq 2$ ist. Des Weiteren stellt ι das Lebesgue-Maß dar, das einem geometrischen Objekt eine Größe zuordnet. In dieser Dissertation werden für das MOSPP mit dem BiCriterionAnt Algorithmus zwei Zieldimensionen mit $k = 2$ optimiert. Mit $f(\mathbf{x}_i)$ werden die Zielfunktionswerte einer Lösung $\mathbf{x}_i \in PF_A$ abgebildet. Das Hypervolumen wird durch das zugehörige k-dimensionale Intervall $[f(\mathbf{x}_i), \mathbf{r}]$ beschrieben, welches den dominierten Raum von $\mathbf{x}_i$ darstellt und durch den Referenzpunkt $\mathbf{r}$ begrenzt wird. Ein Referenzpunkt $\mathbf{r}$ muss dabei im Fall des Minimierungsproblems für eine Menge A nachfolgende Bedingung erfüllen:

$$\exists r_k \text{ mit } r_i = \max_{\mathbf{r} \in A} r_i \tag{4.32}$$

Das heißt, dass der Referenzpunkt in einer Zielfunktion k einen schlechtesten Wert unter den Elementen in A aufweist [Beu11]. Des Weiteren werden sich überschneidende Hyperräume einzelner Lösungen $\mathbf{x}_i \in PF_A$ einfach gezählt. Mit ι wird das Lebesgue-Maß bezeichnet. Das Hypervolumen der approximierten Paretofront PF_A ist zudem durch die wahre PF des betrachteten Optimierungsproblem beschränkt. Resultierend existiert ein Maximalwert für die $\mathfrak{S}$-Metrik. Des Weiteren nimmt die Metrik eine Differenzierung bezüglich der Überlegenheitsrelation vor. Mit einem größeren Metrikwert ist eine Überlegenheit gegenüber anderen PFs mit kleineren Metrikwerten im Sinne der Konvergenz gewährleistet. Ferner liegt die Eigenschaft der Monotonie vor. Die $\mathfrak{S}$-Metrik stellt somit eine der fairsten Metriken im Forschungsfeld der MOO dar [ZTL+03, Beu06]. Zu beachten ist, dass die Ordnung der PFs entsprechend ihrer Lösungsgüte von der Lage des Referenzpunktes abhängig ist [KC02, YLW+07].

[8] Engl. Overall Non-dominated Vector Generation

Parameter x_d	Bezeichner in Auswertung	Beschreibung	Definition $[\underline{x_d}, \overline{x_d}]$
α	alpha	Gewichtungsfaktor (Pheromone)	[1,2]
β	beta	Gewichtungsfaktor (heuristische Information)	[1,2]
ϱ_{ANZ}	plb	Pheromonhinzugabe	[35,80]
ϱ_E	per	Pheromonevaporation	[0.25,0.6]
$\max_{Generation}$		# Generationen	25
$\lvert A \rvert$		# Ameisen	30
$\rho_{\min}$		minimale Pheromonmenge auf $\rho_{i,j}^K, \rho_{i,j}^{EWK}$	0
$\rho_{\max}$		maximale Pheromonmenge auf $\rho_{i,j}^K, \rho_{i,j}^{EWK}$	100
ρ_{ini}		initialisierte Pheromonmenge auf $\rho_{i,j}^K, \rho_{i,j}^{EWK}$	50

Tabelle 4.1.: IRace Algorithmus - Experimentaufbau und Angabe des Parameterraumes für das betrachtete SSPP. Innerhalb des definierten Parameterraumes wird eine Suche nach geeigneten Parameterkonfigurationen vorgenommen.

Der dominierte Hyperraum von einer Lösung $\mathbf{x}_i \in PF_A$ wird auch als $\mathfrak{S}$-Metrikbeitrag bezeichnet. Mit Hilfe der $\mathfrak{S}$-Metrikbeiträge kann die Diversität einer PF bewertet werden. Demnach wird die räumliche Verteilung der Lösungen entlang der PF untersucht. Über die $\mathfrak{S}$-Metrik wird mit dem Hypervolumen bereits indirekt die Diversität und Kapazität in die Bewertung einbezogen. Eine hohe Anzahl von Pareto-optimalen Lösungen wirkt sich positiv auf die Größe des Hypervolumens aus. Zudem deutet ein größeres Hypervolumen grundsätzlich auf eine gleichmäßigere Verteilung der Lösungen entlang der PF hin. Eine gleichmäßige Verteilung der Lösungen ermöglicht dem DM eine Auswahl aus möglichst unterschiedlichen Regionen des Lösungsraumes entsprechend der Problemstellung. Der $\mathfrak{S}$-Metrikbeitrag wird durch

$$\mathfrak{S}_{\text{Beitrag}}(f(\mathbf{x}), PF_A, \mathbf{r}) = \mathfrak{S}(PF_A, \mathbf{r}) - \mathfrak{S}(PF_A \backslash \{f(\mathbf{x})\}, \mathbf{r}) \tag{4.33}$$

definiert. Mittels der Standardabweichung über alle $\mathfrak{S}$-Metrikbeiträge kann die räumliche Verteilung zusätzlich gemessen werden. Demnach führt eine höhere Standardabweichung zu einer ungleichmäßigeren Verteilung.

Die $\mathfrak{S}$-Metrik weist neben den Vorteilen auch einen erheblichen Nachteil auf. Die Berechnung ist mit einer Komplexität von $O(|PF_A|^{Dim-1})$ insbesondere für höher-dimensionale Probleme rechenaufwendig [KC02].

4.3.3.2. Experimentaufbau

Der Experimentaufbau beschreibt die Durchführung der Offline Parameterkonfiguration mittels dem IRace Algorithmus. Resultierend wird für den in dieser Dissertation implementierten BiCriterionAnt Algorithmus die Suche nach einer geeigneten Heuristik-Parameterkonfiguration durchgeführt. Es werden für die Parameter des BiCriterionAnt Algorithmus die in Tabelle 4.1 dargestellten Definitionsbereiche untersucht. Für die Anzahl der Ameisen, die maximale Anzahl von Generationen sowie die Initialisierung der Pheromonmatrizen $\mathcal{P}^K$ und $\mathcal{P}^{EWK}$ ist jeweils ein vordefinierter Wert gegeben.

Der IRace Algorithmus wird auf einer Menge von Probleminstanzen durchgeführt. Dazu werden 101 Probleminstanzen $\psi_i \in \Psi$ mit $i = [0, 100]$ zufällig generiert. Eine Probleminstanz repräsentiert

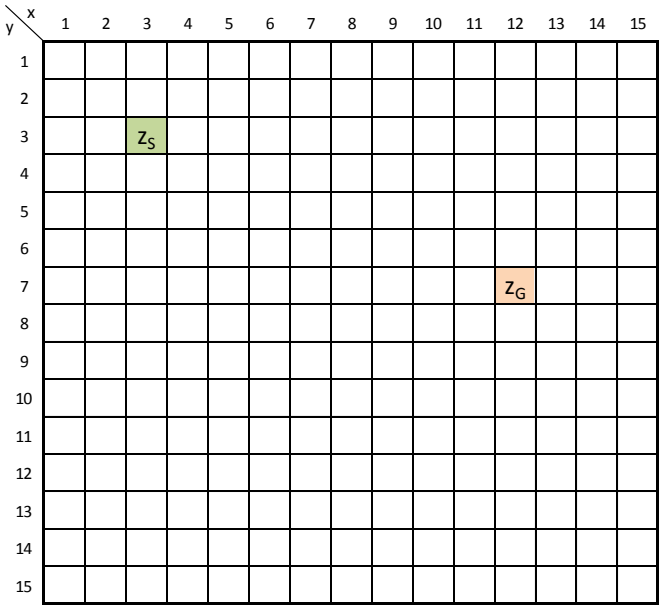


Abbildung 4.7.: Testumgebung zur experimentellen Untersuchung des transformierten Single Shortest Path Problem unter Unsicherheiten.

hierbei eine abstrahierte Systemumgebung Ω' mit der Ausdehnung von 15 mal 15 Zellen. Des Weiteren wird auf den Zellen die Ausprägung der Strömungskosten $\mathbf{d}_R(z_i)$ definiert. Es gilt $\psi_i \in \Psi$ mit i gleich dem prozentualen Anteil von bekannten Zellen in Ω'. Entsprechend ist für ψ_0 die Systemumgebung vollständig unbekannt und für ψ_{100} vollständig bekannt. Die unbekannten Zellen sind über die gesamte Systemumgebung gleichverteilt. Die konkrete Ausprägung der Strömungskosten $\mathbf{d}_R(z_i)$ auf jeder Zelle ist ebenfalls gleichverteilt. In der Systemumgebung sind keine Hindernisse platziert. Gemäß Abbildung 4.7 wird für jede Probleminstanz ein zulässiger Pfad $p_i \in P$ zwischen den Zellen $z_S = [3, 3]$ und $z_G = [12, 7]$ gesucht.

Die Implementierung des IRace Algorithmus basiert auf der IRace Implementierung in der Programmiersprache R von López-Ibáñez et al. [LIDLSB11]. Die variablen Komponenten sind innerhalb der Experimentierung entsprechend der R-Implementierung wie nachfolgend angegeben definiert.

- **Bewertungsmaß/Zielfunktion:**
  Die Bewertungsfunktion beziehungsweise Zielfunktion $f : K \times \Psi \to \mathbb{R}$ anhand derer die Lösungsgüte einer konkreten Parameterkonfiguration $k_i \in K$ für das MOSPP analysiert wird, ist eine Kombination der ONVG-Metrik [VL00] sowie der $\mathfrak{S}$-Metrik und indirekt der $\mathfrak{S}$-Metrikbeiträge [ZT98]. Die zusammengesetzte Zielfunktion wird wie folgt angegeben:

  $$\max ZF = \frac{\text{Hypervolumen}}{50} + \frac{\text{\# Lösungen}}{30} \tag{4.34}$$

  Das Hypervolumen wird mit einem Referenzpunkt $\mathbf{r} = (50, 0)$ bestimmt. Der Referenzpunkt erfüllt die Bedingung 4.32 für Lösungen die gute Approximationen der wahren PF darstellen. Für ein Hypervolumen kleiner gleich null wird das Hypervolumen auf eins gesetzt. Die Gewichtung der Einflussgrößen ist über ein Vorexperiment ermittelt worden.

Entsprechend werden Parameterkonfigurationen gesucht, die auf den untersuchten Probleminstanzen die besten PFs liefern. Das heißt, PFs die eine hohe Anzahl von Pareto-optimalen Lösungen sowie eine gute Annäherung an die wahre PF und eine gleichmäßige Verteilung der Pareto-optimalen Lösungen entlang der approximierten PF aufweisen. Hierbei wird der Mittelwert der Zielfunktionswerte über alle Instanzen bestimmt. Der IRace Algorithmus führt eine Minimierung der Zielfunktionswerte durch, aus diesem Grund wird die ZF mit -1 multipliziert.

- **Budget und Anzahl der Iterationen:**
  Das Budget bestimmt maßgeblich die Anzahl der Iterationen und somit auch die Anzahl der in dem IRace Algorithmus durchzuführenden Einzelexperimente einer Iteration. Die Anzahl der Iterationen ist durch die Abrundungsfunktion

$$\#\text{ Iterationen} = \lfloor 2 + \log_2 \#\text{Parameter} \rfloor \tag{4.35}$$

  definiert. In dem IRace Experiment in dieser Dissertation werden vier Parameter untersucht und ein Budget von 1000 vorgegeben. Die Größenordnung des Budget orientiert sich an dem Vorschlag aus dem IRace-Framework [LIDLSB11]. Resultierend werden 1000 Einzelexperimente verteilt auf vier Iterationen durchgeführt. Die Anzahl der Einzelexperimente wird für jede Iteration wie folgt definiert:

$$\text{Budget}_j = \frac{\text{Budget}_{\text{max}} - \text{Budget}_{\text{verbraucht}}}{\#\text{ Iterationen} - j + 1} \tag{4.36}$$

  mit $j = 1, \ldots, \#\text{Iterationen}$

- **Auswahl von Probleminstanzen:**
  Die Sequenz der abzuarbeitenden Probleminstanzen wird zufällig ausgewählt. Die Wahrscheinlichkeiten, dass eine Probleminstanz gezogen wird, ist über alle Probleminstanzen gleichverteilt. In dieser Dissertation liegt für den IRace Algorithmus die nachfolgende Sequenz der untersuchten Probleminstanzen, zusammengesetzt über alle vier Iterationen, vor.

$$\begin{aligned}
\Psi_{\text{Exp}}(\text{Iteration} = 1) &= (38, 3, 25, 82, 96, 29) \\
\Psi_{\text{Exp}}(\text{Iteration} = 2) &= (17, 27, 24, 4, 20, 54, 77) \\
\Psi_{\text{Exp}}(\text{Iteration} = 3) &= (1, 72, 15, 27, 74, 34, 73, 51) \\
\Psi_{\text{Exp}}(\text{Iteration} = 4) &= (18, 34, 70, 32, 94, 68, 92, 30, 79)
\end{aligned} \tag{4.37}$$

- **Ermittlung neuer Parameterkonfigurationen:**
  Die Anzahl der zu überprüfenden Parameterkonfigurationen in jeder Iteration j ist wie folgt definiert:

$$|K_j| = \left\lfloor \frac{\text{Budget}_j}{\gamma + \min(5, j)} \right\rfloor \tag{4.38}$$

  Gemäß der Definition nimmt die Anzahl der Parameterkonfigurationen mit zunehmenden Iterationen ab. Resultierend können mehr Evaluationen der Parameterkonfigurationen anhand der Probleminstanzen durchgeführt werden. Mittels γ wird das Verhältnis zwischen dem zur Verfügung stehenden Budget und der Anzahl der Parameterkonfigurationen angegeben. In dieser Dissertation gilt gemäß dem IRace-Framework $\gamma = 5$ [LIDLSB11].

Alle Parameterkonfigurationen die nicht durch den statistischen Test aus K entfernt worden sind, werden entsprechend der gemittelten Zielfunktionswerte über alle Probleminstanzen gerankt r_{k_i}. Die Parameterkonfigurationen werden aufsteigend gerankt, so dass der besten Parameterkonfiguration der niedrigste Rank $r_{k*} = 1$ zugeordnet wird. Für neue Parameterkonfiguration wird zuerst eine Elternkonfiguration aus $k_i \in K^{\text{elite}}$ mit einer Wahrscheinlichkeit

$$p_{k_i} = \frac{|K_{j-1}^{\text{elite}}| - r_{k_i} + 1}{|K_{j-1}^{\text{elite}}| \frac{|K_{j-1}^{\text{elite}}|+1}{2}} \tag{4.39}$$

ausgewählt. Resultierend werden bessere Parameterkonfigurationen mit einer höheren Wahrscheinlichkeit ausgewählt. Im nächsten Schritt werden die Parameter der neuen Parameterkonfiguration anhand der Elternkonfiguration ermittelt. Hierbei werden die numerischen Parameter $[\underline{x_d}, \overline{x_d}]$ anhand der trunkierten Normalverteilung $\mathcal{N}(\mu_d^{k_i^{\text{elite}}}, \sigma_d^j)$ bestimmt. Hierbei stellt der Mittelwert $\mu_d^{k_i^{\text{elite}}}$ den konkreten Parameterwert der Elternkonfiguration dar. Die Standardabweichung σ_d^j wird in der ersten Iteration $j = 1$ mit

$$\sigma_d^j = \frac{\underline{x_d} - \overline{x_d}}{2} \tag{4.40}$$

vordefiniert. Im Verlauf der Iterationen wird σ_d^j mit

$$\sigma_d^j = \sigma_d^{j-1} \left(\frac{1}{|K_j^{\text{neu}}|} \right)^{\frac{1}{\#\text{Parameter}}} \tag{4.41}$$

angepasst. Resultierend liegen die aus der trunkierten Normalverteilung gezogenen Parameter mit zunehmenden Iterationen näher um die Parameterwerte der Elternkonfiguration [LIDLSB11].

4.3.3.3. Analyse und Auswertung

Nachfolgend werden die Ergebnisse der durchgeführten Offline Parameterkonfiguration für den BiCriterionAnt Algorithmus, gemäß des zuvor beschriebenen Experimentaufbaus, zusammengefasst. Grundsätzlich wird das Ziel verfolgt, eine Parameterkonfiguration für den BiCriterionAnt Algorithmus zu ermitteln, die für eine Vielzahl von verschiedenen Probleminstanzen zu einer hohen Lösungsgüte führt. Die Lösungsgüte einer Parameterkonfiguration auf einer konkreten Probleminstanz wird mit der Zielfunktion 4.34 beschrieben. Die Parameter innerhalb der untersuchten Parameterkonfiguration stehen in Abhängigkeit zueinander. Hierbei bilden vor allem die Parameterpaare (*alpha,beta*) und (*plb,per*) eine stärkere Abhängigkeit. Mit dem Parameterpaar (*alpha,beta*) wird das Verhältnis von Pheromonen und der zugehörigen heuristischen Information angegeben, das bei der Auswahl von Lösungskomponenten berücksichtigt wird. Eine höhere Gewichtung der Pheromone (*alpha*) führt tendenziell dazu, dass im Verlauf der Generationen eher in der Nachbarschaft von guten Lösungen gesucht wird. Hingegen führt tendenziell eine höhere Gewichtung der heuristischen Information (beta) dazu, dass bisher noch nicht berücksichtigte gute Lösungskomponenten untersucht werden. Für das Parameterpaar (*plb,per*) ist eine Abhängigkeit derart gegeben, dass durch die Pheromonhinzugabe (*plb*) gute Lösungen einer Generation proportional stärker berücksichtigt werden in den nachfolgenden Generationen. Die Pheromonevaporation (*per*) hingegen mindert den Einfluss guter Lösungen auf die Berücksichtigung in nachfolgenden Generationen. Resultierend

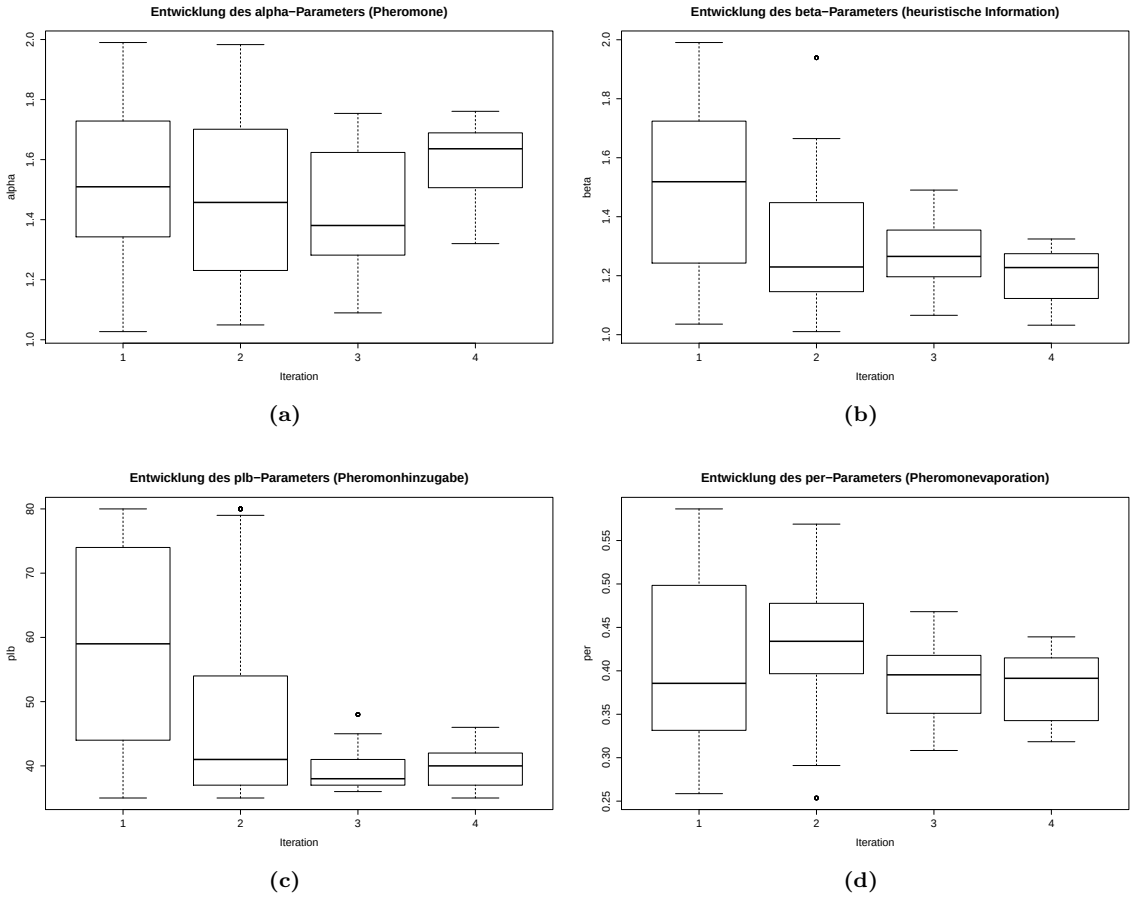


Abbildung 4.8.: IRace - Entwicklung der Parameterausprägungen im Rahmen der Suche nach geeigneten Parameterkonfigurationen für das betrachtete SSPP. Unter (a) *alpha*-Parameter, (b) *beta*-Parameter, (c) *plb*-Parameter und (d) *per*-Parameter.

wird vor allem für vermeintlich gute Lösungen der Einfluss reduziert und die Möglichkeit erhöht ein lokales Optimum überwinden zu können.

Entsprechend dem IRace Algorithmus kann festgestellt werden, dass sich die Definitionsbereiche $[\underline{x_d}, \overline{x_d}]$ der untersuchten Parameter über die Iterationen hinweg immer stärker eingrenzen. Aufgrund der Bestimmung neuer Parameterkonfigurationen mittels trunkierter Normalverteilungen um die Parameter guter Parameterkonfigurationen aus der zuvor durchgeführten Iteration, wird die Suche nach neuen Parameterkonfigurationen gesteuert. In der ersten Iteration wird anhand von sechs Probleminstanzen $\Psi_{\text{Exp}}(\text{Iteration} = 1)$ entsprechend (4.37) der gesamte Definitionsbereich $[\underline{x_d}, \overline{x_d}]$ auf die Parameter x_d überprüft. Für *alpha* sowie *per* und *plb* führen Parameterausprägungen über den gesamten Definitionsbereich zu guten Lösungen. Erst nach der zweiten Iteration grenzt sich der Definitionsbereich hinreichend stark ein. Definitionsbereiche, die mit repräsentativen Parameterkonfigurationen auf den bisher 13 untersuchten Probleminstanzen durchschnittlich zu weniger gute Lösungen geführt haben, werden verworfen. Besonders gut kann der Definitionsbereich für *beta* und *plb* eingegrenzt werden. Der Verlauf der Iterationen und die Suche nach guten Parameterkonfigurationen kann der Abbildung 4.8 entnommen werden.

Des Weiteren ist mit der Abbildung 4.9 der Zusammenhang der Parameterpaare (*alpha*,*beta*) sowie (*plb*,*per*) veranschaulicht. Es ist zu erkennen, dass tendenziell eine höhere Gewichtung des

Konfiguration	alpha	beta	plb	per
101	1.5542	1.2500	36	0.3361
112	1.5065	1.3086	40	0.3400
116	1.6560	1.1049	41	0.4155
110	1.5281	1.2346	38	0.3405

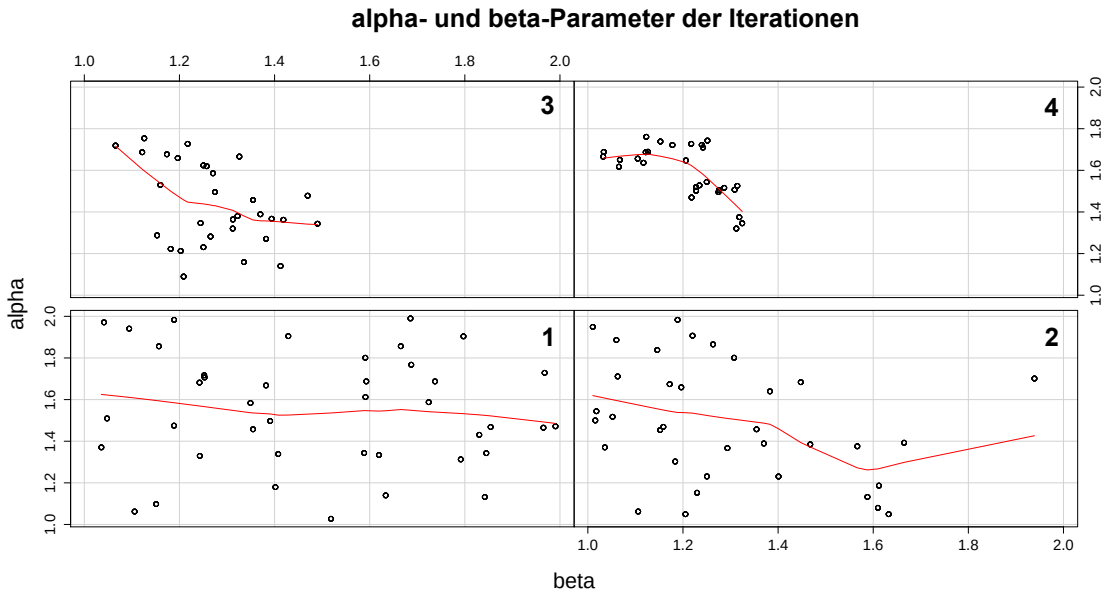
Tabelle 4.2.: IRace Algorithmus - Parameterkonfigurationen, die für die untersuchten Probleminstanzen für das betrachtete SSPP ermittelt worden sind.

alpha-Parameters und niedrigere Gewichtung des beta-Parameters zu geeigneten Lösungen des betrachteten SSPP führen. Entsprechend werden, bei der Bestimmung der Auswahlwahrscheinlichkeit von Lösungskomponenten gemäß der implementierten Gleichung 4.27 im BiCriterionAnt Algorithmus, eher die jeweiligen Pheromonausprägungen anstatt die heuristischen Informationen auf den Lösungskomponenten herangezogen. Resultierend berücksichtigen die Ameisen im BiCriterionAnt Algorithmus stets gute Lösungen in den nachfolgenden Generationen. Aufgrund dem geringen Unterschied zwischen den Gewichtungen (*alpha*,*beta*) besteht zugleich die Möglichkeit, dass die Ameisen eine Suche in der Nachbarschaft von bereits guten Lösungen fortsetzen. Bei der Gegenüberstellung der *plb* und *per* Parameterausprägungen ist festzustellen, dass mit einer Zunahme der abgelegten Pheromonen auf guten Lösungen zugleich eine Zunahme der Pheromonevaporation einhergeht. Die Beobachtung kann dadurch begründet werden, dass der Einfluss guter Lösungen kontinuierlich abnehmen soll. Zum einen können dadurch vermeintlich gute Lösungen, die unter Umständen nur ein lokales Optimum darstellen, überwunden werden. Resultierend können neue Lösungen schneller gefunden werden, weil die Ameisen tendenziell ganz neue Lösungen generieren beziehungsweise die Suche stärker auf Nachbarlösungen konzentrieren.

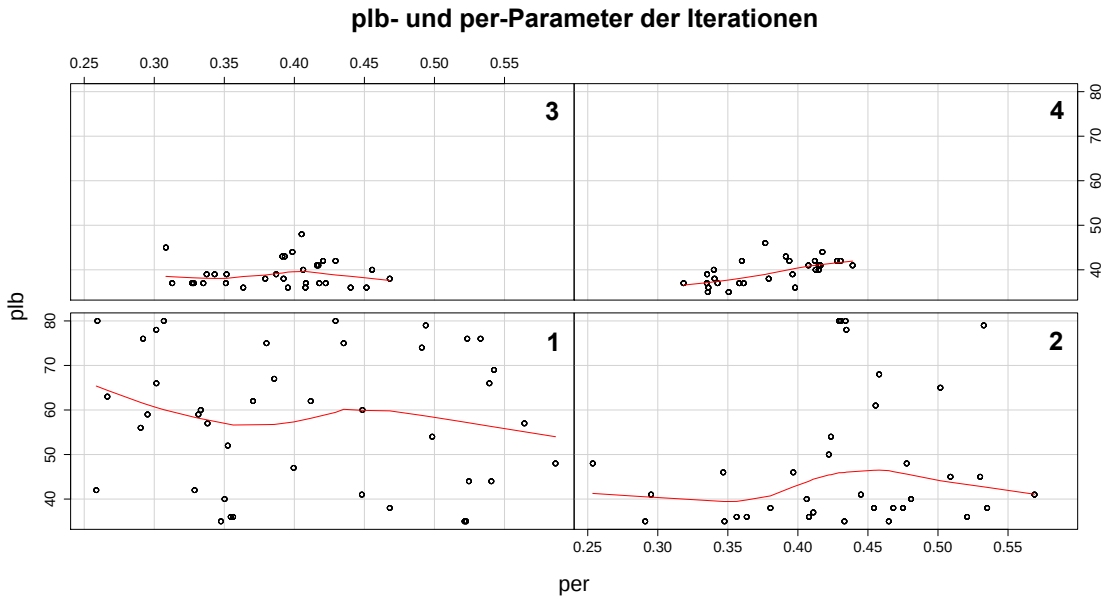
Nach vier Iterationen sind die vier geeignetsten Parameterkonfigurationen gemäß Tabelle 4.2 ermittelt worden. Die ermittelten Parameterkonfigurationen stellen hierbei besonders robuste Parameterkonfigurationen dar, die auf den untersuchten Probleminstanzen eine gute Performance gezeigt haben. Auf der Grundlage der in Tabelle 4.2 wird die beste Parameterkonfiguration mit der ID 101, für das in dieser Dissertation untersuchte SSPP unter der Annahme, dass 25 Generationen, 30 Ameisen, eine minimale Pheromonmenge von null sowie eine maximale Pheromonmenge von 100 und eine Startpheromonmenge von 50 auf $\rho_{i,j}^{K}, \rho_{i,j}^{EWK}$ initialisiert werden, wie folgt angegeben:

$$\begin{aligned}\text{Parameterkonfiguration} = (&\max_{Generation} = 25; |A| = 30; \alpha = 1.55; \beta = 1.25;\\ &\varrho_{ANZ} = 36; \varrho_E = 0.34; \rho_{\min} = 0; \rho_{\max} = 100; \rho_{\text{ini}} = 50)\end{aligned}$$

Für die angegebene Parameterkonfiguration ist ein weiteres Experiment durchgeführt worden. Der Experimentaufbau entspricht dem Experimentaufbau im Rahmen des IRace Experimentes aus Kapitel 4.3.3.2. Die Parameterkonfiguration des BiCriterionAnt Algorithmus ist hierbei auf sechs verschiedenen Probleminstanzen jeweils mit einem prozentualen Anteil bekannter Zellen von (0,20,40,60,80,100) überprüft worden, um die Lösungsgüte untersuchen zu können. Es sind jeweils 30 Wiederholungen auf jeder Probleminstanz vorgenommen worden. Für jede Probleminstanz ist die konkrete Ausprägung der Strömungsrichtung $\mathbf{d}_R(z_i)$ auf jeder Zelle über alle Wiederholungen unverändert. In der Abbildung 4.10 ist die Lösungsgüte entsprechend dem Hypervolumen, der Anzahl Pareto-optimaler Lösungen und der Standardabweichung über die $\mathfrak{S}$-Metrikbeiträge veranschaulicht.



(a)

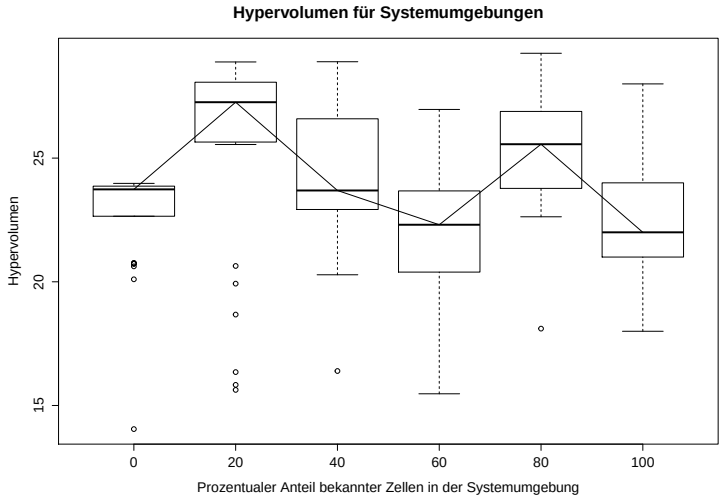


(b)

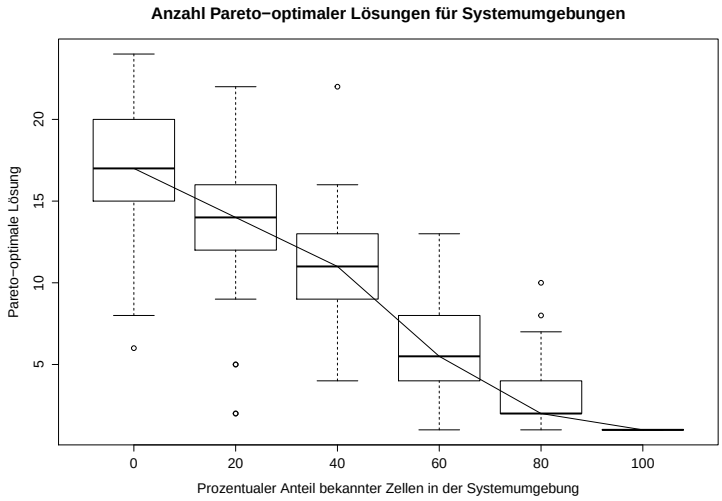
Abbildung 4.9.: IRace - Paarweise Gegenüberstellung der Parameterausprägungen für das betrachtete SSPP für die vier Iterationen. Unter (a) Parameterpaar (*alpha*,*beta*) und (b) Parameterpaar (*plb*,*per*).

Die Untersuchung zeigt, dass die Mittelwerte der Hypervolumina, der entsprechenden Pareto-optimalen Lösungen über die betrachteten Probleminstanzen, zwischen 21.95 und 25.46 schwanken. Es kann vermutet werden, dass der BiCriterionAnt Algorithmus, aufgrund der heterogenen Ameisen und resultierend aus der unterschiedlichen Gewichtungen der Zielfunktionen innerhalb einer Ameisenkolonie, gute Lösungen auf Probleminstanzen bestimmt, die sich hinsichtlich der Struktur bekannter und unbekannter Zellen unterscheiden. Auf einer vollständig unbekannten Systemumgebung muss stets eine Routenplanung unter Berücksichtigung der Eintrittswahrscheinlichkeit und Kosten durchgeführt werden. Hingegen muss der BiCriterionAnt Algorithmus ebenfalls auf vollständig bekannten Systemumgebungen gute Lösungen bestimmen können. Entsprechend liegt für jede Route in einer vollständig bekannten Systemumgebung eine Eintrittswahrscheinlichkeit von eins vor. Ferner kann mit Ausnahme indifferenter Routen nur eine Pareto-optimale Lösung existieren, nämlich die kürzeste Route. Im Mittel nimmt die Anzahl der Pareto-optimalen Lösungen in der Untersuchung stetig ab. Hierbei werden im Mittel 17.33 Pareto-optimale Lösungen auf vollständig unbekannten Systemumgebungen gefunden. Auf vollständig bekannten Systemumgebungen wird hingegen jeweils nur eine Pareto-optimale Lösung, die kürzeste Route, gefunden. Des Weiteren nimmt die Standardabweichung über die Anzahl Pareto-optimaler Lösungen mit steigender Anzahl bekannter Zellen in der Systemumgebung kontinuierlich ab. Die Beobachtung lässt sich ebenfalls über die abnehmende Anzahl konkreter Ausprägungen einer Route begründen. Je mehr unbekannte Zellen auf einer Route liegen, desto mehr Kombinationsmöglichkeiten liegen bezüglich des Eintritts konkreter Kosten auf den Zellen vor, die von den Ameisen untersucht werden. Resultierend kann eine Route zu verschiedenen Realisierungen unter Angabe der Kosten und zugehörigen Eintrittswahrscheinlichkeit beziehungsweise Pareto-optimalen Lösungen führen. Insofern ist das tendenziell gleichbleibende Hypervolumen auf den betrachteten Probleminstanzen positiv zu bewerten, weil sich eine niedrige Anzahl Pareto-optimaler Lösungen negativ auf die Ausprägung des Hypervolumens auswirkt. Weiterhin ist festzustellen, dass mit steigender Anzahl bekannter Zellen in der Systemumgebung die Standardabweichung über alle $\mathfrak{S}$-Metrikbeiträge der Pareto-optimalen Lösungen tendenziell zunimmt. Grundsätzlich ist zu beobachten, dass die Pareto-optimalen Lösungen für Probleminstanzen mit einem Anteil von bis zu 60 % unbekannter Zellen eine sehr geringe Standardabweichung aufweisen. Resultierend liegen die Pareto-optimalen Lösungen nahezu gleichmäßig verteilt entlang der Paretofront. Eine ungleichmäßigere Verteilung der Pareto-optimalen Lösungen ist für Probleminstanzen mit einem hohen Anteil bekannter Zellen zu beobachten. Für die vollständig bekannte Systemumgebung in der nur eine Pareto-optimale Lösung gefunden worden ist, entfällt die Standardabweichung über die $\mathfrak{S}$-Metrikbeiträge.

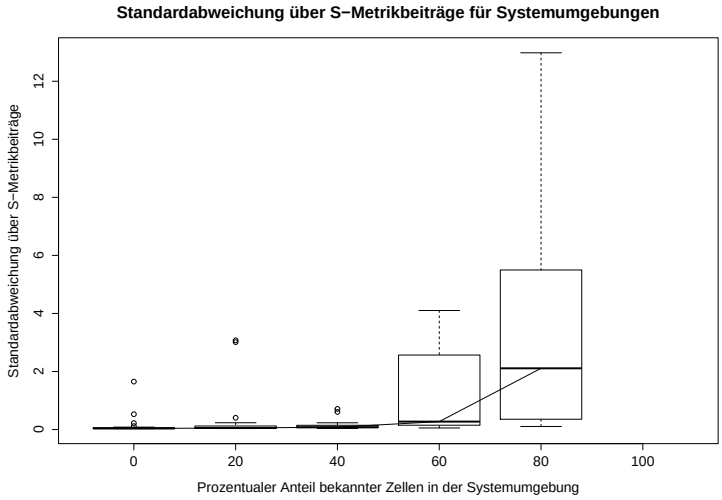
In der Abbildung 4.11 ist der Lösungsraum für eine vollständig unbekannten Systemumgebung angegeben. Auf jeder Zelle sind die geschätzten Strömungsinformationen $\mathbf{d}_R(z_i)$ bekannt. Die Strömungsinformationen sind zufällig auf der Grundlage einer Gleichverteilung ermitteln worden. Es wird eine Route zwischen $z_S = [3, 3]$ und $z_G = [12, 7]$ gesucht. Die längste Route besteht aus 162 Zellen und die kürzeste Route aus neun Zellen. Die kürzeste Route, bestehend aus neun Zellen, entspricht zugleich auch der absolut, robusten Route gemäß der Definition 4.3. Für das Experiment wird die zuvor ermittelte Parameterkonfiguration genutzt. Die Anzahl der Generationen wird auf 100 und die Anzahl der Ameisen auf 50 erhöht. Die Farbkodierung der Lösungselemente ist für beide Abbildungen in 4.11 a) und b) jeweils identisch und bezieht sich auf Routen. Mit einer Route wird die Reihenfolge der Zellen beschrieben. Ferner werden konkrete Realisierungen einer Route unter Angabe der Kosten und Eintrittswahrscheinlichkeit der zugehörigen Route farblich zugeordnet. Mit der Ähnlichkeit der Farbe kann kein Rückschluss auf die räumliche Nähe von zwei Routen



(a)



(b)



(c)

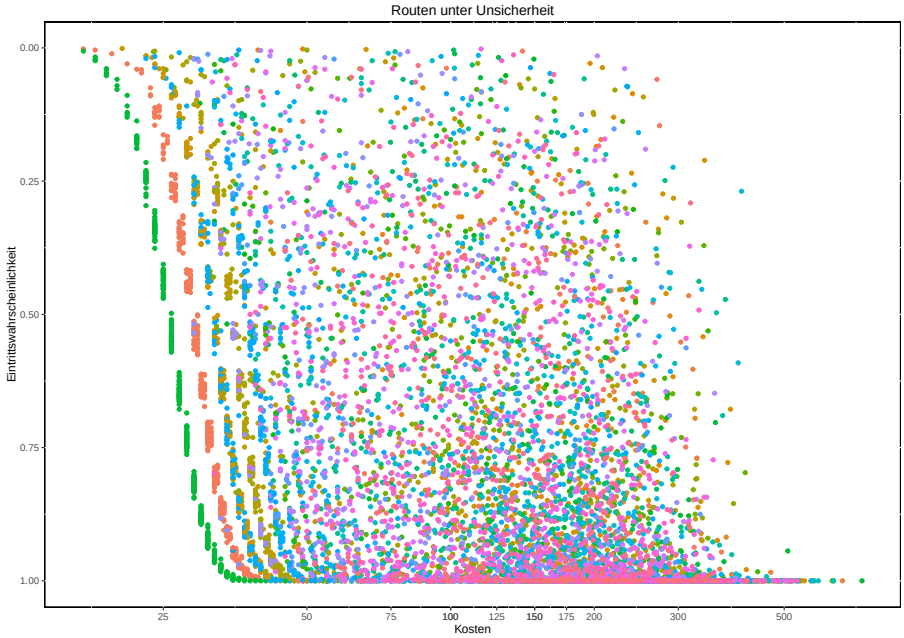
Abbildung 4.10.: BiCriterionAnt - Lösungsgüte der vier Iterationen auf verschiedenen Systemumgebungen für das betrachtete SSPP. Unter (a) Hypervolumen, (b) Anzahl Pareto-optimaler Lösungen und (c) Standardabweichung über die $\mathfrak{S}$-Metrikbeiträge.

vorgenommen werden. Die Farbkodierung veranschaulicht lediglich den Zeitpunkt, wann eine Route das erste Mal bestimmt worden ist. Es ist zu erkennen, dass nach einer Startphase die Ameisen zuerst voneinander unterschiedliche Routen erkunden. Die dislozierte Suche wird durch die heterogene Beschreibung der Ameisen und der daraus unterschiedlichen Gewichtung der Zielfunktionen unterstützt. Nach neun Generationen ist eine erste gute Route (Orange) identifiziert worden. Aufgrund der zusätzlichen Verstärkung der Attraktivität durch die Hinzugabe von Pheromonen, wird die Route detaillierter untersucht. Resultierend werden konkrete Realisierungen ermittelt. Parallel dazu werden zwei weitere Routen (Braun, Khaki) ermittelt und weitergehend untersucht. In der 58. Generation wird eine geeignetere Route (grün) identifiziert und die detaillierte Untersuchung der Routen Orange und Braun wird weitgehend eingestellt. Im weiteren Verlauf werden zusätzlich noch gute Routen (Blau, Lila, Magenta) ermittelt und untersucht. Insgesamt sind 8771 zulässige Ausprägungen von 3458 unterschiedlichen Routen ermittelt worden. Es ist zu erkennen, dass große Teilräume des gesamten Lösungsraumes, im Sinne der Exploration, untersucht worden sind. Die Untersuchung und Exploitation geeigneter Lösungen ist dabei auf ausgewählte Routen konzentriert worden. Des Weiteren ist es den Ameisen möglich gewesen, die Exploitation zwischen unterschiedlichen Routen zu wechseln. Die Möglichkeit besteht nur, wenn zum einen die Pheromonhinzugabe ϱ_{ANZ} und Pheromonevaporation ϱ_E für die Probleminstanz geeignet definiert ist. Dadurch kann die Pheromonkonzentration auf Lösungskomponenten (Zellen), die im Verlauf der Suche durch geeignetere ersetzt worden sind, kontinuierlich abgebaut werden. Zum anderen muss ein geeignetes Verhältnis zwischen den Gewichtungsparametern α und β vorliegen. Resultierend ist es einigen Ameisen möglich, von Lösungskomponenten (Zellen) mit hoher Pheromonkonzentration abweichen zu können, um unter Umständen dominierende Lösungen auffinden zu können.

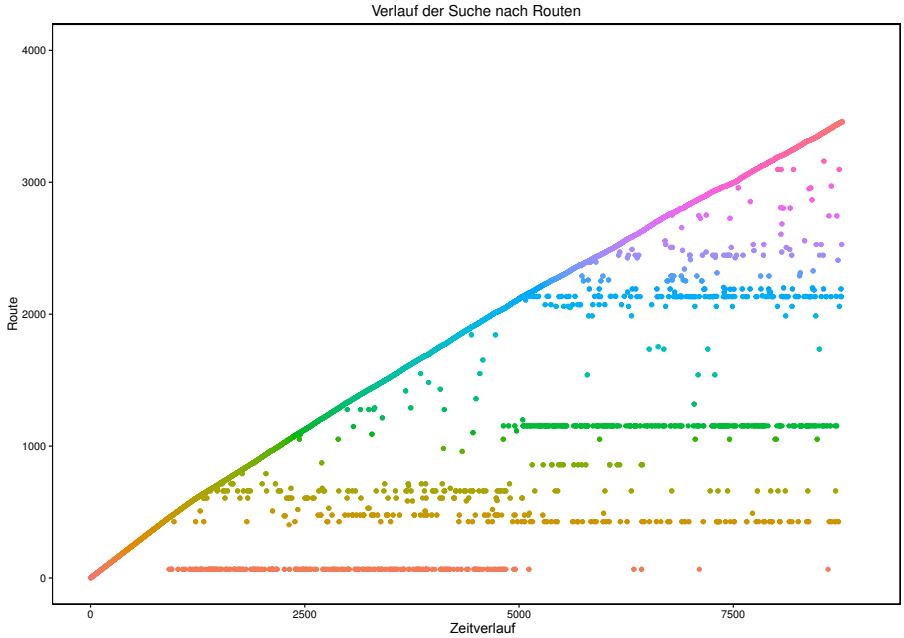
4.4. Zusammenfassung

Die Untersuchungen zeigen, dass eine Transformation des SSPP in das MOSPP vorgenommen werden kann. Hierbei werden die Unsicherheiten, beziehungsweise die Eintrittswahrscheinlichkeiten als zusätzliche Zielfunktion formuliert. Womöglich kann die Transformation auf eine Vielzahl von ähnlichen Problemstellungen unter Unsicherheiten übertragen werden. Die Vorteile liegen vor allem in der besseren Vergleichbarkeit von Lösungen. Im Sinne der MOO wird eine Ordnungsrelation auf den Lösungen eingeführt. Ferner wird eine quantitative Aussage über die Robustheit von Lösungen ermöglicht. Die Ursache hierfür liegt in der Modellierung begründet. In der Literatur werden die Unsicherheiten in den Nebenbedingungen formuliert. Eine Optimierung erfolgt anhand der übergeordneten Zielfunktion. Entsprechend wird eine optimierte, robuste Lösung bestimmt, die über alle möglichen Szenarien zu guten Lösungen führt.

Für das MOSPP ist eine geeignete Modellierung der Systemumgebung erstellt worden. Auf den Lösungskomponenten (Zellen) sind die Strömungsinformationen in Form eines Vektors repräsentiert. Für jede der acht möglichen Strömungsrichtungen existiert eine Eintrittswahrscheinlichkeit. Zur Lösung des MOSPP ist die BiCriterionAnt Heuristik genutzt worden. Die Heuristik ist auf das konkrete Problem angepasst worden. Es konnte gezeigt werden, dass eine gute Exploration des Lösungsraumes möglich ist. Besonders geeignete Lösungen (dominierende Lösungen) werden weitergehend, im Sinne der Exploitation, untersucht. Die Einführung einer Ordnungsrelation ermöglicht die Vergleichbarkeit der ermittelten Routen und deren konkreten Ausprägungen.



(a)



(b)

Abbildung 4.11.: BiCriterionAnt - Veranschaulichung des Lösungsraumes für eine Beispielinstanz des SSPP. Unter (a) Lösungsraum mit konkreten Realisierungen auf verschiedenen Routen, (b) Zeitverlauf der Suche nach Routen.

Untersuchungen in der Literatur zeigen, dass die Lösungsgüte von Heuristiken in Abhängigkeit zur Definition der Heuristik-Parameterkonfiguration steht. Um das MOSPP im weiteren Verlauf der Dissertation für das Routenplanungsproblem innerhalb des TCP anwenden zu können, ist eine Offline Parameterkonfiguration durchgeführt worden. Für das TCP ist es nicht möglich, eine konkrete Probleminstanz anzugeben. Die Systemumgebung unterliegt fortlaufend Änderungen. Neue Informationen über Strömungskosten und Hindernisausprägungen werden gesammelt. Resultierend existiert eine zuvor unbekannte Menge von möglichen Probleminstanzen. Der IRace Algorithmus ermittelt eine robuste Heuristik-Parameterkonfiguration, die für eine Vielzahl von möglichen Probleminstanzen eine hohe Lösungsgüte erzielen kann. Es ist eine Menge von repräsentativen Probleminstanzen erstellt worden, die dem IRace Algorithmus als Grundlage dient. Die Lösungsgüte der ermittelten Heuristik-Parameterkonfiguration wird anhand des Hypervolumens, der $\mathfrak{S}$-Metrik und der ONVG-Metrik überprüft.

Zusammenfassend sind die Untersuchungen im Sinne einer Machbarkeitsstudie erfolgreich durchgeführt worden. Ferner sind die Grundlagen zur Implementierung des MOSPP in das TCP gelegt worden. Im nachfolgenden Kapitel 5 der Dissertation werden die Auswirkungen der Routenplanung unter Unsicherheiten auf das TCP untersucht.

5. Ein Terrain Covering Modell unter Unsicherheiten

Den Vorteil hat der Weg zum Wahren,
daß man sich auch unsicherer Schritte,
ja eines Umweges oder Fehltrittes sogar
immer gerne erinnert.
Johann Wolfgang von Goethe (1749–1832)

In diesem Kapitel wird ein neues Terrain Covering Modell beschrieben, das eine robuste Abdeckung einer unbekannten Systemumgebung ermöglicht. Hierbei werden, aufbauend auf den Ansätzen und resultierenden Erkenntnissen bestehender TCMs aus Kapitel 2, das Task Allocation Problem und erstmals das Stochastic Shortest Path Problem (Kapitel 4) für das Terrain Covering Problem untersucht. Auf der Grundlage der aktuellen räumlichen Positionen der Agenten in der Systemumgebung sowie der bereits übernommenen Aufgaben wird eine optimierte Zuweisung der noch zu bearbeitenden Aufgaben vorgenommen. Die Koordination der Agenten wird mittels einer hybriden Organisationsform sichergestellt. Es werden die Vorteile einer zentralen und dezentralen Koordination in einem auktionsbasierten Algorithmus implementiert. Das Gebot einer Auktion wird durch eine Route repräsentiert. Vor allem für das Online TCP bestehen Unsicherheiten über die Ausprägung der unbekannten Regionen einer Systemumgebung, die im Verlauf der Abdeckung befahren werden. Mittels der Routenplanung unter Unsicherheiten kann eine Approximation der resultierenden Kosten einer Befahrung ermittelt und für die Aufgabenplanung berücksichtigt werden.

In diesem Kapitel wird das SMO[1]-TCM eingeführt und im Rahmen einer Machbarkeitsstudie untersucht, um die positiven Auswirkungen auf die Effizienz und Robustheit einer vollständigen Abdeckung aufzeigen zu können. Hierbei wird das SMO-TCM anhand von verschiedenen Umgebungsstrukturen und Einflussfaktoren analysiert.

5.1. Das Stochastic Multiobjective-Terrain Covering Modell - Ein onlinebasiertes Multi-Robot Terrain Covering Modell unter Unsicherheiten

Mit dem SMO-TCM wird das Ziel verfolgt, eine Simulation unter realitätsnahen Annahmen durchführen zu können. Die Taxonomie bestehender TCMs für ein MA(R)S hat aufgezeigt, dass nur wenige TCMs [SYTX06, HAS12] eine Untersuchung von mehreren Zielfunktionen ermöglichen.

[1] Engl. Stochastic Multiobjective

Ebenfalls untersuchen nur wenige TCMs [YJL08] und TMMs [SAB+00, SB05] eine Optimierung unter Unsicherheiten. Hierbei wird zudem nur eine einkriterielle Optimierung ermöglicht. Insbesondere werden hierbei Inkonsistenzen und Ineffizienzen der aktuellen Abdeckung aufgrund von einer beispielsweise gestörten Kommunikation identifiziert. Es wird vorgeschlagen, eine geeignete Neuplanung von Aufgaben und eine Reorganisation der Agenten durchzuführen. Bezüglich einer realen Abdeckung, für die zuvor keine Umgebungsinformationen vorliegen, muss ein autonomes MA(R)S grundsätzlich Entscheidungen unter Unsicherheiten treffen. Hierbei können bereits gesammelte Umgebungsinformationen genutzt werden, um auf mögliche Ausprägungen der unbekannten Umgebung schließen zu können.

Mit dem SMO-TCM wird eine Untersuchung von Unsicherheiten für den einkriteriellen Fall durchgeführt. Aufgrund des verfolgten Ansatzes, eine Transformation des einkriteriellen SSPP in ein allgemeines mehrkriterielles Problem zu überführen, wird ebenfalls eine Untersuchung für das mehrkriterielle SSPP ermöglicht. Vor allem können Unsicherheiten vom Typ I einbezogen werden. Unsicherheiten vom Typ I umfassen unkontrollierbare Faktoren, die aus einer dynamischen oder unbekannten Systemumwelt resultieren und zu unvollständigen Umgebungsinformationen führen. Mit Hilfe eines geeigneten Umgebungsmodells und eines Schätzmodells können optimierte Abdeckungen ermittelt werden. Hierbei werden neugewonnene Umgebungsinformationen kontinuierlich genutzt, um dynamisch eine Schätzung der unbekannten Umgebung vorzunehmen. Eine optimierte Abdeckung wird mittels der robusten Optimierung untersucht. Entsprechend werden Umgebungsinformationen, die mit Unsicherheiten behaftet sind, geeignet bewertet.

Zusammenfassend werden die nachfolgenden Zielsetzungen mit der Einführung des SMO-TCM verfolgt:

- Beitrag zur Verbesserung der Akzeptanz und positive Wahrnehmung des TCP und MARS
- Beitrag zur Weiterentwicklung von neuen, robusten Ansätzen für das TCP zur Erweiterung des Einsatzspektrums eines MARS
- Einführung erweiterter Annahmen für eine realitätsnähere Simulation
- Einführung und Untersuchung einer Entscheidungsunterstützung unter Unsicherheiten für autonome Agenten hinsichtlich des TCP
- Untersuchung der Auswirkungen auf eine Abdeckung unter Einflussnahme von Unsicherheiten, hier Typ I
- Einführung und Untersuchung des SSPP für das TCP

Nachfolgend werden zuerst die Annahmen eingeführt auf denen das SMO-TCM basiert. Des Weiteren werden Maßnahmen zur Steigerung der Robustheit für das MA(R)S integriert. Daran anschließend wird die allgemeine Funktionsweise des MA(R)S beschrieben sowie die Routenplanung im Sinne des SSPP auf das SMO-TCM überführt.

5.1.1. Annahmen

Für das SMO-TCM werden geeignete Annahmen getroffen, um Rückschlüsse auf reale Beobachtungen hinsichtlich des Einsatzes von einem MA(R)S ziehen zu können. Das vereinfachte Abbild Ω' einer realen Systemumgebung Ω wird in Form einer abstrahierten zweidimensionalen Darstellung mittels eines planaren Graphen beschrieben. Die Systemumgebung unterteilt sich in quadratische

Zellen z_i mit $\Xi' \subseteq \cup_{i \in I} z_i$. Eine Zelle kann zwei Ausprägungen annehmen. Entweder ist eine Zelle frei und kann vollständig befahren werden oder sie gilt als blockiert, wenn die Zelle teilweise oder vollständig durch ein Hindernis belegt ist. Die Ausmaße, der in Größe und Form identischen Zellen, richtet sich nach der Sensorenreichweite der Agenten $a_k \in A$. Hierbei wird die Annahme getroffen, dass ein Agent jede der acht direkten Nachbarzellen auf Hindernisse untersuchen kann. Die Nachbarschaft einer Zelle z_i definiert sich durch die Menge aller mit z_i direkt verbundenen Zellen. Zwei Zellen sind direkt miteinander verbunden, wenn sie eine gemeinsame Zellgrenze aufweisen. Eine Zelle gilt als besucht, wenn ein Agent die Zelle befahren hat. Eine Zelle kann von mehreren Agenten zeitgleich befahren werden. Eine Systemumgebung Ω' ist vollständig abgedeckt, wenn die Menge aller besuchten Zellen gleich der Menge aller freien Zellen in Ω' ist. Eine vollständige Abdeckung impliziert ferner, dass keine unbekannte Zelle mehr in der Systemumgebung vorliegt. Eine vollständige Abdeckung der besuchbaren Zellen kann stets gewährleistet werden, auch wenn ein Teil oder die gesamte Systemumgebung unbekannt ist. Ebenfalls ist das SMO-TCM für alle Systemumgebungen anwendbar, unabhängig von Ausprägungen der Hindernisstrukturen oder Umgebungsbegrenzungen. Hierbei kann eine Anpassung der Zellgröße notwendig sein. Hingegen muss die maximale flächenmäßige Ausdehnung der abzudeckenden Systemumgebung angegeben werden. Für den Fall, dass die Systemumgebung unbekannt ist, muss eine genügend große Ausdehnung angegeben werden. Sollte sich während der Abdeckung herausstellen, dass die tatsächliche Systemumgebung kleiner ist als die ursprünglich angegebene Ausdehnung, resultieren keine Beeinträchtigungen hinsichtlich einer vollständigen Abdeckung.

Die Abdeckung der Systemumgebung wird durch autonome Agenten durchgeführt. Hierbei treffen die Agenten alle Entscheidungen bezüglich einer optimierten Aufgabenbearbeitung einschließlich einer Routenplanung selbstständig, ohne eine Einflussnahme und Unterstützung durch menschliche Bediener. Hierfür ist die Ausprägung eines integrierten Prozesses der Wahrnehmung, Analyse, Kommunikation mit anderen Agenten, Planung, Entscheidungsunterstützung und Handeln notwendig. Das Handeln basiert auf den aus der Systemumwelt wahrgenommenen und analysierten Informationen. Alle Agenten verfügen über die gleichen Sensoren sowie Effektoren und sind baugleich. Des Weiteren verwaltet jeder Agent ein entsprechendes Abbild Ω' der Systemumgebung Ω und aktualisiert das Abbild fortlaufend auf Grundlage der wahrgenommenen Informationen. Es werden Informationen bezüglich Strömungsrichtungen der Zellen sowie Hindernissen gespeichert. Die Wahrnehmung der Strömungsrichtung setzt einen Besuch der jeweiligen Zelle durch einen Agenten voraus. Hingegen können Informationen bezüglich der Befahrbarkeit für alle Zellen der direkten Nachbarschaft wahrgenommen werden. Die Zellen unterliegen keinen Änderungen. Dementsprechend kann sich zum einen der Zustand [blockiert|frei] und zum anderen die tatsächlichen Strömungskosten auf den Zellen nicht verändern. Zudem unterliegen die Sensoren keinen Störungen. Demnach behalten gesammelte Informationen über die gesamte Abdeckung ihre Gültigkeit.

Die Kommunikationsreichweite der Agenten muss vordefiniert werden. Demnach ist es Agenten möglich, Informationen direkt miteinander, innerhalb der Kommunikationsreichweite, auszutauschen. Des Weiteren besteht die Möglichkeit, indirekt Informationen miteinander, innerhalb eines mobile Ad-hoc Kommunikationsnetzwerkes, auszutauschen. Ein Kommunikationsnetzwerk umfasst alle Agenten, die direkt oder indirekt über andere Agenten miteinander kommunizieren können. Alle Agenten fungieren gleichzeitig als Sender und Empfänger, um Informationen weiterleiten zu können [BSW08, WHH$^+$08]. Entsprechend können Informationen über die eigene Kommunikationsreichweite hinweg übermittelt werden. Hindernisse und Umgebungsstrukturen haben keinen

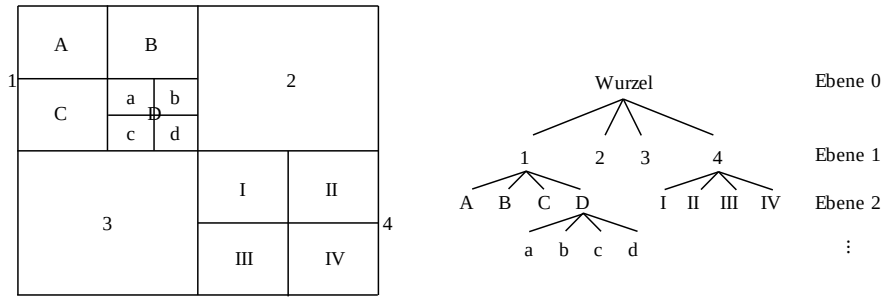


Abbildung 5.1.: Beispiel einer Quadtree Zerlegung, um eine Reihenfolge der zu besuchenden Zellen zu bestimmen. Unter a) Hierarchische Zerlegung einer Systemumgebung, b) Kodierter Quadtree mit Hierarchieebenen (nach [LC98], Seite 660).

Einfluss auf die Kommunikation.

Wird ein Hindernis durch einen Agenten wahrgenommen, dann wird die Zelle als blockiert gespeichert. Liegt sogar eine blockierte Zelle auf der geplanten Route und hindert somit dem Agenten an der Befahrung, dann wird im aktuellen MANET zum einen die Aufgabenbearbeitung überprüft und eine zugehörige Route in jedem Fall neu geplant. Werden beispielsweise Hindernisse vollständig umfahren und resultierend unbekannte Zellen wahrgenommen, die aufgrund von blockierten Zellen durch keinen Agenten erreicht werden können, werden die unbekannten Zellen aus ebenfalls als blockiert markiert.

Die Reihenfolge der zu bearbeitenden Aufgaben und somit die Reihenfolge der zu überprüfenden Zellen basiert maßgeblich auf der zu Beginn der Abdeckung ermittelten Quadtree Zerlegung. Die Systemumgebung Ω wird in gleichgroße Zellen unterteilt. Es muss die maximale horizontale und vertikale Ausdehnung $\Omega_{x,y}$ der Systemumgebung bekannt sein. Die Größe der Zellen entspricht der Sensorenreichweite. Bei der Quadtree Zerlegung wird die zweidimensionale Repräsentation der Systemumgebung Ω' in Hierarchieebenen unterteilt. Jede Hierarchieebene enthält eine bestimme Anzahl von Zellen. Die Hierarchieebene 0 enthält die Wurzel des Baumes und wird durch die gesamte Systemumgebung repräsentiert. Ausgehend von der gesamten Systemumgebung wird in der Hierarchieebene 1 eine Unterteilung in vier gleichgroße Teilregionen vorgenommen. Die Teilregionen werden jeweils durch eine Zelle im Zentrum der Teilregionen repräsentiert. Das Vorgehen wird rekursiv fortgesetzt, bis die Größe der Teilregionen jeweils einer Zelle entspricht. Die jeweiligen Zellen der Hierarchieebenen spiegeln ausgehend von der Wurzel die Reihenfolge der zu bearbeitenden Zellen wider [LC98]. In Abbildung 5.1 ist eine Beispielzerlegung angegeben. Zusammenfassend wird mit Hilfe der Quadtree Zerlegung eine dislozierte Abdeckung der Systemumgebung unterstützt. Entsprechend werden verschiedene Teilregionen durch Agenten verteilt untersucht. Insbesondere durch die in der Systemumgebung räumlich verteilten Aufgabenziele werden unterschiedlich Regionen parallel abgedeckt. Ferner können Informationen über die Systemumgebung wie beispielsweise Hindernisse und verursachte Kosten der Zellabdeckungen gesammelt werden. Die Informationen können fortlaufend in die Aufgabenerfüllung einschließlich der Routenplanung einfließen und einen positiven Einfluss auf die Effizienz nehmen.

In jedem Zeitschritt befährt ein Agent eine freie Zelle in der direkten Nachbarschaft. Die Strömungsrichtung hat einen Einfluss auf den Energieverbrauch, weil die Agenten alle Zellen in gleichbleibender Geschwindigkeit befahren. Hierbei resultieren Kosten, die abhängig von der Strömungsrich-

tung und der Fahrtrichtung des Agenten bestimmt werden. Resultierend liegen bei einer frontalen Gegenströmung höhere Kosten und somit ein höherer Energieverbrauch vor. Um eine optimierte Routenplanung durchführen zu können, wird eine kontinuierliche Approximation der Strömungsrichtung durchgeführt. Hierbei wird die Annahme getroffen, dass die Strömungsrichtung ähnlich, der Zellen in der zugehörigen Nachbarschaft ist. Aus diesem Grund werden die Informationen über die Strömungsrichtungen der bekannten Zellen genutzt, um eine Approximation für unbekannte Zellen bestimmen zu können. Die Informationen bezüglich der Strömungsrichtungen $\mathbf{d}_R(z_i)$ diffundieren zum Zeitschritt t der Zelle z_i in die Nachbarschaft $\Phi(z_i, t)$ über alle Zellen $z_j \in \Phi(z_i, t)$. Der Informationsübergang nimmt mit der Entfernung $f(z_i, z_j)$ von z_i zu z_j exponentiell ab. Somit bestimmt sich für eine unbekannte, sprich bisher nicht besuchte Zelle z_i zum Zeitschritt t die approximierte Strömungsrichtung zu

$$\mathbf{d}_R(z_i) \leftarrow \mathbf{d}_R(z_i) + \sum_{j \in \Phi(z_i,t)} \frac{1}{f(z_i, z_j)} \mathbf{d}_R(z_j) \tag{5.1}$$

mit

$$f(z_i, z_j) = \max(|x_{z_i} - x_{z_j}|, |y_{z_i} - y_{z_j}|) \tag{5.2}$$

Der Informationsübergang wird für einen maximalen Radius von $f(z_i, z_j) \leq 5$ durchgeführt. Der Radius ist mittels einer Experimentierung ermittelt worden. Nichtsdestotrotz sollte in weiteren Untersuchungen der Informationsübergang untersucht werden.

Bilden mehrere Agenten ein MANET und tauschen Informationen bezüglich der Systemumgebung aus, ist nachfolgendes Vorgehen zur Bestimmung einer abgestimmten Approximation unbekannter Zellen realisiert. Zuerst wird auf der Grundlage der verfügbaren Strömungsrichtungen $\mathbf{d}_R(z_i)$ mit $z_i \in Z$ eine fusionierte Systemumgebung über alle Agenten $a_k \in$ MANET ermittelt. Mit Z wird nachfolgend die Menge aller Zellen in Ξ' beschrieben. Es wird für jede Zelle z_i der Informationsgehalt über alle Agenten geprüft und die Strömungsrichtung $\mathbf{d}_R(z_i)$ mit der höchsten Eintrittswahrscheinlichkeit, auf einer der acht Einträge innerhalb des Vektors, übernommen. Entsprechend kann garantiert werden, dass bereits bekannte Zellen mit einer resultierenden Eintrittswahrscheinlichkeit von eins auf einen der acht Einträge übernommen werden. Im Anschluss daran wird die zuvor in Gleichung (5.1) beschriebene Diffusion der Informationen über unbekannte Zellen durchgeführt. Abschließend werden die fusionierten Strömungsinformationen an die Agenten im zugehörigen MANET verteilt.

Zu Beginn der Abdeckung sind alle Zellen unbekannt. Demnach liegen keine Informationen bezüglich der Strömungsrichtung vor. Die Eintrittswahrscheinlichkeit der acht möglichen Strömungsrichtungen sind gleichverteilt. Es ergibt sich für eine Zelle z_i der entsprechende Einheitsvektor $\mathbf{d}_R(z_i) = (.125\ \ .125\ \ .125\ \ .125\ \ .125\ \ .125\ \ .125\ \ .125)^T$. Ferner gelten die Annahmen aus 4.3.1.

Die Kosten zur Befahrung einer unbekannten Zelle mit approximierten Strömungsinformationen werden gemäß der Gleichung (4.24) bestimmt.

Bezugnehmend auf die Darstellung der Teilbereiche des TCP in Kapitel 2.4 werden weitere grundlegende Annahmen für das SMO-TCM eingeführt. Die Agenten können zu jeder Zeit eine exakte Positionsbestimmung durchführen. Dementsprechend können geplante Routen zuverlässig befahren werden und es resultieren keine Positionsabweichungen, die zu Fehlern wie Inkonsistenzen führen

können. Die Agenten befahren in jedem Zeitschritt das Zentrum einer benachbarten Zelle. Folglich ist die Bewegungsgeschwindigkeit unabhängig von der Strömung nicht aber von den verursachten Kosten. Die Annahme wird eingeführt, um zum einen eine direkte Vergleichbarkeit mit anderen Modellen gewährleisten zu können. Zum anderen können die Kosten zur Abdeckung einer Systemumgebung unabhängig des Explorationsfortschrittes analysiert werden. Die Strömung führt ebenfalls zu keinen Abweichungen von der geplanten Route. Während der Abdeckung wird ein Abbild der Systemumgebung erstellt. Sensoren, die eine Kartographie von Umgebungsstrukturen ermöglichen, unterliegen keinen Messfehlern. Zudem können die individuell kartographierten Systemumgebungen innerhalb einer vordefinierten Kommunikationsreichweite ausgetauscht und direkt miteinander in Beziehung gesetzt werden.

Auf der Grundlage der Annahmen können Maßnahmen integriert werden, die zu einem robusteren Verhalten der Agenten führen sollen. Hierbei bietet vor allem die Abdeckung der Systemumgebung mittels dislozierter und verteilter Agenten ein großes Potenzial. Im gleichen Maße müssen jedoch auch mögliche Inkonsistenzen bezüglich der fusionierten Umgebungsinformationen erkannt und geeignet aufgelöst werden.

5.1.2. Maßnahmen zur Steigerung der Robustheit

Die Abdeckung einer unbekannten Umgebung stellt hohe Anforderungen an autonome Agenten. Zum einen bestehen Herausforderungen hinsichtlich einer exakten Positionsbestimmung, Bewegungskontrolle, Kartographie und Zusammenführung von Informationen zu einem übergeordneten Lagebild. Zum anderen müssen Aufgaben zwischen den Agenten und Routenplanungen auf der Grundlage von unvollständigen und unsicheren Informationen getroffen werden. Ferner müssen die Agenten ein robustes Verhalten aufweisen, um auf zuvor unbekannte Situationen geeignet reagieren zu können. Vor allem sollten Ansätze zur Konfliktlösung integriert sein, um Inkonsistenzen und Störungen erkennen und aufheben zu können [Fer94].

Das SMO-TCM integriert verschiedene Ansätze, um ein robustes Verhalten der Agenten und eine robuste Optimierung unter Unsicherheiten auszuprägen. Hierfür werden die Ansätze aus Kapitel 3.3.2 aufgenommen und geeignet an das TCP angepasst:

- **Integriertes Beobachtungssystem:**
  Ein integriertes Beobachtungssystem ermöglicht die Identifikation von Änderungen bezüglich des Lagebildes, einschließlich der Aufgabenerfüllung, Systemumwelt und des Verhaltens anderer Agenten. Jeder Agent verfügt über ein Beobachtungssystem, welches kontinuierlich in jedem Zeitschritt eine Prüfung des individuellen Lagebildes durchführt, um Inkonsistenzen der aktuellen Aufgabenbearbeitung und Änderungen der Systemumwelt zu erkennen. Hierzu wird nach einer Fusionierung der individuell verwalteten Aufgabenlisten überprüft, ob die momentan zu bearbeitende Aufgabe bereits durch einen anderen Agenten erfüllt worden ist oder gerade bearbeitet wird. Des Weiteren werden im Verlauf der Abdeckung neu hinzugewonnene Informationen der Systemumwelt identifiziert, die dazu führen können, dass die geplante Route aufgrund von zuvor nicht berücksichtigten Hindernissen nicht mehr befahren werden kann. Grundsätzlich ist eine Auflösung der Inkonsistenzen und Anpassung der Routen nicht Bestandteil eines Beobachtungssystems. Zusammenfassend stellt die Identifikation von

Änderungen, Inkonsistenzen und Ineffizienzen und somit das integrierte Beobachtungssystem oftmals eine Grundlage für die nachfolgenden Ansätze dar.

- **Zeitabhängig periodisches System:**
  Ein zeitabhängig periodischer Ansatz der Aufgabenplanung kann sich positiv auf eine Komplexitätsreduzierung und Steigerung der Robustheit auswirken. In diskreten Zeitabständen oder zu vordefinierten Ereignissen kann das System in einen stabilen Zustand zurückgesetzt werden. Des Weiteren unterliegt die Aufgabenplanung unter Unsicherheiten einer hohen Komplexität. Insbesondere die Bestimmung einer Reihenfolge von nacheinander zu bearbeitenden Aufgaben kann aufgrund der Unsicherheiten zu ineffizienten Entscheidungen führen. In Zusammenwirkung mit einem integrierten Beobachtungssystem kann die Aufgabenbearbeitung der Agenten zurückgesetzt werden und eine Neuplanung der Aufgaben vorgenommen werden. Des Weiteren wird über einen marktbasierten Ansatz eine kontinuierliche Planung der Aufgaben und Koordination der Agenten vorgenommen. Die Aufgabenplanung wird stets auf einen engen Zeithorizont beschränkt. Die Unsicherheiten bezüglich der Bearbeitung von Aufgaben nehmen mit immer weiter in die Zukunft gerichteten Betrachtungen zu.

- **Verteilte Infrastruktur:**
  Eine verteilte Infrastruktur ermöglicht eine parallele Bearbeitung von Aufgaben durch dislozierte Agenten. Daneben bedarf es eines erhöhten Koordinierungsaufwands der Agenten, um die Aufgabenbearbeitung auf das übergeordnete Ziel zwischen den Systemkomponenten abzustimmen. Für das SMO-TCM bietet der Einsatz von verteilten Agenten den Vorteil, dass Informationen aus unterschiedlichen Bereichen der Systemumgebung parallel gesammelt werden können. Die Informationen können in einem gemeinsamen Lagebild fusioniert werden und für eine Schätzung der noch unbekannten Systemumgebung genutzt werden. Des Weiteren ist das SMO-TCM robuster gegenüber dem Ausfall eines Agenten. Die Aufgaben eines ausgefallenen Agenten können durch andere Agenten übernommen werden. Zusätzlich können weitere Agenten in einen MAS mit verteilter Infrastruktur aufgenommen werden und tragen zur Erhöhung der Performance und Robustheit bei.

- **Flexibles Planungssystem:**
  Auf der Grundlage eines integrierten Beobachtungssystems wird eine angepasste und flexible Planung ermöglicht, um möglichst frühzeitig auf Änderungen und Ineffizienzen reagieren zu können. In Abhängigkeit vordefinierter Situationen wird mit dem marktbasierten Ansatz eine Koordinierung der Aufgaben und Planung der Routen kontinuierlich durchgeführt. Zudem wird eine wiederholte Planung von Aufgaben unterstützt, wenn Ineffizienzen erkannt worden sind. Eine situativ bedingte Aufgabenplanung hat für das TCP den Vorteil, dass Ressourcen nur im Bedarfsfall gebunden werden.

- **Verteilte Ressourcennutzung:**
  Während der Problemlösung bewegen sich die Agenten in der Systemumgebung. Ist das weitere Vorgehen unbekannt, müssen die Agenten sämtliche Aktionen stoppen. Ungeplante und somit nicht überprüfte Bewegungen in der Systemumgebung bürgen das Risiko von ineffizienten Aktionen oder sogar Kollisionen, die zu einem Ausfall führen können. Um ein Aufstoppen und somit eine möglichst schnelle und effiziente Problemlösung ermöglichen zu können, kann die Definition von Teilproblemen sowie deren Lösung mit Hilfe von verteilten Ressourcen einen positiven Einfluss nehmen. Der marktbasierte Ansatz des SMO-TCM unterstützt eine

verteilte Problemlösung in dem Sinne, dass geeignete Routen von jedem Agenten bestimmt werden. Anschließend werden die Teilprobleme der Routenplanungen zusammengeführt, um das Problem der Koordinierung zwischen den Agenten bearbeiten zu können. Zudem kann trotz eines Ausfalls einzelner Agenten die Aufgabenerfüllung fortgesetzt werden. Demnach ist ein MAS mit verteilter Ressourcennutzung und Problemlösung robuster gegenüber Störungen der Agenten als ein MAS mit zentraler Ressourcennutzung.

5.1.3. Funktionsweise

Die Menge A umfasst alle Agenten a_k, die eingesetzt werden, um eine Systemumgebung Ω vollständig abzudecken. Die Systemumgebung Ω wird gemäß der Quadtree Zerlegung in Zellen z_i gleicher Größe repräsentiert. Des Weiteren wird auf der Grundlage der Quadtree Zerlegung die Reihenfolge der Aufgaben $\mathcal{T}$ mit $\mathcal{Z} \rightarrow \mathcal{T}, z \rightarrow \mathfrak{t}$ festgelegt. Entsprechend wird jeder Zelle $z_i \in \mathcal{Z}$ genau eine Aufgabe $\mathfrak{t}_i \in \mathcal{T}$ zugeordnet. Es existiert solange eine entsprechende Aufgabe $\mathfrak{t}_i$ bis die Zelle z_i durch einen Agenten besucht wird. Eine Zelle z_i wird durch die horizontale und vertikale Lage und einem Zustand definiert. Zu Beginn der Abdeckung sind alle Zellen unbekannt. Wird eine Zelle durch einen Agenten befahren, ist die Zelle in jedem Fall frei. Zusätzlich kann der Zustand einer Zelle in der direkten Nachbarschaft $\Phi_{a_k}(z_i, t)$ eines Agenten zum Zeitschritt t durch die Sensoren wahrgenommen werden. Resultierend liegt für jede Zelle immer einer der drei Zustände [Unbekannt|Frei|Hindernis] vor. Resultierend wird eine $(0,1)$-Abdeckungsmatrix $\mathcal{M}_C$ und $(0,1)$-Hindernismatrix $\mathcal{M}_O$ abgeleitet. Hierbei gilt für alle Einträge in $\mathcal{M}_C, \mathcal{M}_O$

$$m_{C_{i,j}} = \begin{cases} 1, & \text{wenn } z_{i,j} \text{ ist besucht} \\ 0, & \text{sonst} \end{cases} \tag{5.3}$$

und

$$m_{O_{i,j}} = \begin{cases} 1, & \text{wenn } z_{i,j} \text{ ist blockiert} \\ 0, & \text{sonst.} \end{cases} \tag{5.4}$$

Weiterhin liegen vollständige Informationen bezüglich der Strömungsrichtungen für alle befahrenen Zellen vor. Hingegen werden für alle anderen Zellen eine Schätzung der Strömungsrichtungen vorgenommen. Die Genauigkeit der Schätzung nimmt mir zunehmender Abdeckung zu, weil mehr Umgebungsinformationen einbezogen werden können.

Für jeden Agenten wird die selbe Aufgabenreihenfolge $\mathcal{T}$ der zu befahrenden Zellen gemäß der Quadtree Zerlegung vorinitialisiert. Die Aufgabenliste wird von jedem Agenten selbstständig aktualisiert. Zusätzlich verfügt jeder Agent über eine individuelle Repräsentation Ω' der Systemumgebung. Selbstständig wahrgenommene oder ausgetauschte Informationen bezüglich der Zellzustände führen zu einer individuellen Anpassung von Ω'. In gleicher Weise verwaltet jeder Agent eine individuelle Matrix, die entsprechend der Systemumwelt die Strömungsrichtung für jede Zelle abbildet. Falls keine Informationen über die Systemumwelt vorliegen, werden die individuellen Repräsentationen der Systemumgebung derart initialisiert, dass jede Zelle den Zustand unbekannt aufweist. Die Einträge der heuristischen Informationen bezüglich der Strömungen werden in der Heuristikmatrix $\mathcal{H}^{EWK}$ mit dem nachfolgenden Vektor

$$\eta^{EWK(z_i)} = (.125 \;\; .125 \;\; .125 \;\; .125 \;\; .125 \;\; .125 \;\; .125 \;\; .125)^T, \forall z_i \in \mathcal{Z} \tag{5.5}$$

Algorithmus 5.1 : Grundalgorithmus für SMO-TCM

```
Eingabe : |A|, Ω'
Ausgabe : Vollständige Abdeckung
Initialisiere: Aufgabenliste 𝒯 nach Quadtree Zerlegung, Kostenmatrix mit gleichverteiltem
Wahrscheinlichkeitsvektor, a_k ∈ A mit (z_S, 𝒯, Abdeckungsmatrix = ∅, Hindernismatrix = ∅,
Kostenmatrix)
while Unbesuchte Zellen do
    foreach Agent a_k do
        Untersuche Nachbarschaft: Aktualisiere Aufgabenliste, Hindernismatrix
        Ermittle MANET: Aktualisiere Aufgabenliste, Abdeckungsmatrix, Hindernismatrix
        if Hindernis liegt auf der Route des zugehörigen Agentens a_k(Route) then
            Füge das Ziel der Route der Aufgabenliste 𝒯 vorne hinzu
            Verteilt Aufgabenliste im MANET
            a_k(Route) = ∅
        end
        if a_k(Aufgabenliste) != ∅ then
            if a_k(Route) == ∅ then
                while a_k(Route) == ∅ ∧ a_k(Aufgabenliste) != ∅ do
                    Initiiere Auktion mit (a_k, MANET, Aufgabenliste, Abdeckungsmatrix,
                    Hindernismatrix, Kostenmatrix)
                    (siehe Algorithmus 5.2)
                end
            end
        end
    end
    foreach a_k ∈ A do
        if a_k(Route) != ∅ then
            Gehe zur nächsten Zelle
            Aktualisiere Abdeckungsmatrix
        end
    end
end
return Vollständige Abdeckung
```

initialisiert. Entsprechend sind die Eintrittswahrscheinlichkeiten der Realisierungen über alle möglichen acht Strömungsrichtungen gleichverteilt.

Die Koordination der autonomen Agenten basiert auf einem definierten Regelwerk und Handlungsanweisungen. Auf der Grundlage eines Informationsaustausches treffen die Agenten alle für die Befahrung und vollständigen Abdeckung notwendigen Entscheidungen selbst. Grundsätzlich kann jeder Agent dezentral eine vollständige Abdeckung gewährleisten und ist nicht auf andere Agenten angewiesen. Zudem existiert keine übergeordnete, zentrale Steuerungseinheit. Ein zentraler Informationsaustausch kann sich jedoch positiv auf die Gesamtperformance auswirken. Zusammenfassend wird ein hybrider Ansatz umgesetzt. Neben einer dezentralen Aufgabenbearbeitung werden mittels eines marktbasierten Ansatzes unterschiedliche Konzepte und Mechanismen eines zentralen Ansatzes kombiniert. In dem Algorithmus 5.1 wird der grundlegende Ablauf einer Abdeckung definiert und im nachfolgenden beschrieben. Hierbei wird die zuvor in Kapitel 4.3 beschriebene Routenplanung für das SSPP in den markbasierten Ansatz integriert.

Jeder Agent a_k startet in einer zu Beginn zugewiesenen Zelle $z_S \in \mathcal{Z}$. Solange noch unbekannte Zellen z_i in der individuellen Repräsentation der Systemumgebung Ω' eines Agenten, befährt dieser die noch abzudeckenden Zellen. Hierbei richtet sich die Reihenfolge nach den zugehörigen Aufgaben $\mathfrak{t}_i \in \mathcal{T}$. In jedem Zeitschritt wird das MANET gemäß der Kommunikationsreichweite bestimmt. Das MANET umfasst alle direkt und indirekt benachbarten Agenten. Die Zugehörigkeit der Agenten zu einem MANET kann sich fortlaufend ändern. Des Weiteren kann die Anzahl existierender MANETs von einem bis zu $|A|$ variieren. Tendenziell führt eine erhöhte Kommunikationsreichweite zu größeren MANETs. Die eindeutige Kennzeichnung der Agenten ermöglicht den Aufbau eines MANET. Ausgehend von jedem Agenten wird eine Verbindungsanfrage zu allen direkt benachbarten Agenten innerhalb der definierten Kommunikationsreichweite geschickt. Jeder Agent der eine Verbindungsanfrage erhalten hat, bestätigt den Empfang. Dieses Vorgehen ermöglicht jedem Agenten einen Vektor zu ermitteln, der die direkt benachbarten Agenten aufführt. Alle Vektoren werden anschließend über eine ODER-Relation zusammengeführt. Nachfolgend wird mit dem Ausdruck in (5.6) ein Beispiel exemplarisch für den Agenten a_1 und dessen empfangenen Verbindungsvektoren der Agenten $a_2, \ldots, a_5$ veranschaulicht.

$$\begin{array}{c} \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{array} \begin{array}{c} a_1 \\ \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \end{array} \overset{OR}{\rightarrow} \begin{array}{c} a_2 \\ \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \end{array} \overset{OR}{\rightarrow} \begin{array}{c} a_3 \\ \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \end{array} \overset{OR}{\rightarrow} \begin{array}{c} a_4 \\ \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \end{array} \overset{OR}{\rightarrow} \begin{array}{c} a_5 \\ \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \end{array} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \hat{=} \{a_1, a_4, a_5\} \tag{5.6}$$

Die Agenten a_1, a_4, a_5 können keine Aussage darüber treffen, ob die Agenten a_2, a_3 jeweils ein eigenständiges MANET bilden oder in einem gemeinsamen zusammengeführt sind. Innerhalb von jedem MANET wird die Gesamtheit aller gesammelten Informationen ausgetauscht. Die Informationen schließen die noch zu bearbeitenden Aufgaben in $\mathcal{T}$, die Abdeckungs- $\mathcal{M}_C$ und Hindernismatrizen $\mathcal{M}_O$ sowie die heuristischen Informationen bezüglich der Strömungsinformationen $\mathcal{H}^{EWK}$ ein. Mit Hilfe des Informationsaustausches können Redundanzen in der Abdeckung minimiert werden und geeignete Routen hinsichtlich Güte und Robustheit ermittelt werden. Der Informationsaustausch wird in jedem Zeitschritt durchgeführt. Wird hierbei eine Hinderniszelle auf einer Route der Agenten identifiziert, dann wird die Route gelöscht und das Aufgabenziel der Aufgabenliste $\mathcal{T}$ vorne hinzugefügt und somit neu vergeben. Mit einem Aufgabenziel wird eine konkrete Zelle assoziiert, die durch einen Agenten besucht werden soll.

Die Selbstkoordination der Agenten wird mit einem hybriden Ansatz umgesetzt. Zum einen führen die Agenten eine kontinuierliche Abstimmung über die noch offenen Zellen während der gesamten Abdeckung durch. Zum anderen findet eine Koordination ausschließlich zu diskreten Ereignissen statt. Es werden zwei Ereignisse unterschieden, für die eine Selbstkoordination mit Hilfe eines marktbasierten Ansatzes durchgeführt wird. Demzufolge wird eine Auktion initiiert, wenn ein Agent über keine aktuell zu bearbeitende Aufgabe verfügt. Des Weiteren wird eine Auktion genau dann initiiert, wenn ein Agent auf ein Hindernis trifft. Die geplante Route kann entsprechend nicht mehr befahren werden. Aus diesem Grund kann eine Überprüfung der Performance und resultierenden Neuplanung der unterbrochenen Aufgabe zu einer Performancesteigerung führen.

Für beide Ereignisse muss eine noch zu bearbeitende Aufgabe an den Agenten vergeben werden. Hierbei fungiert der Agent als Auktionator und stellt die Aufgabe im MANET zur Auktion bereit. Alle Agenten im MANET nehmen unabhängig von ihrer momentanen Aufgabenbearbeitung als Bieter an der Auktion teil. Resultierend ist es möglich, dass ein Agent mehrere Aufgaben übernimmt. Die bereits übernommenen Aufgaben werden von dem Agenten in einer geordneten Liste *offeneZiele* verwaltet. Verfügt der Agent über keine aktuell zu bearbeitende Aufgabe, dann spiegelt das Gebot die Kosten für die Route vom aktuellen Standort zu der in der Auktion zu vergebenen Zelle wider. Andernfalls setzt sich ein Gebot aus den Kosten der bisher geplanten Route zu den bereits übernommenen Zellen aus der Liste *offeneZiele* sowie der ermittelten Route ausgehend vom letzten Ziel zu der in der Auktion zu vergebenen Zelle zusammen. Unter den mitbietenden Agenten wird das geeignetste Gebot beziehungsweise Route ermittelt. Anschließend werden alle Agenten über das Ergebnis informiert. Die geeignetste Route wird dem zugehörigen Agenten angefügt. Es werden vom Auktionator solange Auktionen nacheinander durchgeführt, bis dieser selber eine Auktion gewinnt und entsprechend eine neue Route zur angebotenen Zelle zugewiesen bekommt. Für den Sonderfall, dass neben dem Auktionator kein anderer Agent im gleichen MANET ist, wird der Auktionator in jedem Fall der Gewinner der Auktion sein. In diesem Fall wird die Bestimmung einer geeigneten Route durchgeführt, welche ebenfalls Bestandteil einer Auktion ist.

Die Bestimmung einer Route, welche als Gebot einer Auktion fungiert, basiert auf dem in Kapitel 4.3 vorgestellten Ansatz zur Lösung des SSPP. Für die bereits bestehende Route werden die gemittelten Kosten entsprechend der aktuell vorhanden Strömungsinformationen auf den Zellen bestimmt. Basierend auf den Kosten wird anschließend die Eintrittswahrscheinlichkeit approximiert. Die Kosten und Eintrittswahrscheinlichkeit für die in der Auktion stehende Zelle wird analog dem vorgestellten SSPP in Kapitel 4.3 angegeben. Hierbei werden ebenfalls zwei mögliche Realisierungen einer Route bestimmt. Neben der exakten Realisierung entsprechend des zugrundeliegenden Entscheidungsprozesses der Ameise auf der Route, wird zudem eine zufällige Realisierung bestimmt. Für beide Realisierungen wird basierend auf den resultierenden Kosten die entsprechende Eintrittswahrscheinlichkeit approximiert. Folgend liegen die Kosten und zugehörige Eintrittswahrscheinlichkeit für die unter Umständen noch zu befahrende Route sowie für die zwei Realisierungen der neu bestimmten Route vor. Die Kosten der zwei Routen (noch zu befahrende Route und neue Route) werden aufaddiert und die Eintrittswahrscheinlichkeit unter Berücksichtigung des Verhältnisses der jeweiligen Kosten angegeben. Um die Redundanzen bei der Abdeckung von Zellen und somit auch die Fahrtkosten reduzieren zu können, werden zusätzlich die noch unbesuchten Zellen berücksichtigt. Hierbei werden von den ermittelten Kosten die Anzahl der unbesuchten Zellen auf der Route abgezogen. Befährt ein Agent bisher noch unbesuchte Zellen in Strömungsrichtung resultieren Kosten in Höhe von null Einheiten, da sich die Fahrtkosten von jeweils einer Einheit mit den unbesuchten Zellen aufheben.

Auf der Grundlage der Pareto-optimalen Routen wird die Auswahl einer konkreten Route vorgenommen. Es wird ein dynamischer Ansatz realisiert, der versucht, den Explorationsfortschritt in der Entscheidungsfindung zu berücksichtigen. Zu Beginn der Exploration wird eher eine Route mit geringen Kosten ausgewählt. Die Eintrittswahrscheinlichkeit ist hierfür ebenfalls eher gering. Im Verlauf der Exploration werden zunehmend Routen mit einer hohen Eintrittswahrscheinlichkeit ausgewählt. Das Vorgehen wird über zwei Sachverhalte begründet. Zu Beginn der Exploration liegen nur wenige Strömungsinformationen vor. Resultierend sind die approximierten Routen generell mit hohen Unsicherheiten behaftet. Auch eine Route mit einer hohen Eintrittswahrscheinlichkeit

kann auf falschen Informationen beruhen. Des Weiteren liegt zu Beginn der Exploration eine hohe Wahrscheinlichkeit vor, dass unbekannte Zellen auf der Route abgedeckt werden.

Neben der Routenplanung nimmt die Selbstkoordination einen hohen Stellenwert ein. In dem Algorithmus 5.2 wird die Selbstkoordination im Sinne einer Auktion zusammengefasst.

Algorithmus 5.2 : Auktionsalgorithmus für SMO-TCM

Eingabe : MANET, Aufgabenliste, Abdeckungsmatrix, Hindernismatrix, Kostenmatrix

Initialisiere: Gebote $B = \emptyset$.
foreach $a_k \in MANET$ **do**
  Ermittle geeignetste Route von der momentanen Position über bereits übernommene Aufgabenziele bis zum Aufgabenziel $\mathfrak{t}_i$
  Ermittle die Kosten der Route
  Ermittle die Eintrittswahrscheinlichkeit für die Route
  Ziehe die Anzahl unbesuchter Zellen von den Kosten ab
  Füge Route als Gebot b_i mit entsprechender Güte der Menge B hinzu
end
Bestimme bestes Gebot $b_i \in B$
Agent a_k mit bestem Gebot b_i übernimmt $\mathfrak{t}_i$ und es wird die Route und das offene Ziel dem Agenten a_k hinzugefügt

5.2. Eine experimentelle Untersuchung des Stochastic Multiobjective-Terrain Covering Modells

Im nachfolgenden wird das zuvor beschriebene SMO-TCM im Rahmen einer Machbarkeitsstudie untersucht. Dazu werden die Untersuchungsgegenstände beschrieben und geeignete Metriken eingeführt. Im Allgemeinen wird der Einfluss und die Auswirkungen auf die Abdeckung, hinsichtlich der Anzahl eingesetzter Agenten sowie Einschränkungen der Kommunikation, verschiedener Umgebungsstrukturen und die Nutzung zusätzlich ermittelter Umgebungsinformationen unter Unsicherheit, untersucht. Entsprechend der Untersuchungsgegenstände werden Experimente definiert und zuvor aufgestellte Hypothesen überprüft. Die Analyse des SMO-TCM wird anhand von zwei Vergleichsmodellen durchgeführt.

5.2.1. Vergleichsmodelle

Neben dem SMO-TCM sind zwei weitere TCMs implementiert worden, welche zur Analyse des TCP unter Unsicherheiten und zur Überprüfung des SMO-TCM herangezogen werden. Für einen direkten Vergleich wird die Routenplanung im SMO-TCM mit dem A*-Algorithmus nach Hart et al. [HNR68] ausgetauscht. Resultierend können die Auswirkungen einer zusätzlichen Betrachtung von approximierten Kosten für das TCP analysiert werden. Das TCM wird im Folgenden als A*-TCM bezeichnet. Des Weiteren ist ein modifizierter Node Counting (NC) Algorithmus implementiert worden.

Der NC Algorithmus unter anderem nach [WLB99, KSL01] weist für das TCP insbesondere zu Beginn der Abdeckung einer unbekannten Umgebung eine hohe Performance hinsichtlich einer Minimierung von Redundanzen auf [SK04]. Nachfolgend werden die Annahmen und Implementierungen des A*-TCM und NC-TCM beschrieben.

Für die Vergleichsmodelle wird ebenfalls eine abstrahierte zweidimensionale Darstellung der Systemumgebung Ω' mit frei befahrbaren und blockierten Zellen z_i vollständig abgedeckt. Die vollständige Abdeckung kann für eine befahrbare Systemumgebung Ω' stets gewährleistet werden, auch wenn zu Beginn der Zustand aller Zellen unbekannt ist. Auf jeder Zelle sind die Strömungskosten $\mathbf{d}_R(z_i)$ eindeutig definiert. Die Systemumgebung unterliegt keinen Änderungen. Hierbei wird die Abdeckung durch autonome Agenten vorgenommen. Alle Entscheidungen werden selbstständig ohne das Einwirken eines menschlichen Bedieners getroffen. Für die Entscheidungsprozesse weisen die Agenten bestimmte Fähigkeiten bezüglich der Wahrnehmung, Kommunikation, Informationsanalyse und Aktionsplanung auf. Die homogenen Agenten verfügen über die gleichen Sensoren und Effektoren. Störungen hinsichtlich der Sensoren und Effektoren treten nicht auf. Informationen können entsprechend der Kommunikationsreichweite innerhalb des MANET ausgetauscht werden. Die Agenten können in jedem Zeitschritt eine Zelle der direkten Nachbarschaft befahren. Die Nachbarschaft enthält alle acht direkt angrenzenden Zellen. Des Weiteren kann ein Agent in jedem Zeitschritt die Nachbarschaft auf die Befahrbarkeit überprüfen. Informationen bezüglich der Strömungskosten können erst beim Besuch einer Zelle bestimmt werden.

5.2.1.1. A*-Terrain Covering Modell

Das A*-TCM orientiert sich stark an der beschriebenen Funktionsweise des SMO-TCM in Kapitel 5.1.3. Um die Auswirkungen einer Berücksichtigung von zusätzlichen Umgebungsinformationen unter Unsicherheiten aufzeigen zu können, betrachtet das A*-TCM eine Routenplanung ausschließlich auf der Grundlage der Entfernung zum Ziel und der unbesuchten Zellen auf der Route. Entsprechend werden die zusätzlichen Strömungsinformationen nicht zur Entscheidungsfindung berücksichtigt. Zur Ermittlung einer Route, welches als Gebot in eine Auktion eingeht, wird der A*-Algorithmus verwendet. Existiert für den Agenten bereits eine noch zu befahrene Route zu bereits übernommenen Zielen, wird die bestehende Route mit der neu zu ermittelnden Route zusammengesetzt. Hierbei wird zuerst die kürzeste Route entweder zwischen der aktuellen Position oder der letzten Zelle der noch zu befahrenden Route und der Zielzelle ermittelt. Anschließend wird die Anzahl der unbesuchten Zellen auf der kürzesten Route bestimmt und von der Länge abgezogen. Resultierend können mehrere Routen die gleichen Gesamtkosten bei unterschiedlicher Länge aufweisen. Gibt es mehrere Gebote beziehungsweise Routen mit minimalen Gesamtkosten gewinnt genau der Agent, dessen Route die kürzere ist. Für den Fall, dass es Routen mit gleicher Kostenstruktur gibt, gewinnt der Agent die Auktion, der zuerst das Gebot abgegeben hat.

Der A*-Algorithmus hat im schlechtesten Fall eine Laufzeit von $O(|V|^2)$ bezüglich der Knoten beziehungsweise Zellen, bezugnehmend auf das TCP. Hierbei basiert der A*-Algorithmus auf einer Bewertungsfunktion $f(z_i)$, die die Kosten von der Startzelle z_S zu z_i und den approximierten Kosten von z_i zur Zielzelle z_G abbildet. Die Kosten entsprechen für das A*-TCM der kürzesten Entfernung, die ein Agent zwischen zwei Zielen bewältigen muss. Demnach wird $f(z_i)$ für z_S und z_G wie folgt angegeben:

$$f(z_i) = g(z_i) + h(z_i) \tag{5.7}$$

mit

$$g(z_i) = \#\text{Vorgängerzellen} \tag{5.8}$$

$$h(z_i) = |x_{z_i} - x_{z_G}| + |y_{z_i} - y_{z_G}| \tag{5.9}$$

Jeder Knoten beziehungsweise jede Zelle z_i wird hierbei zu jedem Zeitschritt einer der drei disjunkten Klassen [unbekannt|bewertet|ausstehend] zugeordnet. Zu Beginn des A*-Algorithmus sind alle Zellen unbekannt, dass heißt die Zellen sind noch nicht untersucht worden. Ferner liegen keine Informationen bezüglich möglicher Routen zur Zielzelle vor. Ist eine Zelle bewertet, dann ist die kürzeste Route zu genau dieser Zelle bereits ermittelt worden. Auf der Suche nach der kürzesten Route werden ausgehend von der derzeit kürzesten Route die angrenzenden Nachbarzellen systematisch untersucht, um gegebenenfalls eine bessere Route zu finden. Die ausstehenden Nachbarzellen werden gemäß der Bewertungsfunktion entsprechend in einer Warteschlange gespeichert und werden für weitere Untersuchungen herangezogen. Der A*-Algorithmus wird nachfolgend in dem Algorithmus 5.3 nach [HNR68] angegeben.

Algorithmus 5.3 : A*-Algorithmus zur Bestimmung der kürzesten Route

Eingabe : Ω'
Ausgabe : Route r_i

Initialisiere: Bestimme $\mathcal{Z}$ auf der Grundlage von Ω', Route $r_i = \emptyset$, Warteschlange $W = \emptyset$.
Füge z_S der Warteschlange W hinzu
while $z_j \neq z_G$ *und* $W \neq \emptyset$ **do**
  Wähle $z_i \in W$ mit besten $f(z_i)$ aus
  z_i =bewertet
  Untersuche alle angrenzende Nachbarzellen $z_n \in \Phi(z_i, t)$ und z_n ist frei befahrbar
    if $z_n \notin W$ **then**
      Ermittle $f(z_n)$ und speichere Vorgängerzelle
  end
    if $z_n \in W$ **then**
      Ermittle $f(z_n)$ neu und aktualisiere gegebenenfalls die Vorgängerzelle
  end
  Sortiere W gemäß $f()$
  $z_j = z_i$
end
if $z_j = z_G$ **then**
  Ermittle r_i anhand der Vorgängerzellen von z_G zu z_S
end
if $z_j \neq z_G$ **then**
  Kein zusammenhängender Graph zwischen z_S und z_G, $r_i = \emptyset$
end

5.2.1.2. Node Counting-Terrain Covering Modell

Das NC-TCM stellt einen intuitiven Ansatz zur Abdeckung einer unbekannten Systemumgebung dar. Zu Beginn der Abdeckung werden alle Agenten in die Systemumgebung platziert sowie die Abdeckungs- und Hindernismatrix mit der Nullmatrix initialisiert. In jedem Zeitschritt wird zuerst

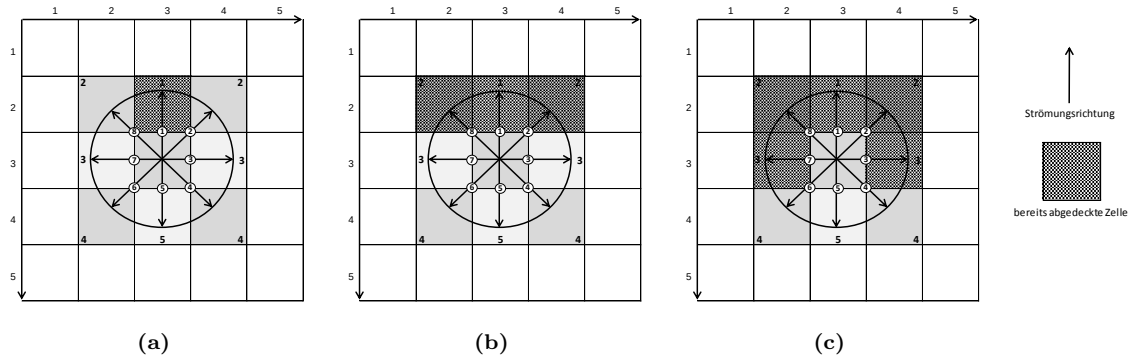


Abbildung 5.2.: Fahrkosten für das Node Counting-Terrain Covering Modell und drei Fälle (a), (b) und (c) für die jeweils zwei Zellen mit gleichem Kostenprofil vorliegen.

die Abdeckungsmatrix aktualisiert. Hierbei wird die aktuelle Position von einem Agenten durch die zugehörige Zelle repräsentiert. Der entsprechende Eintrag in der Abdeckungsmatrix wird um eins erhöht. Entsprechend beschreibt die Abdeckungsmatrix die Anzahl redundanter Abdeckungen auf jeder Zelle. Anschließend ermittelt jeder Agent basierend auf der Kommunikationsreichweite das MANET, in dem Informationen über die Systemumgebung ausgetauscht werden, um ein fusioniertes Lagebild zu erstellen. Dazu wird jeweils eine zusammengefasste Abdeckungs- und Hindernismatrix über die Agenten im MANET ermittelt. Für jeden Eintrag der zusammengefassten Matrix wird jeweils der maximale Eintrag über alle Matrizen $a_k \in$ MANET übernommen. Auf der Grundlage des fusionierten Lagebildes wird die Entscheidung, hinsichtlich der Abdeckung der nächsten Zelle, getroffen. Zuerst werden genau die Nachbarzellen ermittelt, welche befahrbar sind und zugleich eine minimale Abdeckung aufweisen. Weist die ermittelte Menge mehr als eine Zelle auf, werden die zugehörigen approximierten Fahrtkosten bestimmt. Hierbei wird die Strömungsrichtung der Zelle, auf der sich der Agent aktuell befindet, betrachtet. Es wird angenommen, dass die gleiche Strömungsrichtung auch auf den angrenzenden Nachbarzellen vorliegt. Entsprechend der Strömungsrichtung und der resultierenden Fahrtrichtung zu den ermittelten, angrenzenden Zellen werden die Fahrtkosten bestimmt. Aufgrund der modellierten Strömungskosten können maximal zwei Zellen die gleiche minimale Abdeckung und die gleichen minimalen Kosten aufweisen. Liegt dieser Fall vor, wird eine der Zellen zufällig befahren. Mit der Abbildung 5.2 sind die drei Fälle, für die jeweils zwei Zellen mit gleichen Fahrtkosten vorliegen, zusammengefasst. Das in dieser Dissertation modifizierte NC-TCM kann dem Algorithmus 5.4 entnommen werden.

In Untersuchungen [KL01, SK04, MB08] wird gezeigt, dass vor allem zu Beginn der Abdeckung eine hohe Performance der Agenten vorliegen kann. Aufgrund des Greedy-Ansatzes werden redundante Abdeckungen vermieden. Ferner wirkt sich eine minimierte Abdeckungsredundanz grundsätzlich positiv auf den totalen Energieverbrauch der Agenten aus. Der Austausch von Informationen bezüglich bereits abgedeckter Zellen nimmt einen positiven Einfluss auf die totale Explorationszeit. Zugleich können Ineffizienzen zum Ende der Abdeckung auftreten. Aufgrund von Hindernisstrukturen oder aufgrund von zufallsbasierten Auswahlentscheidungen können noch unbesuchte Zellen unter Ineffizienzen abgedeckt werden. Insbesondere wenn unbesuchte Zellen von Zellen mit bereits erheblichen Redundanzen umgeben sind, ist eine optimierte Abdeckung kaum möglich. Hierbei wirkt sich vor allem nachteilig aus, dass nicht alle zur Verfügung stehenden Informationen in die Entscheidungsfindung einbezogen werden.

Algorithmus 5.4 : Modifiziertes Node Counting-Terrain Covering Modell

Eingabe : Ω'
Ausgabe : Vollständig abgedeckte Systemumgebung Ω'

Initialisiere: Bestimme $\mathcal{Z}$ auf der Grundlage von Ω', Agenten $a_k \in A$ mit der Nullmatrix 0_{mn} für die Abdeckungsmatrix und Hindernismatrix
while $\exists z_i \in Z$, *für die gilt: noch nicht abgedeckt und frei befahrbar* **do**
  foreach $a_k \in A$ **do**
    Abdeckungsmatrix(Position$_{a_k}$)=Abdeckungsmatrix(Position$_{a_k}$) $+ 1$
    Bestimme Strömungskosten auf Position$_{a_k}$
  end
  foreach $a_k \in A$ **do**
    Bestimme MANET
    Bestimme fusionierte Abdeckungsmatrix innerhalb des MANET
    Bestimme fusionierte Hindernismatrix
    Bestimme Menge möglicher Zellen $z_j \in Z_{a_k}$ mit

$$z_j \in \Phi_{a_k}(z_i, t), \text{ für die gilt } \min \text{ Abdeckungsmatrix}(z_j)$$

    if $|Z_{a_k}| > 1$ **then**
      Bestimme $z_j \in Z_{a_k}$ mit minimierten Fahrtkosten
    end
    Befahre z_j
  end
end

Neben der Implementierung des modifizierten NC-TCM wird das NCs[2]-TCM als Vergleichsmodell in den nachfolgenden Untersuchungen genutzt. Das NCs-TCM unterscheidet sich nur geringfügig vom NC-TCM. Die Auswahlentscheidung einer Zelle berücksichtigt keine Strömungskosten. Resultierend wird für jeden Agenten die Menge der frei befahrbaren Zellen in der Nachbarschaft ermittelt. Anschließend wird in der Nachbarschaft die niedrigste Abdeckungsredundanz bestimmt. Zellen mit der niedrigsten Abdeckungsredundanz werden einer Menge zugeordnet. Anschließend wird eine Zelle aus der Menge zufällig mittels einer Gleichverteilung bestimmt.

5.2.2. Metriken

Grundsätzlich können drei verschiedene Klassen von Metriken für TCMs unterschieden werden. Neben Metriken der Robustheit sowie der Gebrauchstauglichkeit können die Performance und Effizienz der TCMs analysiert werden, um verschiedene Untersuchungsgegenstände bewerten zu können [MK93]. Die Robustheit untersucht beispielsweise inwiefern das TCM und speziell das Verhalten der Agenten auf falsche Nachrichten oder eine unterbrochene Kommunikation reagiert. Hierbei sollte es den Agenten möglich sein, Konflikte zu identifizieren und aufzulösen, um negative Auswirkungen auf die Robustheit und Performance zu reduzieren. Im gleichen Maße kann die Robustheit in Bezug auf Änderungen der Systemumgebung analysiert werden. Des Weiteren können die Auswirkungen eines Totalausfalles untersucht werden. Für diesen Fall sollte stets eine vollständige Abdeckung garantiert werden können. Die Gebrauchstauglichkeit nimmt vor allem im Sinne der Akzeptanz realer Implementierungen und des Abbaus von Barrieren einen hohen Stellenwert

[2]Engl. Node Counting Stochastic

ein. Die Inbetriebnahme und Bedienbarkeit realer Agenten muss in einem vertretbaren Verhältnis zu der Komplexität der Anwendung stehen. Metriken der Gebrauchstauglichkeit umfassen eher qualitative Bewertungen. Die Anzahl anwendungs- und umgebungsspezifischer Anpassungen der Roboter sowie die Notwendigkeit zusätzlicher Informationen vor einer Abdeckung wirken sich negativ aus. Des Weiteren sollte die Benutzerschnittstelle ein hohes Maß an einfacher und intuitiver Bedienbarkeit aufweisen [MK93].

Das SMO-TCM wird im Rahmen der Machbarkeitsstudie anhand der nachfolgenden Metriken der Performance und Effizienz überprüft. Hierbei wird mit der Performance die Leistungsfähigkeit der Agenten und somit der Grad der übergeordneten Zielerreichung gemessen. Hingegen analysieren die Effizienzmetriken die Wirtschaftlichkeit der Zielerreichung. Für TCMs kann die Zielerreichung verschiedene Ausprägungen annehmen. Es können beispielsweise die totale Abdeckungs- und Explorationszeit, der totale Energieverbrauch, die Gesamtanzahl redundanter Abdeckungen oder die Gesamtanzahl von Kurs- und Richtungsänderungen der Agenten gemessen werden. Insbesondere bei der mehrkriteriellen Optimierung können konträre Zieldefinitionen vorliegen, die eine umfassende Analyse der Untersuchungsgegenstände erfordert.

- **Totale Abdeckungszeit/totale Explorationszeit:**
  Die totale Abdeckungszeit misst die Anzahl der Zeitschritte, bis eine erstmalige Abdeckung der Systemumgebung vorliegt. Streng genommen muss die gesamte Systemumgebung durch Agenten befahren worden sein. Mittels Sensoren ist eine Untersuchung im Nah- und Fernbereich der Agenten möglich. Entsprechend kann gemäß der übergeordneten Zieldefinition eine Abdeckung der Systemumgebung mit Sensoren ausreichend sein. Diese Form der Abdeckung wird einer vollständigen Exploration der Systemumgebung gleichgesetzt, um beispielsweise eine Kartographie der Systemumgebung zu ermöglichen [MM08]. Für diesen Fall ist in der Regel die totale Explorationszeit kleiner gleich der zugehörigen Abdeckungszeit. Mit zunehmender Sensorenreichweite nimmt die Explorationszeit tendenziell ab.

- **Prozentuale Abdeckung:**
  Die inverse Metrik der prozentualen Abdeckung bezüglich der Zeit wird genutzt, um den Verlauf einer Abdeckung beziehungsweise einer Exploration bewerten zu können [WM04]. Besonders schwer abzudeckende Regionen, beispielsweise konkave Hindernisstrukturen oder Verengungen können zu einer Verzögerung im Fortschritt der Abdeckung führen [SAB$^+$00, ZJKK05]. Diesbezüglich kann zudem der Verlauf einer Abdeckung untersucht werden. Eine optimale Abdeckung zeigt einen linearen Verlauf mit einem Anstieg proportional zur Anzahl der Agenten. Für online TCMs liegt häufig ein logarithmischer Verlauf der Abdeckung vor. Entsprechend nimmt im Verlauf die Zeit zur Abdeckung einzelner Regionen zu.

- **Abdeckungsverteilung:**
  Die Metrik der Abdeckungsverteilung gehört der Klasse der Effizienzmetriken an und misst die räumliche Verteilung von redundanten Abdeckungen. Mit Hilfe dieser Metrik können Regionen identifiziert werden, die im Verlauf der Abdeckung besondere Herausforderungen dargestellt und somit zu einer redundanten Abdeckung geführt haben. Die Agentenanzahl sollte keinen Einfluss auf die Abdeckungsverteilung nehmen [BM08]. Die Summe über die Anzahl redundanter Abdeckungen quantifiziert die Effizienz und ermöglicht einen Vergleich zwischen verschiedenen TCMs.

Für eine quantitative Messung der Abdeckungsqualität wird die nachfolgende Metrik eingeführt:

$$G_\Lambda = \sum_{i=1}^{\max \text{Mehrfachabdeckung}} i * (\#\text{Zellen, die } i - \text{Abdeckungen aufweisen}) \tag{5.10}$$

Mit max Mehrfachabdeckung wird die maximale Abdeckung über alle Zellen angegeben. Demnach nimmt G_Λ einen besseren Wert für Abdeckungen mit wenig Redundanzen an.

- **Totaler Energieverbrauch:**
  Der totale Energieverbrauch misst den Gesamtenergieverbrauch aller Agenten, der für eine vollständige Abdeckung benötigt wird. Neben Redundanzen bezüglich der Abdeckung können sich weitere Eigenschaften der Systemumgebung negativ auf den Energieverbrauch der Agenten auswirken. Die Systemumgebung kann beispielsweise Höhenunterschiede, verschiedene Bodenbeschaffenheiten oder Strömungen aufweisen. Es muss eine Berücksichtigung dieser Eigenschaften erfolgen, wenn der Energieverbrauch optimiert werden soll. Es ist zu beachten, dass Redundanzen unter Umständen zu einem verbesserten Energieverbrauch führen können. Ein positiver Einfluss auf den Energieverbrauch liegt genau dann vor, wenn alternative Routen vorliegen, die länger sind als die kürzeste Route, aber einen geringeren Energieverbrauch aufweisen. Zudem kann sich ein gleichmäßiger Energieverbrauch positiv auf die übergeordnete Zielerreichung auswirken, weil die Agenten länger im Verband zusammen agieren können. Resultierend stehen gegebenenfalls spezifische Effektoren und Sensoren länger zur Verfügung.

5.2.3. Untersuchungsgegenstände und Hypothesen

In der nachfolgenden Durchführung und Auswertung des eingeführten SMO-TCM stehen Performance- und Effizienzuntersuchungen sowie Untersuchungen der Robustheit hinsichtlich einer vollständigen Abdeckung einer unbekannten Systemumgebung im Mittelpunkt. Diesbezüglich werden verschiedene Untersuchungsgegenstände bewertet, um die Auswirkungen auf die Abdeckung analysieren zu können. Die in dieser Dissertation betrachteten Untersuchungsgegenstände werden nachstehend eingeführt und zugehörige Hypothesen erläutert.

- **Anzahl eingesetzter Agenten:**
  In jedem Zeitschritt kann ein Agent eine benachbarte, freie Zelle besuchen. Mit einer steigenden Anzahl von Agenten können zeitgleich mehr Zellen besucht werden. Hierbei bestimmen die übergeordneten Ziele in welcher Reihenfolge die Zellen besucht werden. Im Rahmen der Minimierung der totalen Explorationszeit sollte sich demnach eine zunehmende Anzahl von Agenten positiv auswirken. In diesem Zusammenhang ist zu untersuchen, welchen Einfluss die Anzahl der Agenten auf die Abdeckungsverteilung nimmt. Tendenziell können sich Ineffizienzen in der Abdeckung kumulieren. Wird hingegen die Minimierung des totalen Energieverbrauches betrachtet, können Redundanzen in der Abdeckung nur bedingt Hinweise auf die Effizienz geben. Aufgrund von unterschiedlichen Kosten, die mit dem Befahren einer Zelle verbunden sind, können kostengünstigere Routen stärker frequentiert sein. Resultierend müssen die Kostenstruktur der Systemumgebung und die ermittelten Routen analysiert werden.

- **Kommunikationsreichweite:**
  In jedem Zeitschritt tauschen die Agenten die Gesamtheit aller Informationen in dem zugehörigen MANET aus. Hierbei bestimmt sich die Größe des MANET maßgeblich anhand der Kommunikationsreichweite. Je mehr Informationen ausgetauscht werden können, desto umfassender ist das Gesamtlagebild. Auf der Grundlage des Gesamtlagebildes werden die Entscheidungen bezüglich der weiteren Abdeckung geplant. Resultierend sollte sich eine umfangreiche Entscheidungsgrundlage positiv auf die Routenplanung und Aufgabenzuteilung auswirken. Ferner sollte sich die übergeordnete Zielerreichung verbessern.

- **Systemumgebung:**
  Die Ausprägung der Systemumgebung kann aufgrund von Hindernis- und Kostenstrukturen einen Einfluss auf die Routenplanung und Aufgabenzuteilung nehmen. Konvexe Hindernisse können durch ein Umfahren vollständig erkundet werden. Hingegen ist die vollständige Erkundung von konkaven Hindernissen schwieriger. Insbesondere bei verengten Zugängen kann eine Exploration und Abdeckung Ineffizienzen aufweisen, wenn der Zugang nicht als solcher erkannt wird und resultierend das weitere Vorgehen nicht geeignet ausgerichtet wird [SAB+00, ZJKK05]. Um die Auswirkungen auf die Abdeckung bewerten zu können, werden exemplarisch drei verschiedene Systemumgebung in der Literatur [SAB+00, ZJKK05] vorgeschlagen: eine freie Systemumgebung, eine Systemumgebung mit konkaven (outdoor-like) und eine mit konvexen (indoor-like) Hindernisstrukturen.

Grundsätzlich wird die Abdeckung einer unbekannten Systemumgebung auf der Grundlage von Entscheidungen unter Unsicherheiten durchgeführt. Möglichst robuste Entscheidungen können sich positiv auf die Performance und Effizienz einer Abdeckung auswirken. Zusätzlich integriert das SMO-TCM weitere Methoden, um ein robustes Verhalten der Agenten ausprägen zu können. Aus diesen Gründen wird das Verhalten der Agenten im Umgang mit Unsicherheiten bewertet.

5.2.4. Experimentaufbau

Nachfolgend wird das in Kapitel 5.1 vorgestellte SMO-TCM hinsichtlich der Untersuchungsgegenstände aus Kapitel 5.2.3 mittels einer experimentellen Überprüfung untersucht. Hierzu werden Simulationsexperimente definiert und die Auswirkungen auf die Untersuchungsgegenstände anhand der zuvor in Kapitel 5.2.2 beschriebenen Metriken analysiert. Ferner werden die Auswirkungen auf die Performance und Zielerreichung des SMO-TCM mit den Ergebnissen des A*-TCM und NC-TCM sowie NCs-TCM verglichen. Insbesondere werden die Vorteile und Realisierbarkeit einer Integration des transformierten SSPP aus Kapitel 4 im Rahmen der Machbarkeitsstudie für das TCP untersucht.

Die TCMs werden anhand von drei charakteristischen Systemumgebungen überprüft. Die freie Systemumgebung, eine outdoor-like Systemumgebung mit konvexen Hindernisstrukturen sowie eine indoor-like Systemumgebung mit konkaven Hindernisstrukturen, die im Rahmen der experimentellen Überprüfung verwendet werden, sind in Abbildung 5.3 (a) bis (c) veranschaulicht. Die Systemumgebungen weisen eine Ausdehnung von 32 mal 32 Zellen auf. Demnach sind in der freien Systemumgebung insgesamt 1024 Zellen abzudecken. Für die outdoor-like sowie indoor-like Systemumgebung müssen abzüglich der Hinderniszellen jeweils 880 Zellen abgedeckt werden. Die

Experiment	Konstante Einflussgrößen	Untersuchungsgegenstand
#Agenten	Systemumgebung=frei, Kommunikationsreichweite=32	#Agenten=[1|3|5]
Kommunikations-reichweite	Systemumgebung=frei, #Agenten=5	Kommunikationsreichweite=[1|4|8|32]
Systemumgebung	Kommunikationsreichweite=32, #Agenten=5	Umgebungstypen=[frei|outdoor|indoor]

Tabelle 5.1.: Experimentaufbau für eine experimentelle Überprüfung der Terrain Covering Modelle.

Startzelle für alle Agenten in jedem Szenario ist $z_S = (0, 0)$. Zusätzlich sind auf jeder Zelle die Strömungskosten $\mathbf{d}_R(z_i)$ für alle Szenarien gemäß der Abbildung 5.3 (d) und (e) definiert.

Für das A*-TCM und das NC-TCM werden jeweils 30 Wiederholungen für jedes Experiment durchgeführt. Für das SMO-TCM werden lediglich 10 Wiederholungen vorgenommen. Die Ausführung einer Wiederholung des SMO-TCM beträgt in Abhängigkeit der Anzahl eingesetzter Agenten in etwa zwischen 3 bis 24 Stunden. Die hohe Laufzeit liegt aufgrund des integrierten BiCriterionAnt Algorithmus zur Ermittlung einer Route vor, der wiederholt im Rahmen eines auktionsbasierten Algorithmus durchgeführt werden. Hierbei werden Routen ermittelt, die sich als weniger effizient herausstellen und somit verworfen werden müssen. Der BiCriterionAnt Algorithmus wird entsprechend der Voruntersuchung mit der zuvor ermittelten Parameterkonfiguration durchgeführt:

$$\begin{aligned}\text{Parameterkonfiguration} = (&\max_{Generation} = 25; |A| = 30; \alpha = 1.55; \beta = 1.25;\\ &\varrho_{ANZ} = 36; \varrho_E = 0.34; \rho_{\min} = 0; \rho_{\max} = 100; \rho_{\text{ini}} = 50)\end{aligned}$$

Die experimentelle Überprüfung wird auf einem PC-Cluster mit 40 CPUs (jeweils Intel Xeon mit 2,3 Gigahertz) durchgeführt. Hierbei werden die in Tabelle 5.1 aufgeführten Experimente durchgeführt. Zur Beschreibung eines Experiments wird die Schreibweise

Experiment: [Systemumgebung|Kommunikationsreichweite|#Agenten]

eingeführt.

5.2.5. Analyse und Auswertung

Mit der Analyse wird das Ziel verfolgt, die Auswirkungen einer robusten Routenplanung im Sinne des SSPP für das TCP zu untersuchen. Hierbei steht das SMO-TCM im Mittelpunkt der Betrachtung und wird anhand der Vergleichsmodelle gegenübergestellt. Gemäß dem Experimentaufbau werden drei Untersuchungsgegenstände: die Anzahl der Agenten, die Kommunikationsreichweite und die Struktur der Systemumgebung analysiert. Ferner werden insbesondere die Auswirkungen auf die totale Explorationszeit, die Abdeckungsredundanz und den totalen Energieverbrauch beobachtet. Die Detailergebnisse sowie die Einzelexperimente sind der CD-ROM beigefügt. Falls nicht anders angegeben, stellen die Ergebnisse die Mittelwerte über alle Experimentdurchläufe dar. Die TCMs werden in den Abbildungen der Analyse wie folgt bezeichnet:

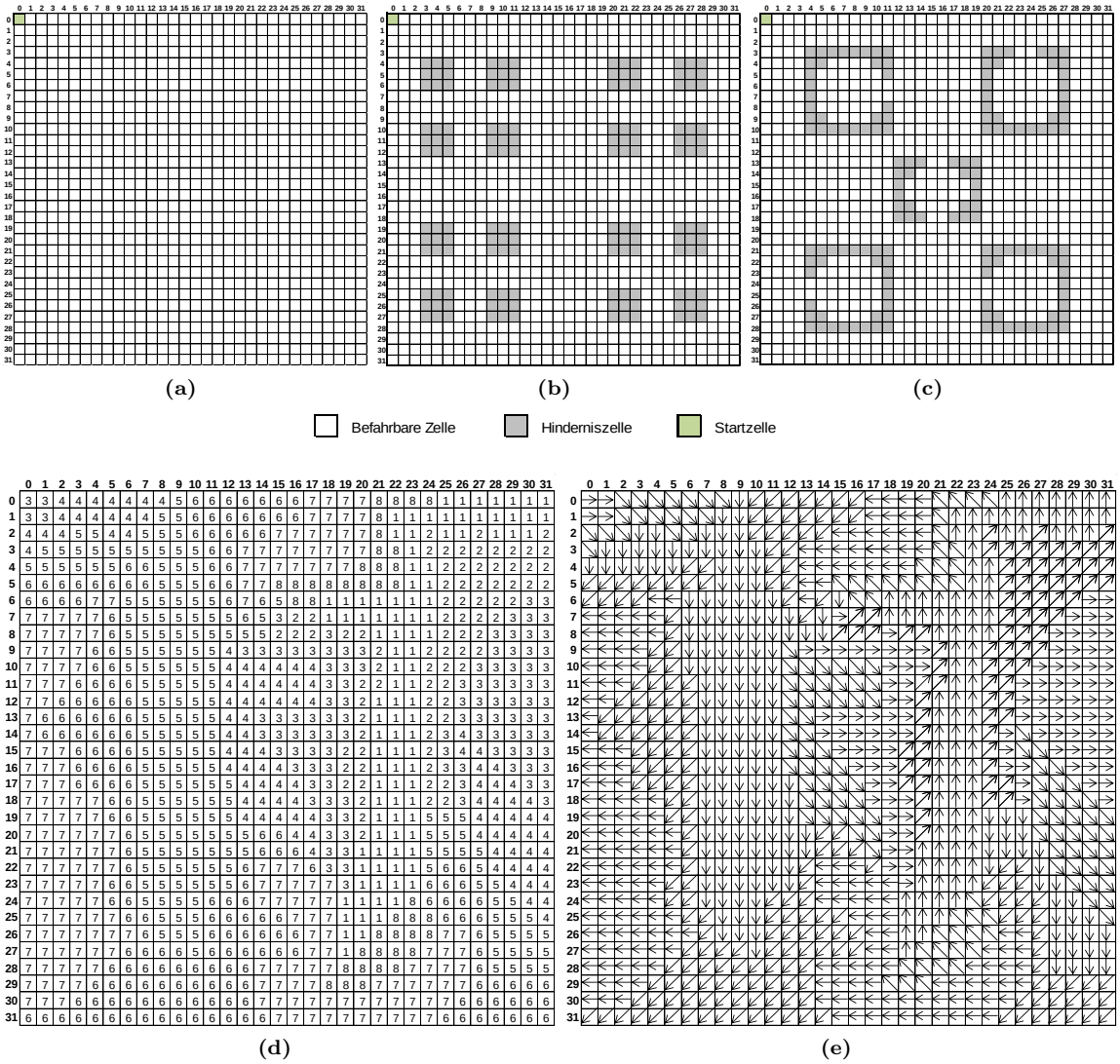


Abbildung 5.3.: Experimentinitialisierung für die Terrain Covering Modelle. Unter (a) freie Systemumgebung, (b) outdoor-like Systemumgebung mit konkaven Hindernisstrukturen, (c) indoor-like Systemumgebung mit konvexen Hindernisstrukturen, (d) kodierte Strömungsrichtungen auf den Zellen und (e) vereinfachte Darstellung der Strömungsrichtungen auf den Zellen.

- A - Quadtree TCM mit Routenplanung ohne Unsicherheitsbetrachtung (A*-TCM)
- SMO - Quadtree TCM mit Routenplanung unter Unsicherheitsbetrachtung (SMO-TCM)
- NC - Node Counting TCM mit Kostenapproximation (NC-TCM)
- NCs - Node Counting TCM ohne Kostenapproximation

Nachfolgend wird die Analyse der Untersuchungsgegenstände, eine detaillierte Darstellung der Exploration aller vier TCMs anhand der freien Systemumgebung mit vollständiger Kommunikation und fünf Agenten [frei|32|5] vorangestellt.

Die Experimente zeigen, dass für das A*-TCM und NC-TCM im Rahmen der Experimente ein deterministisches Verhalten vorliegt. Für das A*-TCM erfolgt ausschließlich für den Fall, dass mehrere Routen mit minimalen, approximierten Kosten in einer Auktion ermittelt worden sind, eine zufällige Auswahl der zu befahrenden Route. Dieser Fall tritt in den Experimenten nicht ein, stattdessen wird stets nur eine eindeutige Route mit minimalen Kosten ermittelt. Für das NC-TCM erfolgt die Auswahl der nächsten Zelle in jedem Zeitschritt anhand von zwei Entscheidungsgrundlagen. Zuerst wird die Menge von Zellen mit minimaler Anzahl redundanter Zellabdeckungen in der Nachbarschaft ermittelt. Daran anschließend werden aus dieser Menge genau die Zellen ermittelt, welche minimale, approximierte Kosten aufweisen. Hierbei können gemäß den Darstellungen in Abbildung 5.2 drei Fälle identifiziert werden, bei denen jeweils aus zwei Zellen zufallsbedingt ausgewählt werden muss. Keiner dieser drei Fälle tritt in den Experimenten ein. Aus diesem Grund wird die Approximation der Kosten in dem NCs-TCM nicht berücksichtigt. Es erfolgt die Bestimmung der nächsten Zelle allein auf der Menge von Zellen mit minimaler Abdeckungsredundanz und anschließender zufälliger Auswahl einer dieser Zellen.

In Abbildung 5.4 werden die Explorationsverläufe hinsichtlich der totalen Explorationszeit dargestellt. Bezüglich der totalen Explorationszeit, zeigt das NCs-TCM für die ersten 90 % der Abdeckung die beste Performance. Daran anschließend steigen die Ineffizienzen erheblich an. Für die letzten 10 % der Abdeckung wird nahezu die gleiche Zeit benötigt, wie für die ersten 90 %. Ursache hierfür sind die fehlenden Positionsinformationen hinsichtlich der noch abzudeckenden Zellen. In einem negativen Fall kann eine noch abzudeckende Zelle von einer Vielzahl von bereits bekannten Zellen eingeschlossen sein, die zudem eine hohe Abdeckungsredundanz aufweisen. Resultierend muss für eine Abdeckung der noch unbekannten Zellen zuvor eine Angleichung der Abdeckungsredundanz aller anderen Zellen erfolgen, damit die Möglichkeit einer Abdeckung besteht. Der beschriebene Fall führt zu erheblichen Ineffizienzen. Vergleichbare Beobachtungen zu dem NCs-TCM sind bereits in [KL01, SK04] beschrieben worden. Ein anderes Explorationsverhalten ist für das sehr ähnliche NC-TCM zu beobachten. Anders als für das NCs-TCM ist die Abdeckung abhängig von den Strömungsinformationen auf den Zellen. Es kann beobachtet werden, dass die Agenten zuerst die Explorationen in Richtung der Zellen konzentrieren, die voraussichtlich zu geringeren Kosten bezüglich der Abdeckung führen werden. Zudem sind höhere Abdeckungsredundanzen zu Beginn der Abdeckung zu verzeichnen. Die Schwierigkeiten bei der Abdeckung der letzten 10 % der Systemumgebung können hingegen für das NC-TCM auf der untersuchten Kostenstruktur nicht beobachtet werden. Bezüglich der totalen Explorationszeit zeigen das SMO-TCM und das A*-TCM dem NC-TCM ähnliche Explorationsverläufe. Aufgrund der Positionsinformationen noch abzudeckender Zellen können die Agenten zielgerichteter eine Exploration der letzten 10 % der Zellen vornehmen.

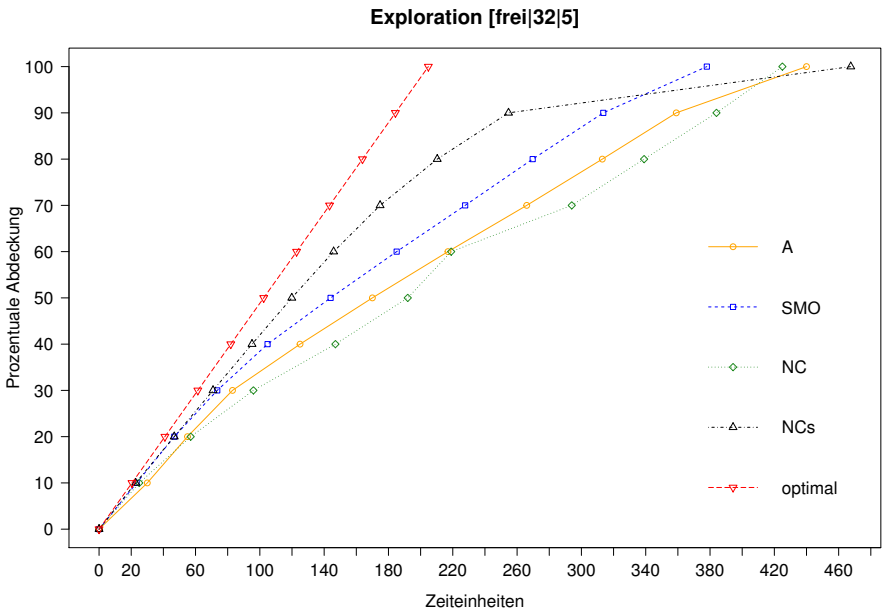


Abbildung 5.4.: Terrain Covering Modelle - Explorationsverlauf mit [frei|32|5]. Darstellung der Mittelwerte über alle Experimentdurchläufe.

Die totale Explorationszeit korreliert mit den Abdeckungsredundanzen. Je mehr Zellen mehrfach abgedeckt werden, desto mehr nimmt die totale Explorationszeit zu. In Abbildung 5.5 können die Abdeckungsredundanzen der TCMs entnommen werden. Für das SMO-TCM und A*-TCM zeigt sich, dass die Anzahl der Zellen mit zunehmender Mehrfachabdeckung kontinuierlich abnehmen. Es liegen zudem Zellen vor, die mehr als fünfmal abgedeckt worden sind. Für das A*-TCM resultieren die Mehrfachabdeckungen vor allem aus der Aufgabenreihenfolge. Dadurch müssen die Agenten das Zentrum der Systemumgebung stärker befahren. Das SMO-TCM kann aufgrund der optimierten Routenplanung dieses Phänomen reduzieren. Hingegen werden im SMO-TCM genau die Zellen stärker frequentiert, die für mehrere Routen eine günstige Kostenstruktur aufweisen. Grundsätzlich ist zu erkennen, dass für das SMO-TCM und A*-TCM die Anzahl der Zellen mit zunehmenden Abdeckungsredundanzen exponentiell abnehmen. Für das SMO-TCM liegt im Mittel eine optimale Einfachabdeckung für 43 % aller Zellen vor. Insgesamt werden 71 % der Zellen höchstens zweimal besucht. Für das NC-TCM und NCs-TCM zeigt sich, dass die Anzahl der Zellen mit zwei und drei Befahrungen höher ist als für das SMO-TCM und A*-TCM. Zudem nimmt für das NC-TCM der Anteil der Zellen, die zwei- oder dreimal befahren worden sind, einen Anteil von 57 % ein. Für das NCs-TCM liegt dieser Anteil mit 85 % am höchsten. Zusammenfassend gilt für die nachfolgenden Abdeckungen der Experimentierung [frei|32|5]:

$$G_\Lambda(\text{SMO-TCM})=1849 < G_\Lambda(\text{NC-TCM})=2130 < G_\Lambda(\text{A*-TCM})=2163 < G_\Lambda(\text{NCs-TCM})=2343$$

Mit der Abbildung 5.6 wird der Explorationsverlauf hinsichtlich redundanter Abdeckungen dargestellt. Grundsätzlich ähneln die Verläufe dem zeitbezogenen Explorationsverlauf in Abbildung 5.4. Aufgrund der Vorgehensweise im NCs-TCM werden die Vorteile hinsichtlich einer redundanten Abdeckung deutlich. Bis zu einem Explorationsfortschritt von 70 % treten verhältnismäßig wenig Ineffizienzen hinsichtlich Mehrfachabdeckungen auf. Erst im letzten Drittel häufen sich die Ineffizienzen. Für die letzten 10 % der Exploration treten im Durchschnitt mehr als dreimal sovie-

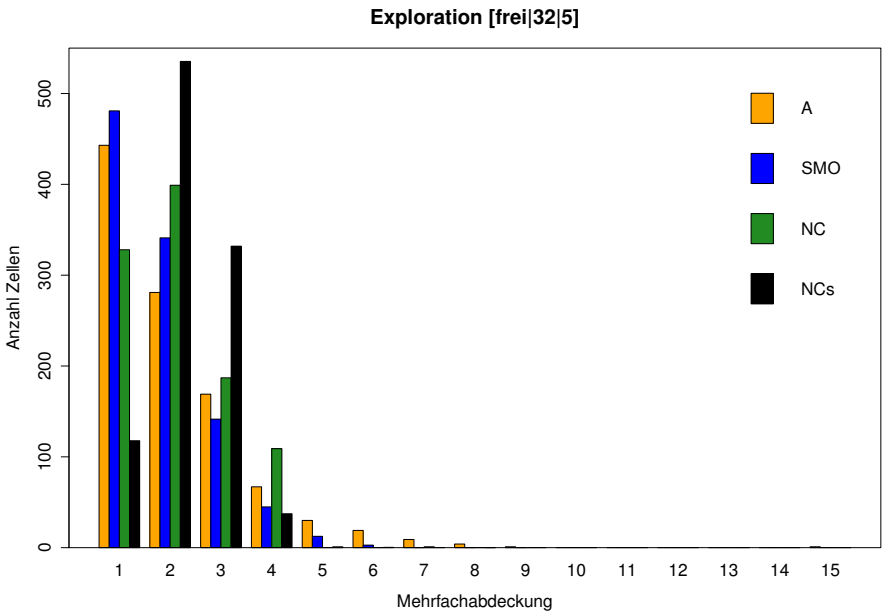


Abbildung 5.5.: Terrain Covering Modelle - Zellen mit Mehrfachabdeckungen mit [frei|32|5]. Darstellung der Mittelwerte über alle Experimentdurchläufe.

le redundante Abdeckungen auf, bezogen auf die gesamten ersten 90 % der Exploration. Hierbei übertrifft die Gesamtanzahl der Mehrfachabdeckungen des NCs-TCM jeweils die Mehrfachabdeckungen aller anderen TCMs. Für das SMO-TCM treten ebenfalls geringe Mehrfachabdeckungen im ersten Drittel der Exploration auf. Danach nehmen die Mehrfachabdeckungen linear mit dem Explorationsfortschritt zu. Für die letzten 10 % der Abdeckung nehmen die Ineffizienzen ebenfalls zu. Für das A*-TCM ist der gleiche Verlauf zu beobachten. Ferner treten zu jedem Zeitpunkt mehr Ineffizienzen auf. Das NC-TCM zeigt eine Besonderheit bezüglich der redundanten Abdeckungen. Für die letzten 10 % der Exploration ist kein Anstieg von Ineffizienzen zu beobachten.

Die Lage der Zellen mit Mehrfachabdeckungen ist in den Abbildungen 5.7, 5.8, 5.9 und 5.10 für die Explorationsfortschritte 25 %, 50 %, 75 % und 100 % dargestellt. Es zeigt sich für das SMO-TCM und A*-TCM, dass die Zellen mit einer höheren Mehrfachabdeckung tendenziell eher im Zentrum liegen. Ferner kann die Aufgabenreihenfolge resultierend aus der Quadtree-Zerlegung erkannt werden. Die Abdeckung schreitet ausgehend von den Zentren der vier Quadranten auf Ebene eins voran. Des Weiteren unterscheiden sich die räumlichen Verteilungen der Zellen mit Mehrfachabdeckungen für das NC-TCM und NCs-TCM. Für die untersuchte Systemumgebung kann das NC-TCM eine gleichmäßigere Abdeckung erzeugen. Hierbei liegen tendenziell zusammenhängende Regionen mit gleicher Mehrfachabdeckung vor. Es kann vermutet werden, dass die Ausprägung der Strömungsinformation einen signifikanten Einfluss auf die Gleichmäßigkeit der Abdeckung nimmt.

Neben der totalen Explorationszeit und der Abdeckungsverteilung muss der totale Energieverbrauch analysiert werden. Hierbei kann von der totalen Explorationszeit und der Anzahl von Mehrfachabdeckungen nur bedingt auf den totalen Energieverbrauch geschlossen werden. Hierfür kann ein einfaches Beispiel gemäß der Abbildung 5.11 konstruiert werden. Werden in dem gegebenen Beispiel ausschließlich die Kosten betrachtet, dann resultieren gemäß der gegebenen Strömungsinformationen für die Route (1) Kosten in Höhe von 44 Einheiten und für die Route (2)

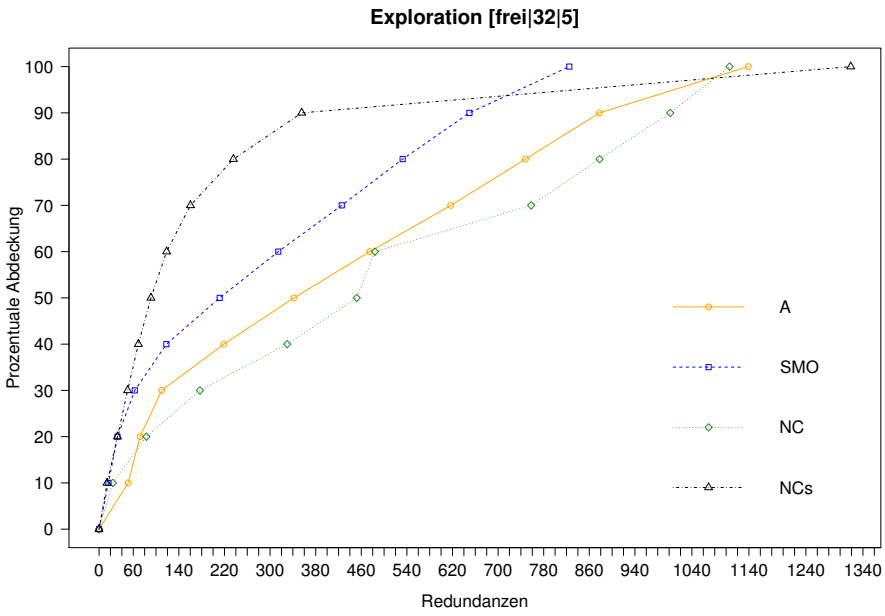


Abbildung 5.6.: Terrain Covering Modelle - Verlauf der redundanten Zellabdeckungen mit [frei|32|5]. Darstellung der Mittelwerte über alle Experimentdurchläufe.

entsprechend 13 Einheiten. Jedoch werden mit der Route (1) zusätzlich neun unbekannte Zellen und mit der Route (2) lediglich eine unbekannte Zelle abdeckt. Mit dieser Einzelfallbetrachtung und der Auswahl der Route (2) wird jedoch nicht berücksichtigt, ob der Kostenvorteil im weiteren Verlauf der Abdeckung nicht sogar hinfällig wird, insofern mehr Kosten aufgewendet werden müssen, um die nun noch unbekannten Zellen abzudecken.

Mit der Abbildung 5.12 ist der totale Energieverbrauch bezüglich des Explorationsfortschritts veranschaulicht. Wie schon die Verläufe der Redundanzen ähneln die Verläufe der Energieverbräuche dem des zeitbezogenen Explorationsverlaufes. Zu Beginn der Exploration liegen die Energieverbräuche für alle TCMs sehr nah beieinander. Die Vorteile einer Berücksichtigung der Strömungsinformationen, wirkt sich aufgrund der zu diesem Zeitpunkt kaum vorhanden Informationen nicht auf den Energieverbrauch auf. Der direkte Vergleich zwischen dem A*-TCM und SMO-TCM zeigt, dass die Energieeinsparung des SMO-TCM im Verlauf der Exploration zunehmen. Des Weiteren ist zu erkennen, dass die Effizienzvorteile des NCs-TCM hinsichtlich der Redundanzen für den Energieverbrauch nicht so stark zu treffen. Für die Redundanzen liegt der größte Effizienzvorteil zu einer Zeit vor, in denen 70 % bis 80 % der Systemumgebung abgedeckt sind. Zu diesem Explorationsfortschritt liegen für das NCs-TCM im Vergleich zum SMO-TCM lediglich 37 % der redundanten Abdeckungen vor. Zur gleichen Zeit liegt der Energieverbrauch für das NCs-TCM bei 80 % beziehungsweise 81 % des SMO-TCM Energieverbrauches. Demnach ist zu erkennen, dass sich eine Berücksichtigung der Strömungsrichtung positiv auf den Energieverbrauch auswirken kann. Jedoch ist ebenfalls anzumerken, dass bis zu einem hohen Explorationsfortschritt die Vorteile einer energieberücksichtigenden Exploration die einer Abdeckung mit wenigen Redundanzen nicht ausgleichen können.

In Abbildung 5.13 ist über den gesamten Explorationsverlauf das Verhältnis zwischen der Abdeckung unbekannter Zellen und den Kosten (Energieverbrauch) dargestellt. Im günstigsten Fall können fünf Agenten in einem Zeitschritt fünf unbekannte Zellen abdecken und müssen dafür Kos-

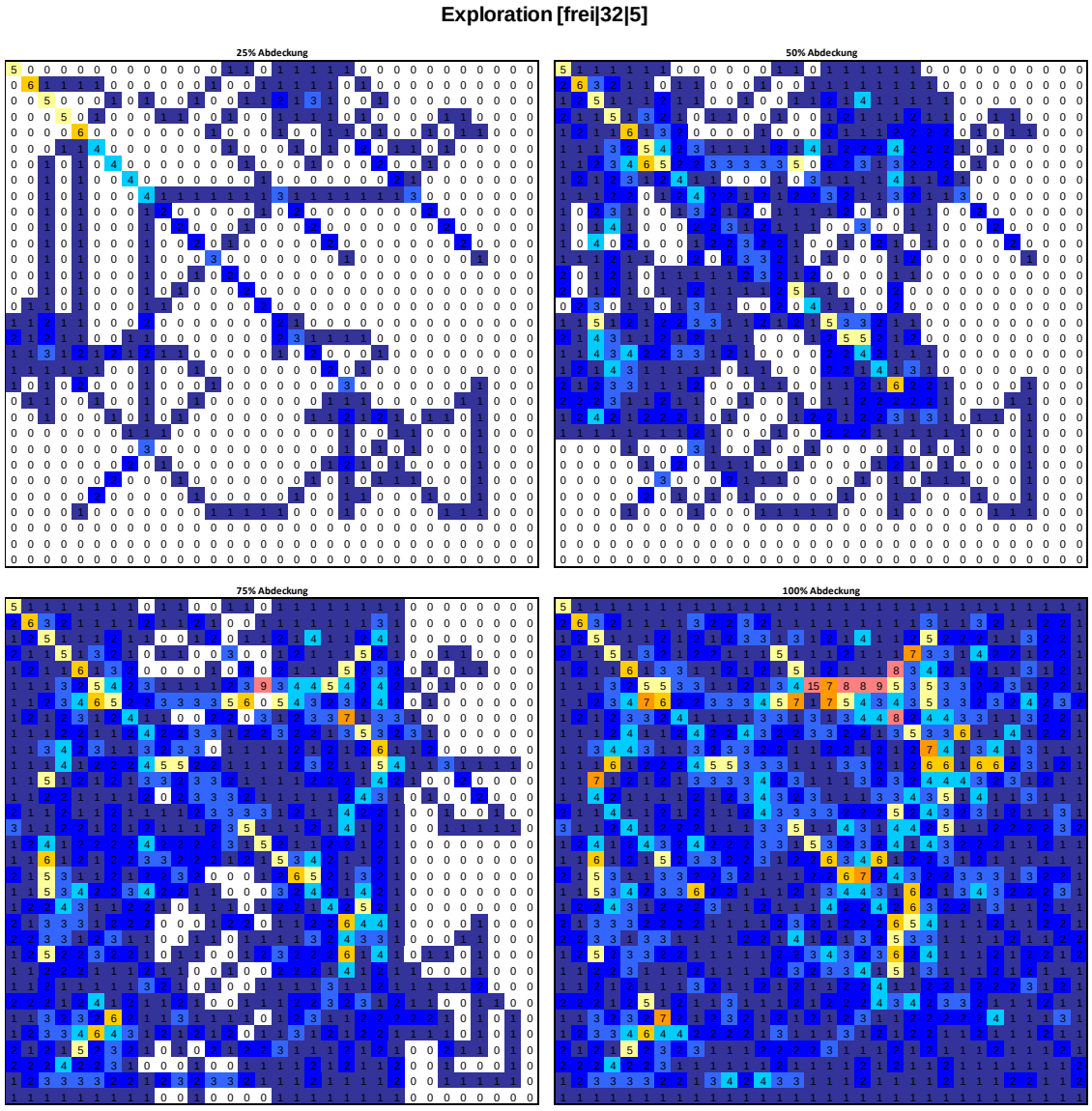


Abbildung 5.7.: A* Terrain Covering Modell - Abdeckung mit [frei|32|5]. Darstellung des besten Experimentdurchlaufes.

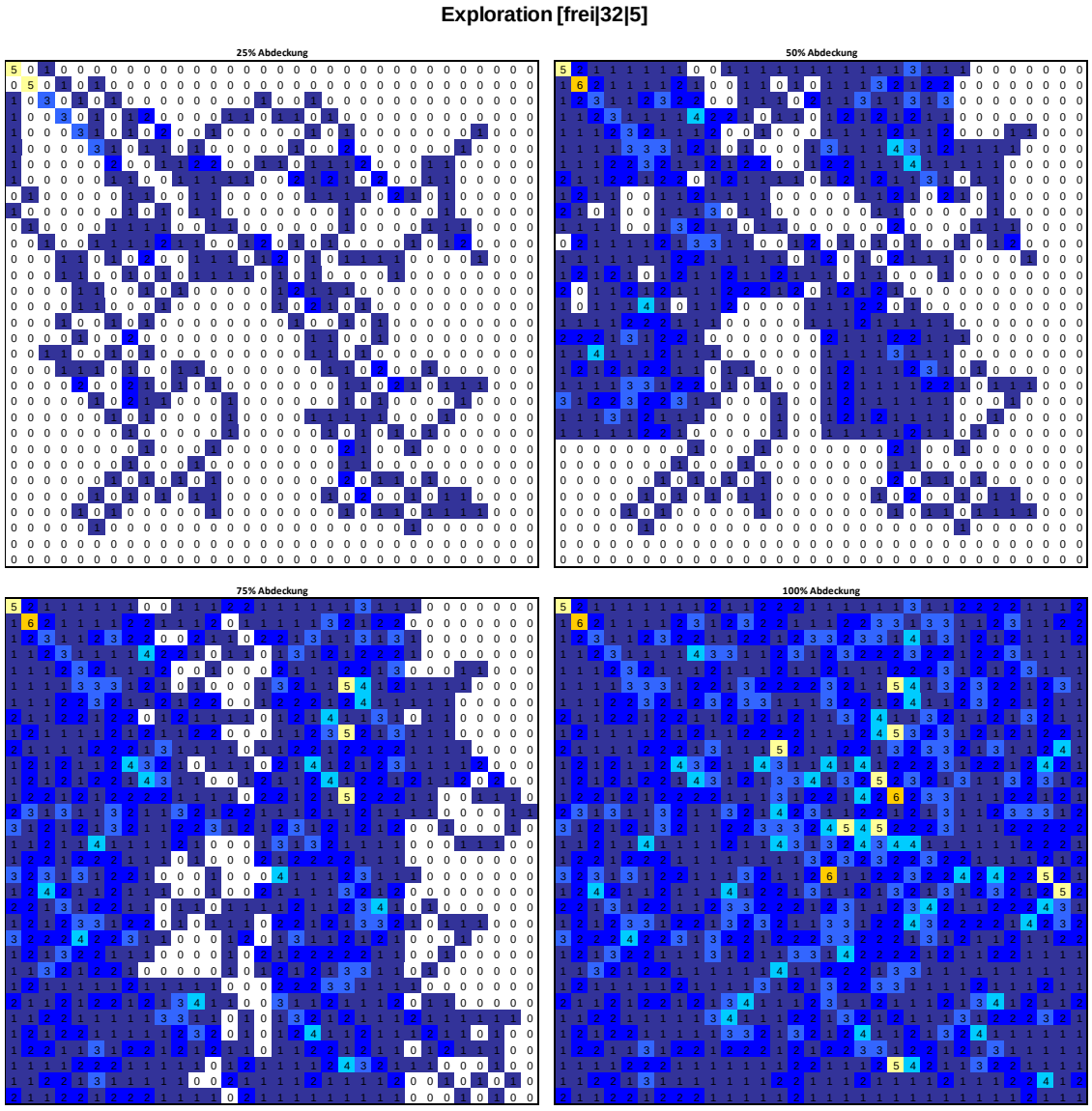


Abbildung 5.8.: SMO Terrain Covering Modell - Abdeckung mit [frei|32|5]. Darstellung des besten Experimentdurchlaufes.

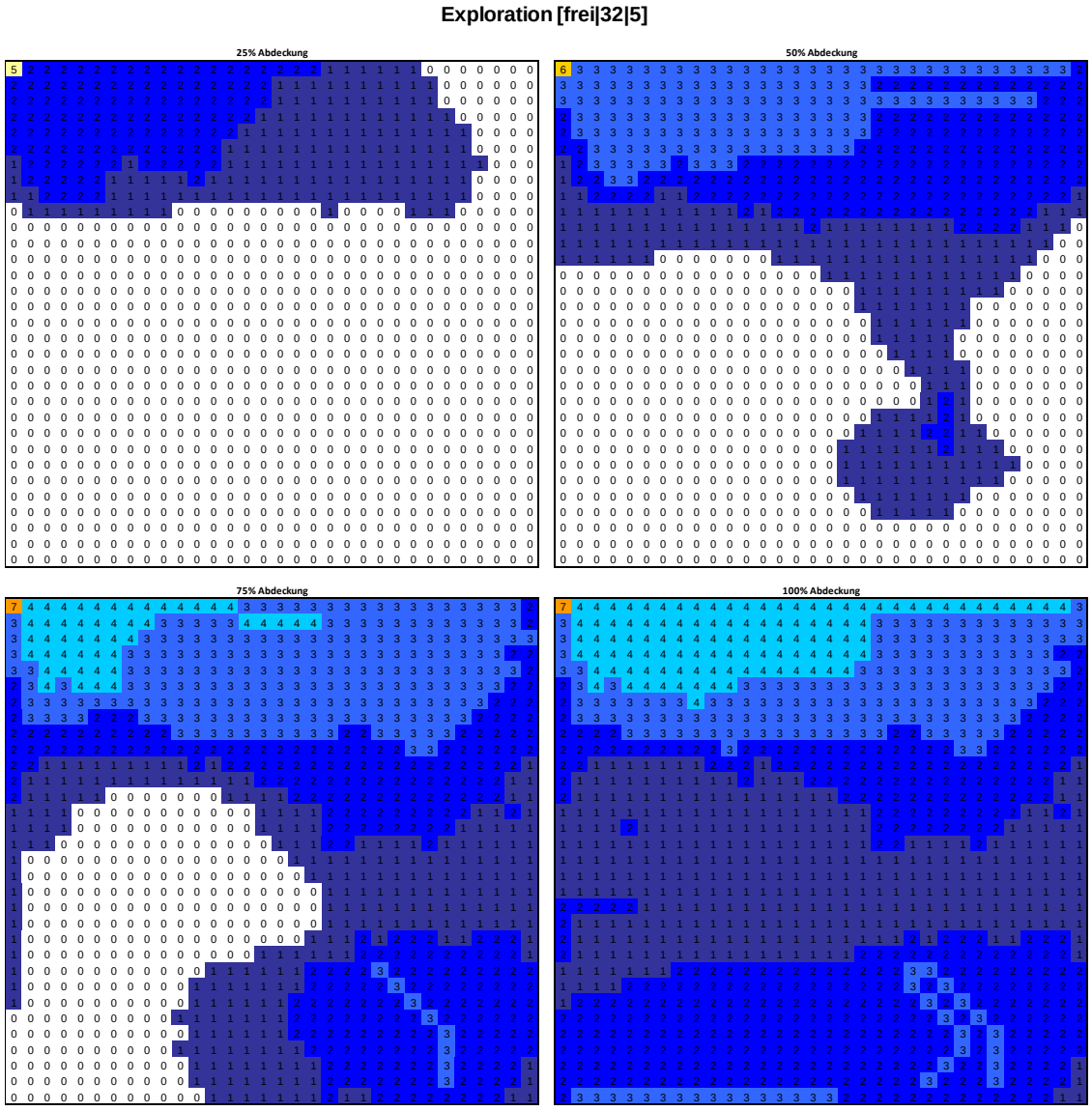


Abbildung 5.9.: NC Terrain Covering Modell - Abdeckung mit [frei|32|5]. Darstellung des besten Experimentdurchlaufes.

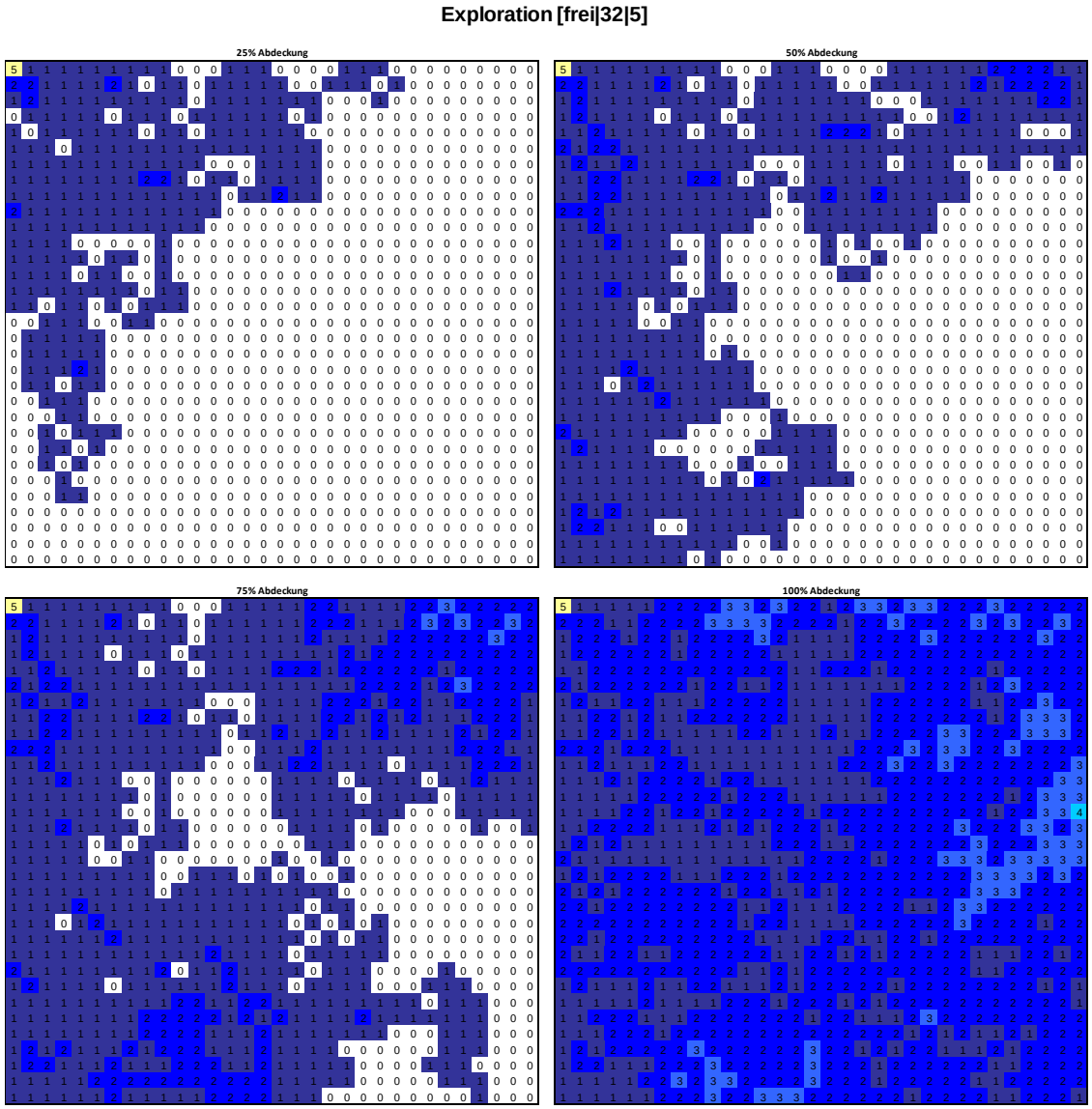
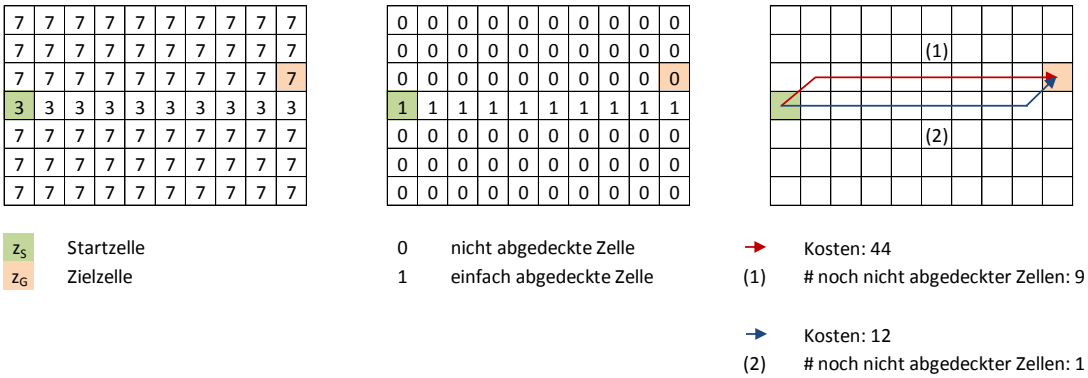


Abbildung 5.10.: NCs Terrain Covering Modell - Abdeckung mit [frei|32|5]. Darstellung des besten Experimentdurchlaufes.



(a) **(b)** **(c)**

Abbildung 5.11.: Beispiel von Möglichkeiten einer Abdeckung unter Unsicherheiten. Unter (a) Start und Zielzelle sowie Strömungsrichtung auf den Zellen, (b) Anzahl der Abdeckungen auf den Zellen und (c) Beispielrouten.

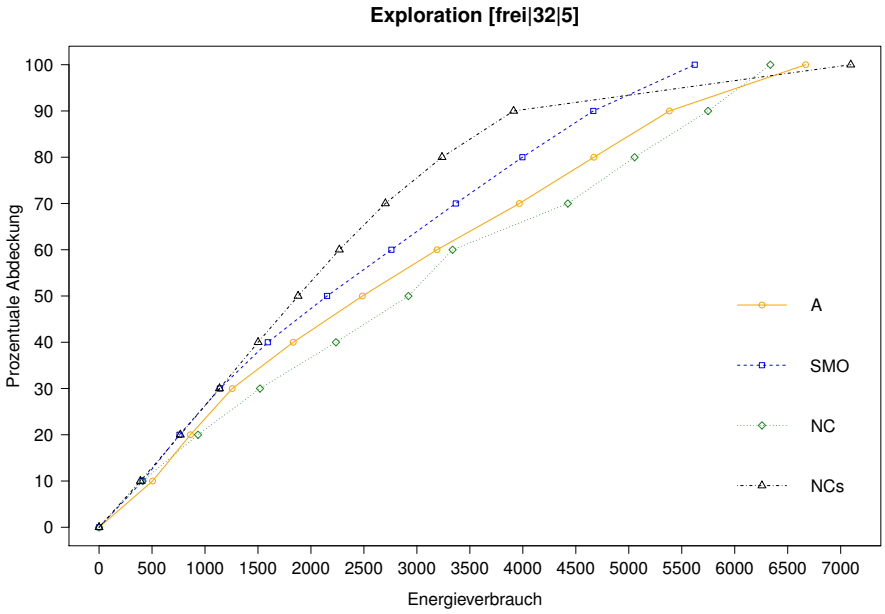


Abbildung 5.12.: Terrain Covering Modelle - Energieverbrauch mit [frei|32|5]. Darstellung der Mittelwerte über alle Experimentdurchläufe.

ten in Höhe von fünf Einheiten aufwenden. Im ungünstigsten Fall werden keine unbekannten Zellen abgedeckt und in diesem Zeitschritt resultieren Kosten in Höhe von 25 Einheiten. Es ist zu beobachten, dass für alle untersuchten TCMs die Kosteneffizienz mit fortschreitender Exploration abnimmt. Aufgrund der abnehmenden Gesamtanzahl von unbekannten Zellen in der Systemumgebung nimmt zugleich die Wahrscheinlichkeit ab, dass unbekannte Zellen auf den möglichen Routen liegen. Für das NCs-TCM, das keine Berücksichtigung der Kosten in die Exploration einbezieht, ist die Abnahme der Kosteneffizienz besonders stark zu sehen. Zu Beginn der Abdeckung liegen genügend unbekannte Zellen in der Nachbarschaft der Agenten. Trotz der hohen Effizienz bezüglich der Abdeckung unbekannter Zellen zeigt sich jedoch, dass im ersten Drittel der Exploration nur eine gemäßigte Kosteneffizienz vorliegt. Im zweiten Drittel der Exploration nimmt die Kosteneffizienz ab. Für das letzte Drittel der Exploration nimmt aufgrund der geringen Anzahl noch abzudeckender Zellen die Energieeffizienz erheblich ab. Die Kosteneffizienz für das NC-TCM zeigt einen gleichmäßigeren Verlauf über die gesamte Exploration. Ursache hierfür besteht in der Berücksichtigung der Kosten, dadurch werden zum Ende der Abdeckung stets Zellen ausgewählt, die neben einer geringen Abdeckung zugleich die niedrigsten Kosten aufweisen. Der Verlauf der Kosteneffizienz zeigt für das SMO-TCM und das A*-TCM einen ähnlichen Verlauf. Die Kosteneffizienz liegt im Durchschnitt für beide TCMs über der Kosteneffizienz des NC-TCM. Im frühen Verlauf der Exploration ist zu erkennen, dass bereits Zeitschritte auftreten, in denen keine unbekannten Zellen abgedeckt werden.

Für das SMO-TCM ist gemäß der Funktionsweise ein Diffusionsmodell integriert worden, das eine Approximation der Strömungsrichtung von unbekannten Zellen ermöglicht. In Abbildung 5.14 ist die durchschnittliche Anzahl der richtig klassifizierten Zellen abgebildet. Eine richtig klassifizierte Zelle liegt genau dann vor, wenn die Strömungsrichtung mit der höchsten Ausprägungswahrscheinlichkeit mit der tatsächlichen Strömungsrichtung übereinstimmt. Im Verlauf der Abdeckung werden bereits abgedeckte Zellen der Menge richtig klassifizierter Zellen zugeordnet. Es ist zu erkennen, dass bereits zu einem Explorationsfortschritt von 30 % durchschnittlich 905 Zellen (entspricht 88 % aller Zellen) richtig klassifiziert sind. Die Approximation von Zellen wird durch die dislozierte Exploration unterstützt, denn um jede bekannte Zelle wird die unbekannte Umgebung approximiert. Die approximierten Strömungsinformationen werden für eine kosteneffiziente Exploration herangezogen. Gemäß der Abbildung 5.12 ist zu erkennen, dass der Performanceunterschied hinsichtlich des Energieverbrauchs für das SMO-TCM gegenüber dem A*-TCM ab einem Explorationsfortschritt von 30 % kontinuierlich zunimmt.

5.2.5.1. Anzahl eingesetzter Agenten

Im Nachfolgenden werden die Auswirkungen der Anzahl eingesetzter Agenten auf die Exploration beschrieben. Grundsätzlich wirkt sich eine zunehmende Anzahl von Agenten positiv auf die totale Explorationszeit aus. Die Agenten können parallel unterschiedliche Zellen abdecken und somit die Exploration schneller vorantreiben. In Abbildung 5.15 kann dieser Sachverhalt für die freie Systemumgebung betrachtet werden. Mit Ausnahme dem NC-TCM steigt die totale Explorationszeit mit Abnahme der eingesetzten Agenten überproportional. Ferner ist zu erkennen, dass sowohl das A*-TCM als auch das SMO-TCM vergleichsweise höhere Ineffizienzen bezüglich dem Einsatz von nur einem Agenten aufweisen. Es kann vermutet werden, dass die Ineffizienzen aufgrund der Beschreibungen gemäß der Abbildung 5.11 in Verbindung mit der Aufgabenreihenfolge resultieren.

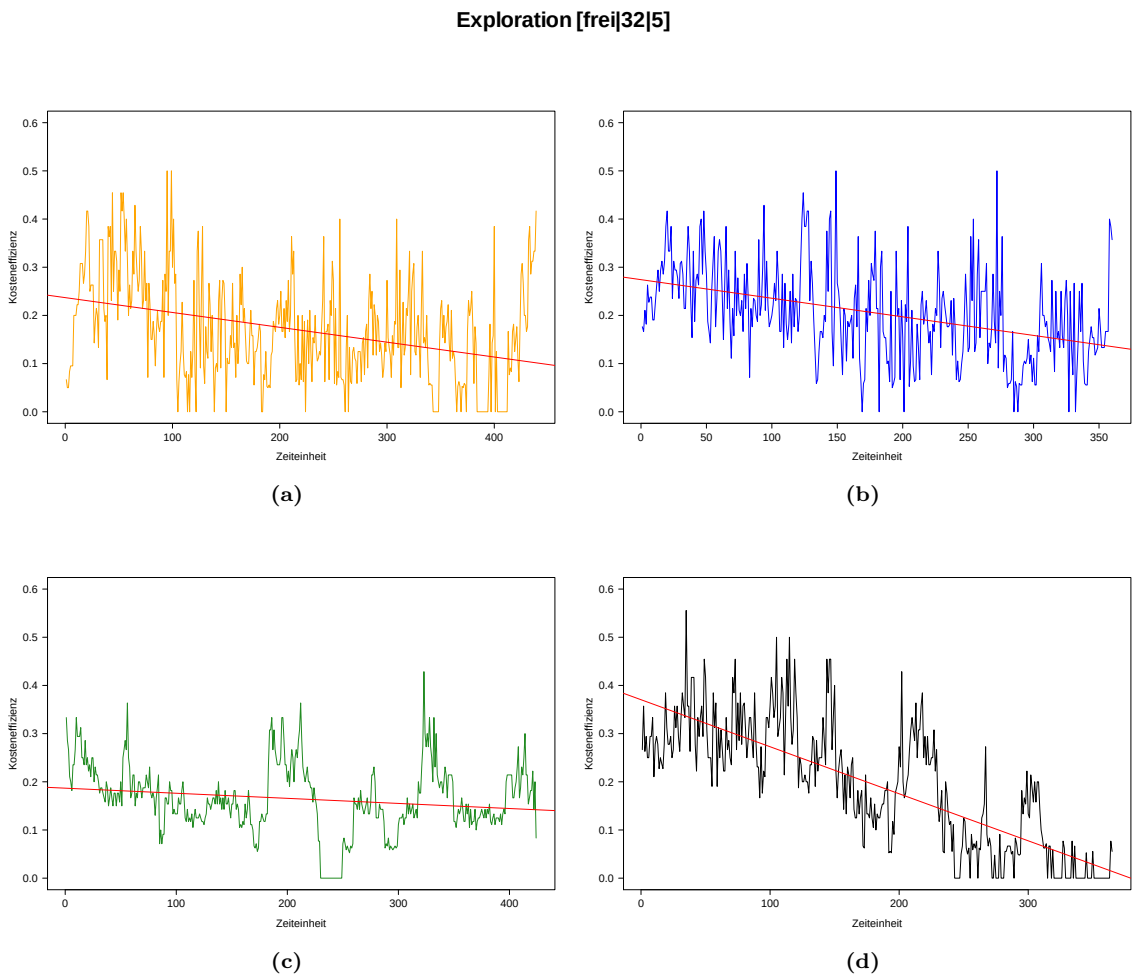


Abbildung 5.13.: Terrain Covering Modelle - Kosteneffizienz mit [frei|32|5]. Darstellung des besten Experimentdurchlaufes. Unter (a) A*-TCM, (b) SMO-TCM, (c) NC-TCM und (d) NCs-TCM.

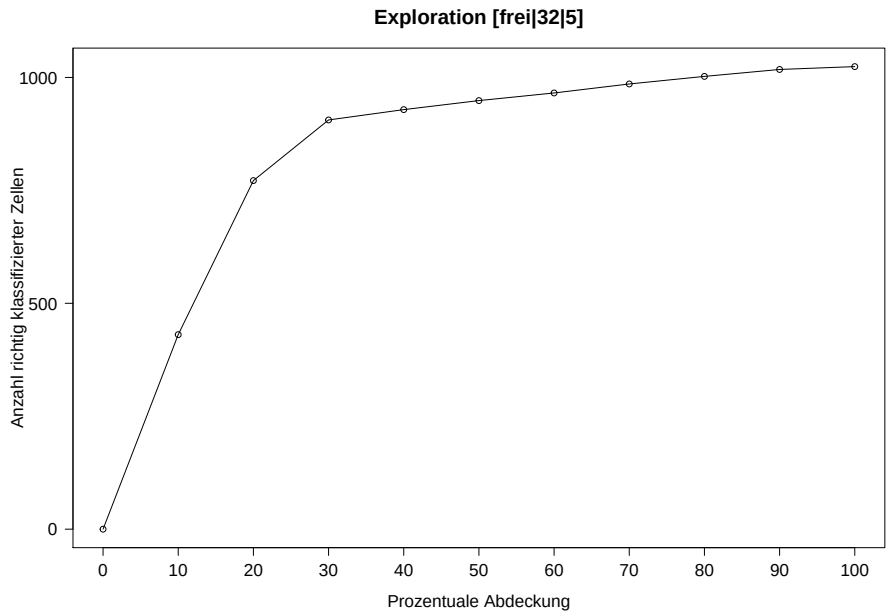


Abbildung 5.14.: SMO-TCM - Richtige Klassifikation der Strömungsrichtung auf den Zellen mit [frei|32|5]. Darstellung der Mittelwerte über alle Experimentdurchläufe.

Bei dem Einsatz mehrerer Agenten kann eine kosteneffiziente Abstimmung der Routen erfolgen. Tendenziell werden über die Auktionen Aufgabenziele genau den Agenten zugewiesen, die sich in der Nähe aufhalten. Wird nur ein Agent eingesetzt, dann muss dieser zwischen den stark dislozierten Aufgabenzielen springen. Ferner steigt zum Ende der Exploration die Wahrscheinlichkeit, dass bereits bekannte Zellen redundant befahren werden müssen. Für das SMO-TCM ist im Vergleich zum NCs-TCM zu erkennen, dass die Standardabweichung geringer ist. Die größere Standardabweichung für das NCs-TCM lässt sich mit Hilfe der Funktionsweise erklären. Die Wahrscheinlichkeit einer Exploration der letzten noch abzudeckender Zellen nimmt mit der Anzahl der Agent ab.

5.2.5.2. Kommunikationsreichweite

Die Kommunikationsreichweite nimmt einen zentralen Einfluss auf den Informationsaustausch. Informationen bezüglich der Strömungsrichtungen oder aber auch der Position von Hindernissen und bereits abgedeckter Zellen, kann durch die Agenten für eine optimierte Exploration genutzt werden. Wird der Informationsaustausch gestört, können Ineffizienzen wie beispielsweise redundante Abdeckungen auftreten. Im Sinne der Robustheit sollte eine vollständige Exploration unter allem Umständen möglich sein. Aufgrund der Funktionsweise der vier vorgestellten TCMs kann eine vollständige Exploration gewährleistet werden, falls mindestens ein Agent funktionsfähig ist. Die Kommunikation kann zudem für alle vier TCMs ausfallen und dennoch ist eine vollständige Exploration gewährleistet. Insbesondere für das A*-TCM und das SMO-TCM nimmt der Informationsaustausch einen wesentlichen Einfluss auf die Performance. Auf der Grundlage der auktionsbasierten Koordinierung müssen möglichst viele Agenten einbezogen werden. Nur dadurch können die Aufgabenziele entsprechend der Zielfunktion auf die Agenten verteilt werden. In Abbildung 5.16 ist der totale Energieverbrauch auf der freien Systemumgebung für fünf Agenten unter verschiedenen Kommunikationsreichweiten dargestellt. Gemäß der Funktionsweise ist es den Agenten möglich, ein

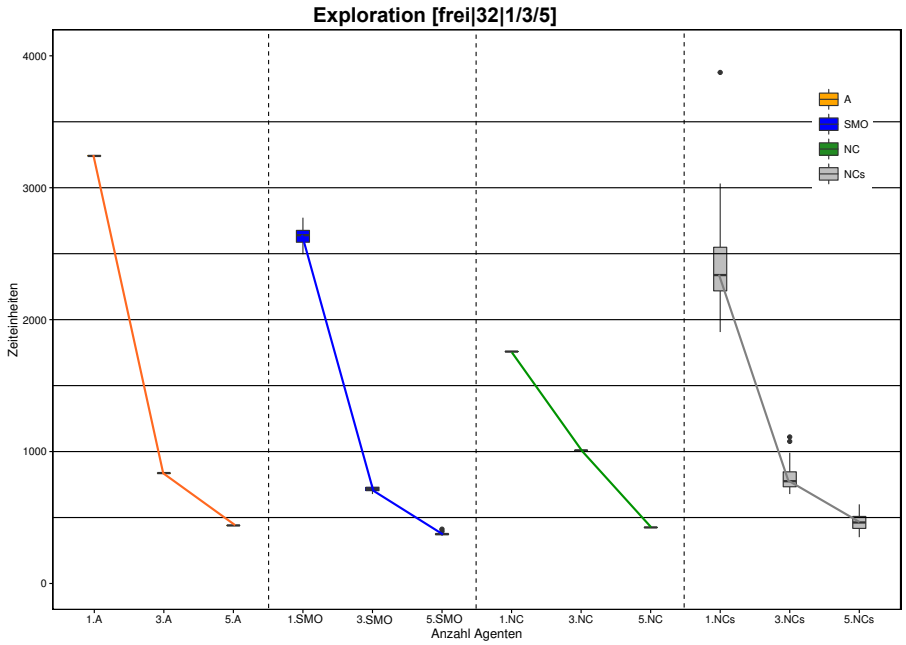


Abbildung 5.15.: Terrain Covering Modelle - Exploration mit [frei|32|1/3/5]. Darstellung der Mittelwerte über alle Experimentdurchläufe.

MANET aufzuspannen, um Informationen möglichst weit übermitteln zu können. Agenten können hierbei Informationen indirekt über Vermittleragenten versenden. Es kann gezeigt werden, dass ein geringer Informationsaustausch sowohl für das A*-TCM als auch für das SMO-TCM zu Ineffizienzen führt, die über denen des NC-TCM und NCs-TCM liegen. Die Ursache hierfür kann unter anderem in dem höheren Planungshorizont in Verbindung mit der Reihenfolge dislozierter Aufgabenziele liegen. Des Weiteren befahren die Agenten unter Umständen ein entferntes Aufgabenziel. Das Aufgabenziel hätte unter Umständen effizienter durch einen Agenten in der Nähe befahren werden können. Das NCs-TCM zeigt in den durchgeführten Experimenten eine hohe Robustheit gegenüber einem eingeschränkten Informationsaustausch. Aufgrund der Funktionsweise des NC-TCM und NCs-TCM werden ausschließlich Informationen bezüglich der Abdeckung aller Zellen ausgetauscht. Der Planungshorizont ist sehr klein, da stets nur der nächste Zeitschritt optimiert wird. Ein sporadischer Informationsaustausch kann sich vor allem auf die Exploration der letzten noch zu befahrenden Zellen auswirken.

Mit der Abbildung 5.17 ist der durchschnittliche Energieverbrauch für das SMO-TCM bezüglich der unterschiedlichen Kommunikationsreichweiten veranschaulicht. Es ist zu erkennen, dass ab einem Explorationsfortschritt von 30 % die ersten Ineffizienzen bei einer geringeren Kommunikationsreichweite auftreten. Im weiteren Verlauf zeigt sich eine Manifestation der Ineffizienzen, die sich ab einem Explorationsfortschritt von 50 % verstärken. Die Beobachtungen können in Verbindung mit der Abbildung 5.14 gebracht werden. Zu Beginn der Exploration stehen noch wenige Informationen bezüglich der Strömungsrichtung auf den Zellen zur Verfügung. Das heißt, eine umfassende Berücksichtigung führt zu Beginn der Exploration nur zu wenigen Energieeinsparungen. Des Weiteren ist zu Beginn der Abdeckung die Wahrscheinlichkeit hoch, dass auf den Routen unbekannte Zellen liegen. Zum Ende nimmt der Informationsaustausch bezüglich noch abzudeckender Zellen einen hohen Stellenwert ein.

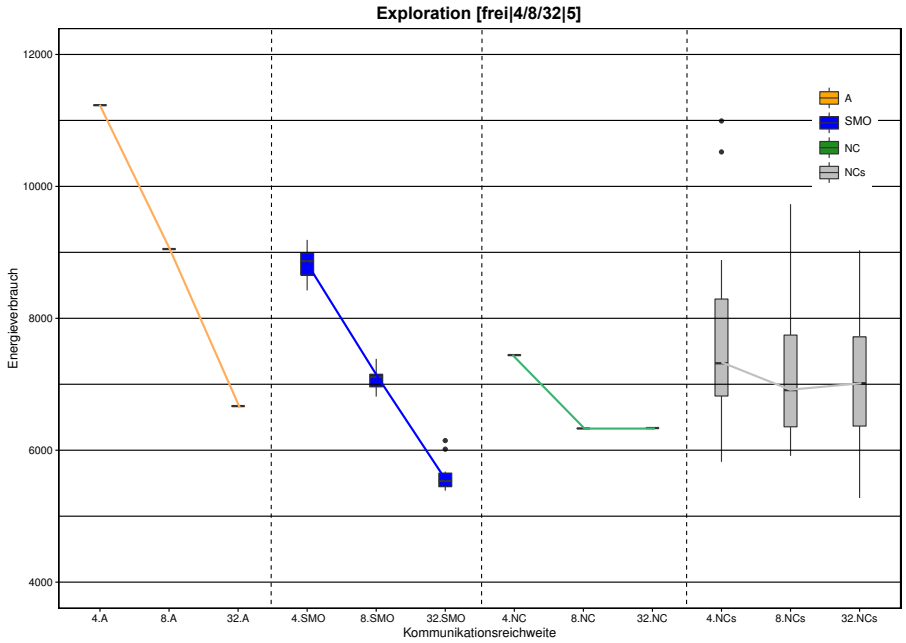


Abbildung 5.16.: Terrain Covering Modelle - Energieverbrauch mit [frei|4/8/32|5]. Darstellung der Mittelwerte über alle Experimentdurchläufe.

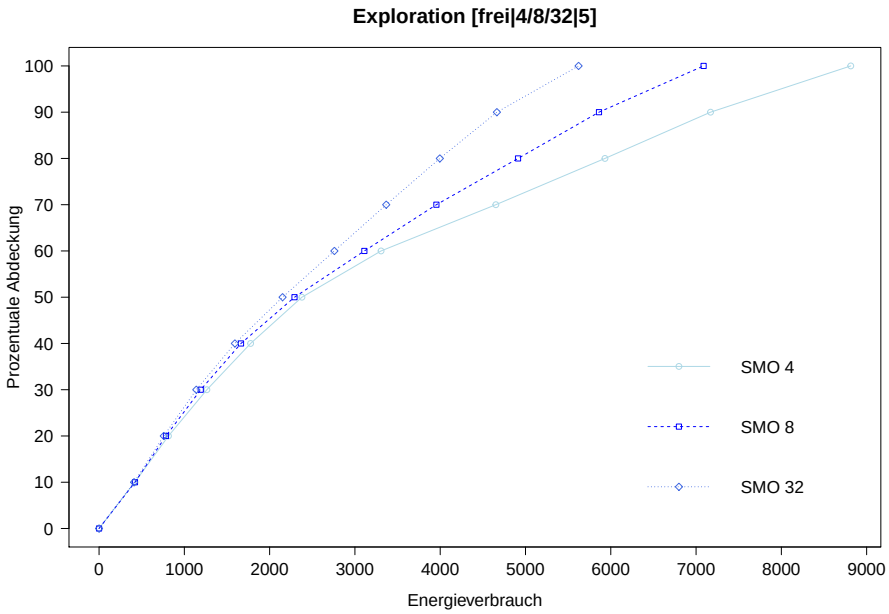


Abbildung 5.17.: SMO-TCM - Verlauf des Energieverbrauches mit [frei|4/8/32|5]. Darstellung der Mittelwerte über alle Experimentdurchläufe.

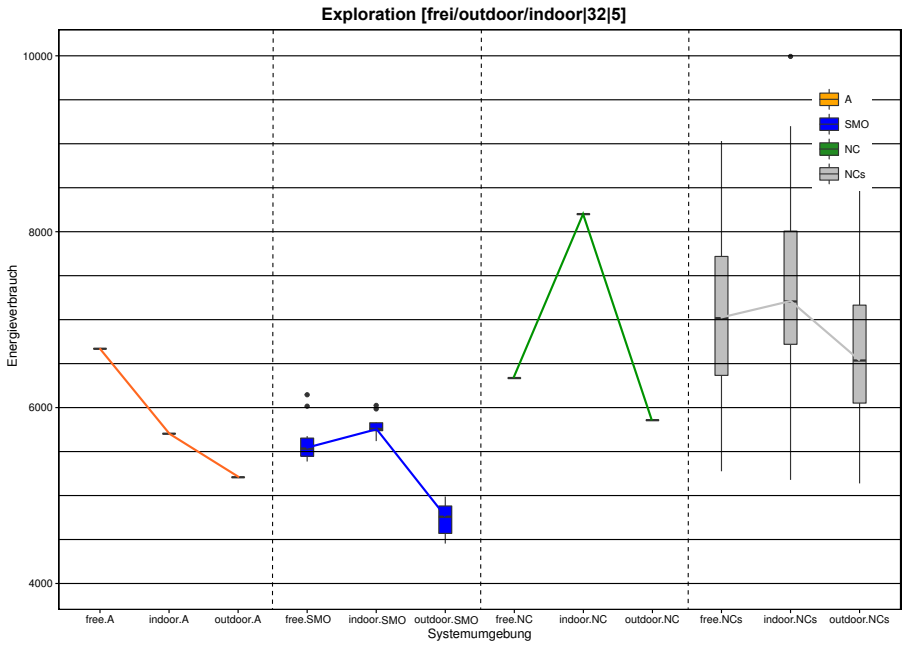


Abbildung 5.18.: Terrain Covering Modelle - Energieverbrauch mit [frei/outdoor/indoor|32|5]. Darstellung der Mittelwerte über alle Experimentdurchläufe.

5.2.5.3. Systemumgebung

Hindernisstrukturen können eine Exploration erschweren und zu Ineffizienzen führen. Insbesondere eine autonome, optimierte Exploration von engen Eintrittsöffnungen und konkaven Hindernissen stellt Agenten vor Herausforderungen. Hierbei müssen die Agenten über die Fähigkeit verfügen, die Ausprägung eines Hindernisses zu erkennen und in der weiteren Exploration zu berücksichtigen. Das A*-TCM und SMO-TCM beziehen alle Hindernisstrukturen bei der Routenplanung kontinuierlich ein. Hingegen berücksichtigt das NC-TCM und NCs-TCM die Hindernisstrukturen nur nachgeordnet. Der beschränkte Planungshorizont von einer Zeiteinheit ermöglicht keine übergeordnete Berücksichtigung von Hindernissen. In Abbildung 5.18 ist der durchschnittliche Energieverbrauch auf allen drei Systemumgebungen dargestellt. Zu erkennen ist, dass die indoor-like Systemumgebung mit konkaven Hindernisstrukturen zu Ineffizienzen führt. Mit Ausnahme des A*-TCM liegt ein erhöhter Energieverbrauch vor, obwohl zirka 14 % weniger Zellen abgedeckt werden müssen. Wird eine normierte Betrachtung der Energieverbräuche vorgenommen, dann resultieren für die outdoor-like Systemgebung im Vergleich zur freien Systemumgebung keine (A*-TCM, SMO-TCM) oder nur geringe (NC-TCM, NCs-TCM) Ineffizienzen. Vergleichbare Beobachtungen für die verschiedenen Systemumgebungen sind in [SAB+00] beschrieben. Eine normierte Gegenüberstellung der Systemumgebungen erfolgt zusammen mit einer realen Systemumgebung in der Zusammenfassung.

Die Beobachtungen hinsichtlich der Exploration der Systemumgebungen kann in der Detailbetrachtung gemäß Abbildung 5.19 bestätigt werden. Der Verlauf des Energieverbrauches zeigt für das SMO-TCM auf der freien und outdoor-like Systemumgebung einen identischen Verlauf. Der Performanceunterschied resultiert auf der geringen Anzahl abzudeckender Zellen hinsichtlich der outdoor-like Systemumgebung. Für die indoor-like Systemumgebung treten vermehrt Ineffizienzen ab einem Explorationsfortschritt von 50 % auf. Vor allem zum Ende der Exploration nehmen die

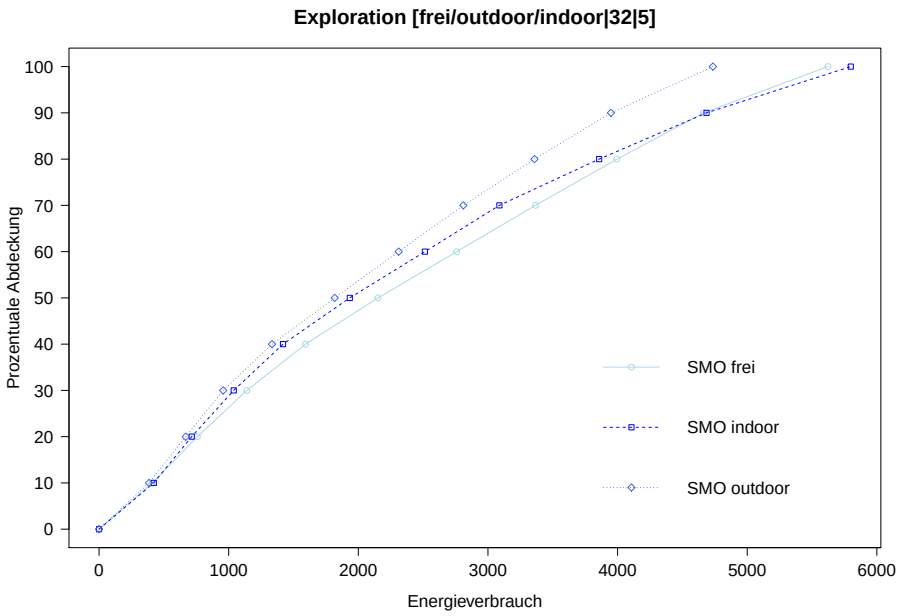


Abbildung 5.19.: SMO-TCM - Verlauf des Energieverbrauches mit [frei/outdoor/indoor|32|5]. Darstellung der Mittelwerte über alle Experimentdurchläufe.

Ineffizienzen überproportional zu. Die Ursache hierfür könnte darin liegen, dass die großen Hindernisse bei einer Umfahrung zu längeren Routen führen. Zu Beginn der Abdeckung können auf diesen Routen wenigstens noch unbekannte Zellen abgedeckt werden.

Die Abdeckungsverteilungen auf der outdoor- und indoor-like Systemumgebung sind jeweils für die TCMs in den Abbildungen 5.20, 5.21 , 5.22 , 5.23 und 5.24, 5.25, 5.26, 5.27 veranschaulicht. Es sind für die outdoor-like Systemumgebung keine Anhäufungen von Ineffizienzen direkt an den Hindernisstrukturen erkennbar. Hingegen zeigt das A*-TCM und SMO-TCM für die indoor-like Systemumgebung jeweils vermehrt Ineffizienzen an den Eintrittsöffnungen von konkaven Hindernissen sowie an den Begrenzungen. Das NC-TCM zeigt für den besten Durchlauf auf der indoor-like Systemumgebung im unteren rechten Quadranten erhebliche Ineffizienzen. Die Agenten haben Schwierigkeiten gehabt, einen Weg aus dem Hindernis herauszufinden.

5.2.5.4. CENETIX Experimentserie

Das Center for Network Innovation and Experimentation (CENETIX) an der U.S. Naval Postgraduate School in Monterey, USA, stellt seit 2004 eine einzigartige Experimentierumgebung bereit. Die Experimentierumgebung umfasst von Netzwerkstrukturen bis hin zu mobilen Plattformen eine Vielzahl von Komponenten, um die interdisziplinäre Forschung vor allem in den Bereichen der Netzwerktechnik und Netzwerkdynamik sowie den Einsatz von mobilen Plattformen oder Sensoren voranzutreiben [CEN15]. Das CENETIX führt jedes Jahr mehrere Live-Experimente mit zivilen und militärischen Partnern durch. Das Core Competence Center For Operations Research Tenacity - Management - Excellence Safety & Security ALLIANCE (COMTESSA) an der Universität der Bundeswehr München beteiligt sich durch Professor Pickl aktiv mit wissenschaftlichen Beiträgen an der Experimentserie. So hat unter anderem vom 18. bis 23. September 2016 vor der

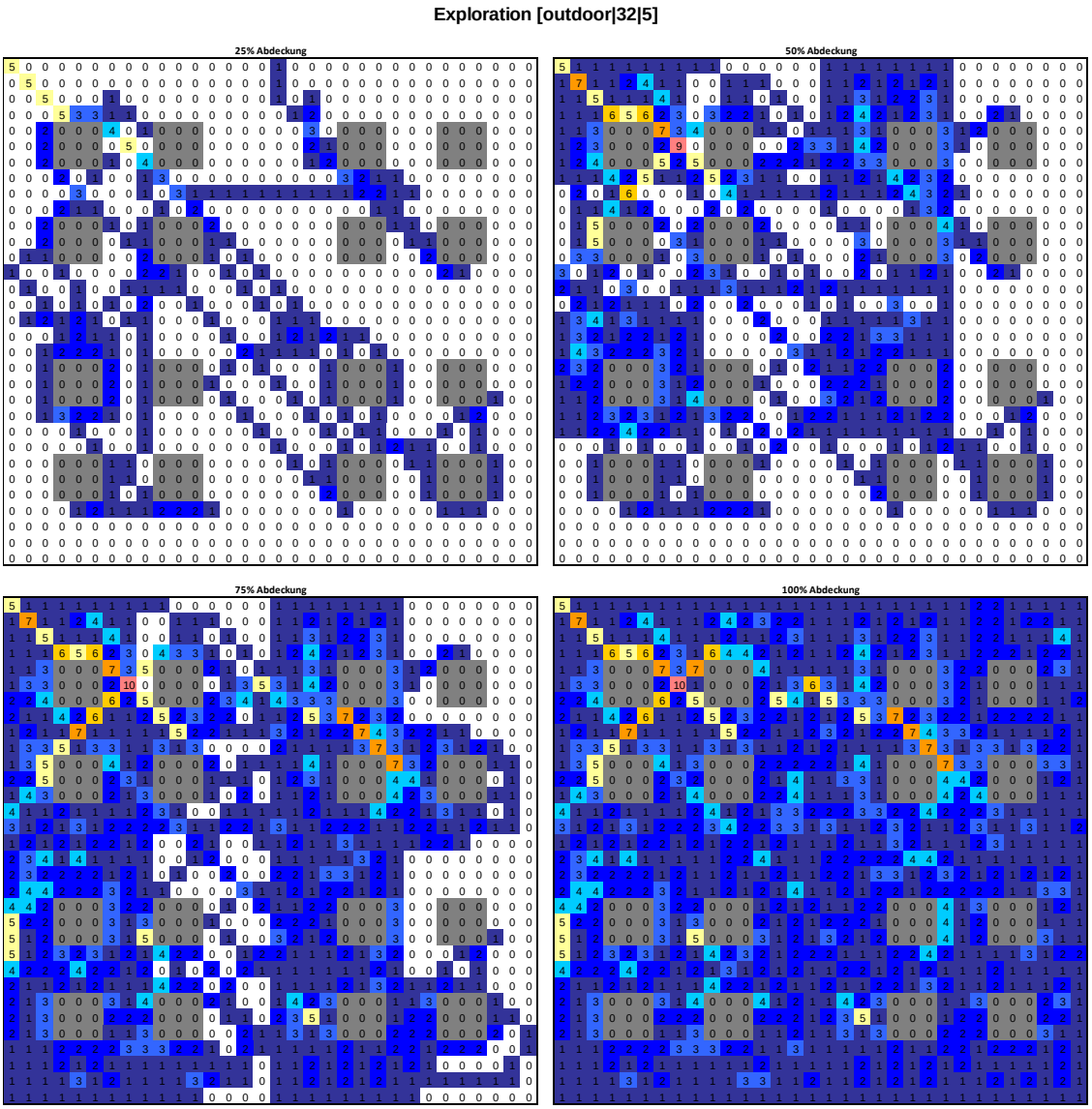


Abbildung 5.20.: A* Terrain Covering Modell - Abdeckung mit [outdoor|32|5]. Darstellung des besten Experimentdurchlaufes.

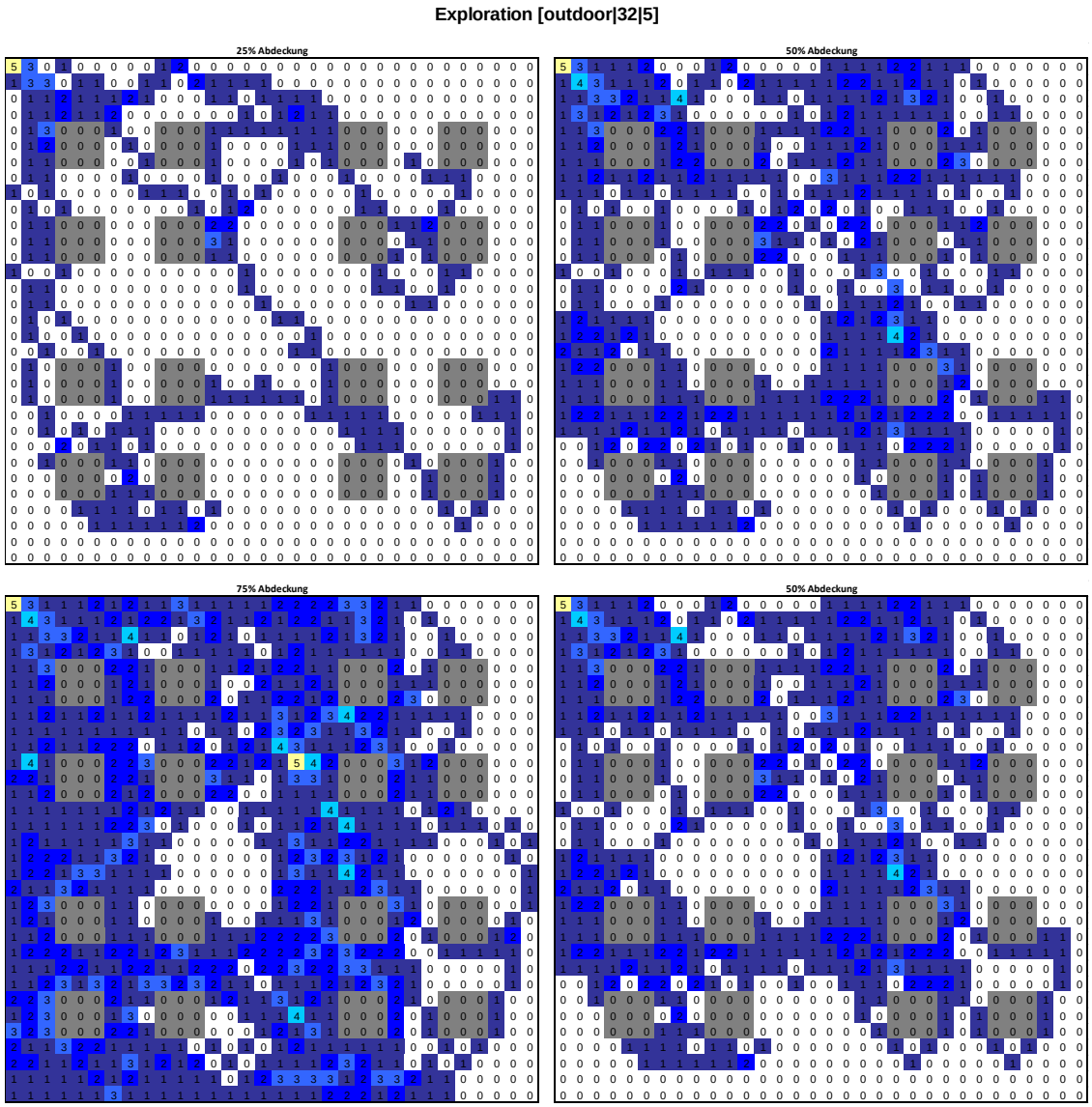


Abbildung 5.21.: SMO Terrain Covering Modell - Abdeckung mit [outdoor|32|5]. Darstellung des besten Experimentdurchlaufes.

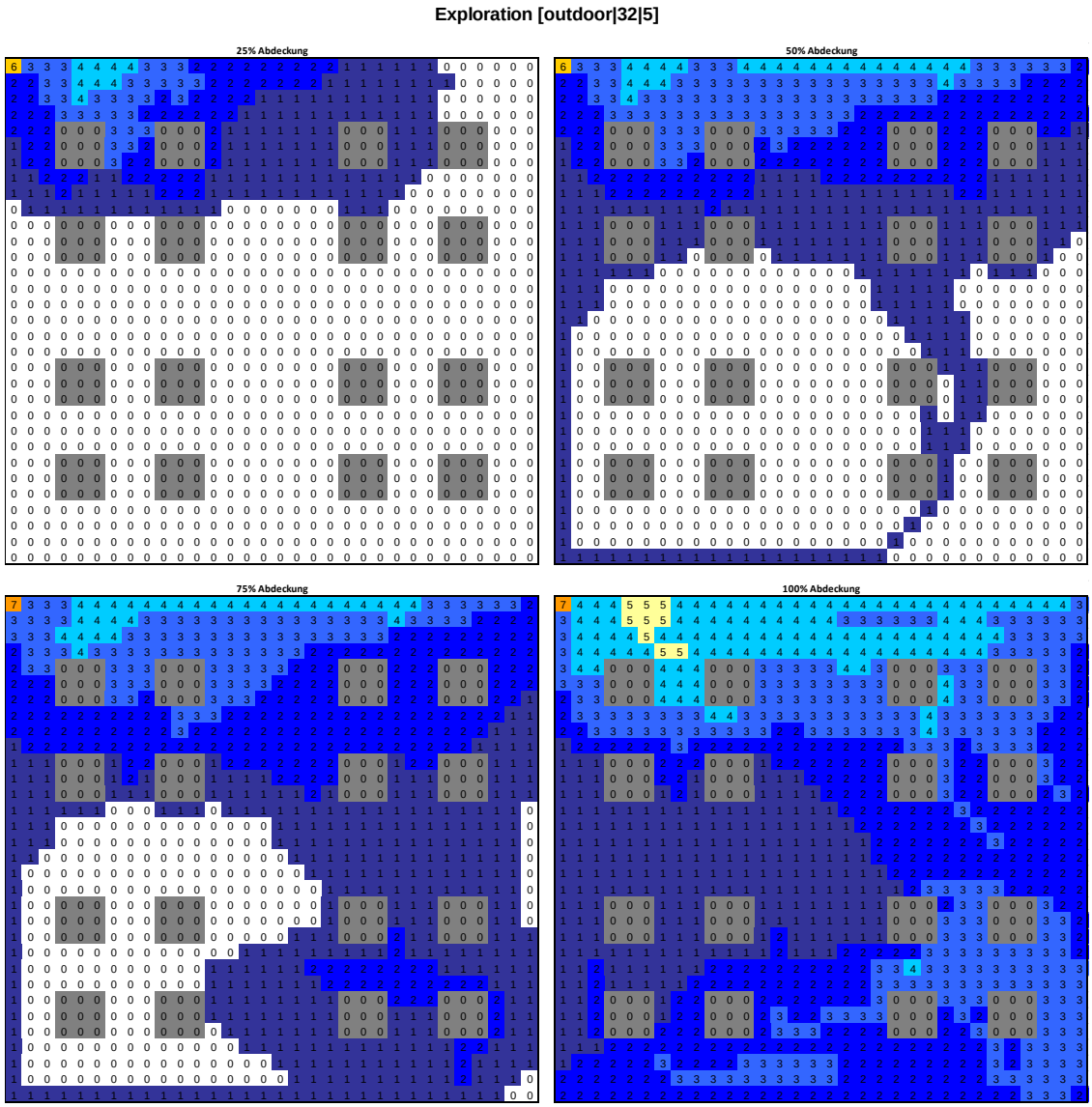


Abbildung 5.22.: NC Terrain Covering Modell - Abdeckung mit [outdoor|32|5]. Darstellung des besten Experimentdurchlaufes.

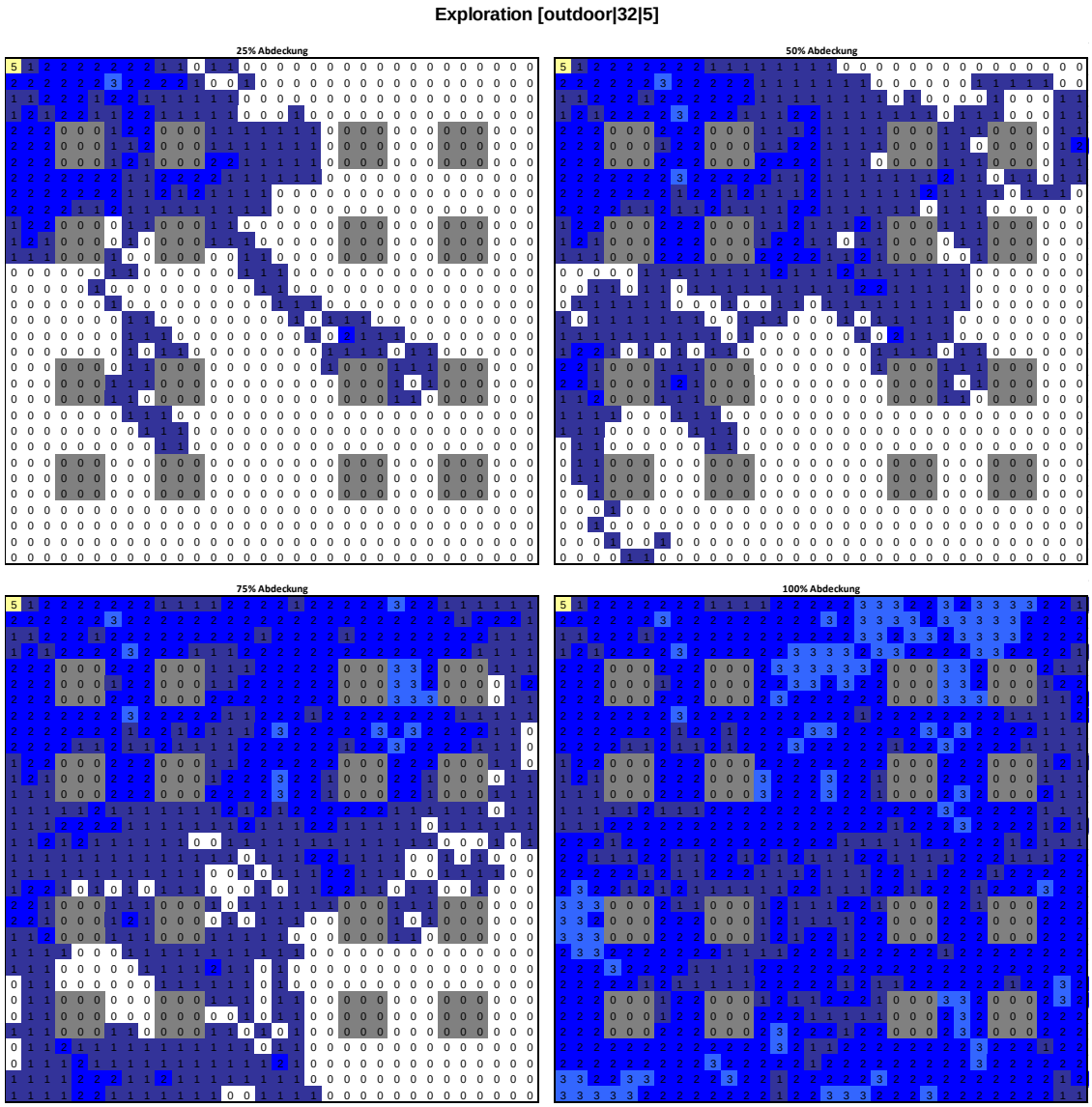


Abbildung 5.23.: NCs Terrain Covering Modell - Abdeckung mit [outdoor|32|5]. Darstellung des besten Experimentdurchlaufes.

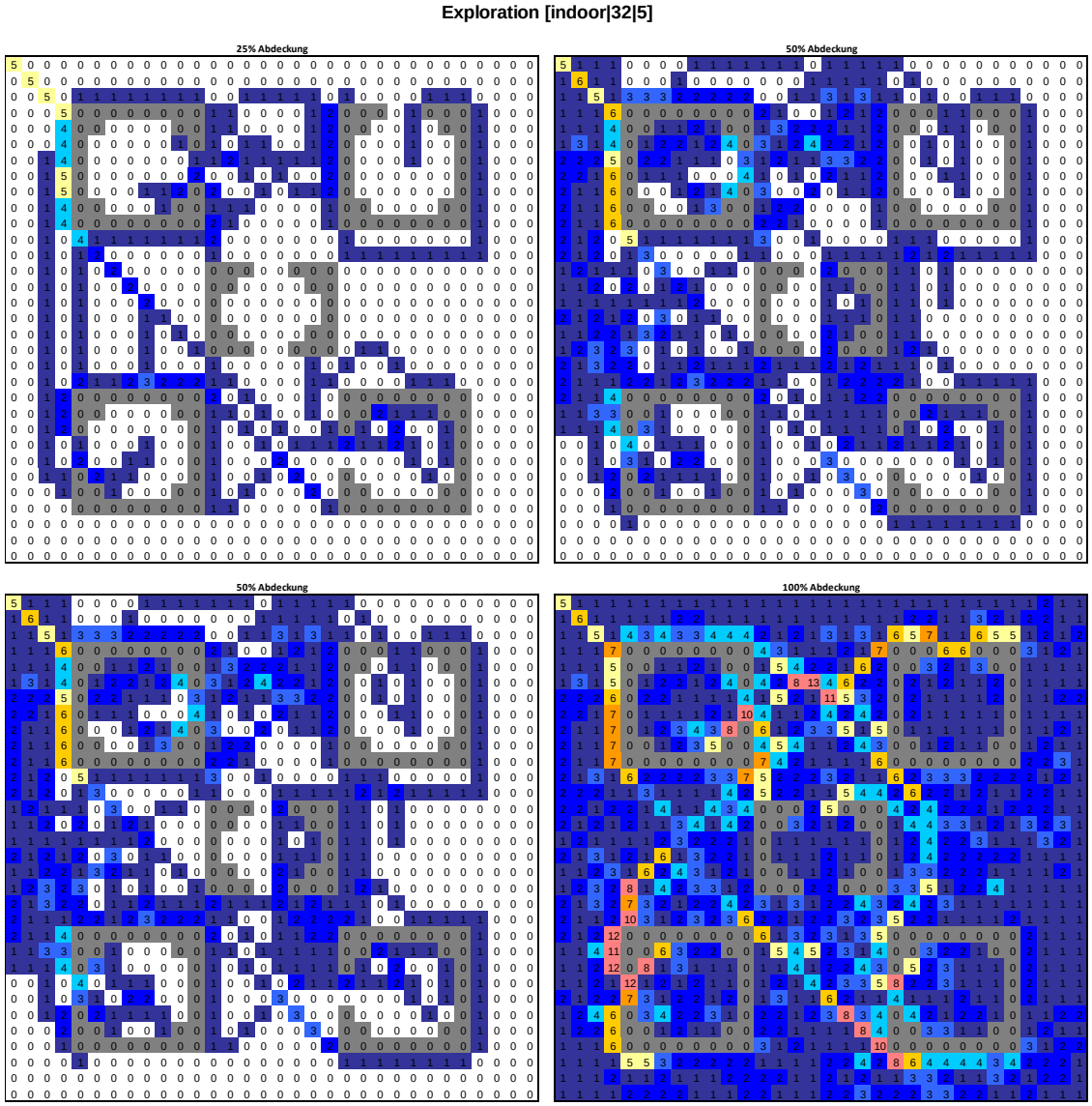


Abbildung 5.24.: A* Terrain Covering Modell - Abdeckung mit [indoor|32|5]. Darstellung des besten Experimentdurchlaufes.

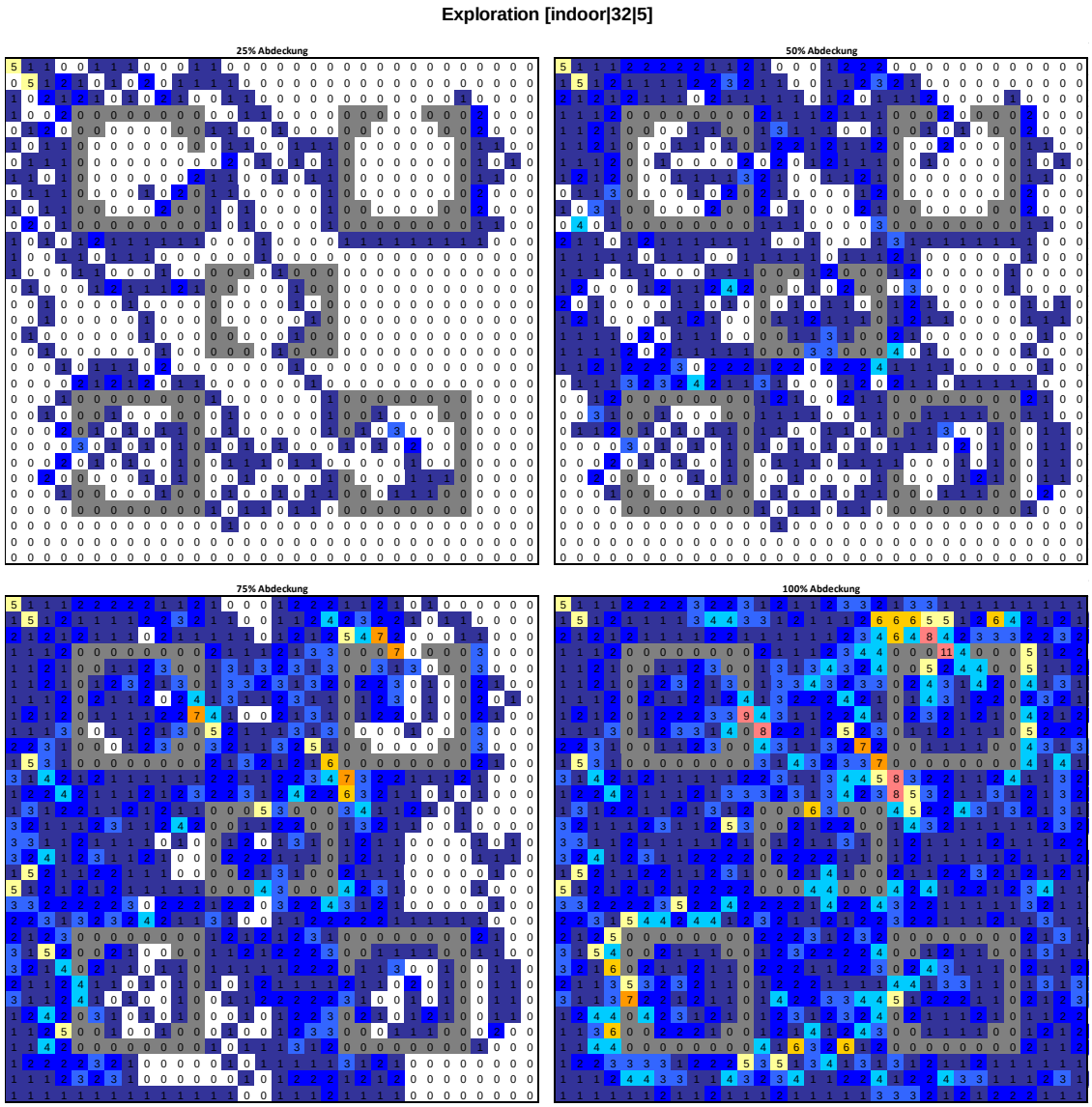


Abbildung 5.25.: SMO Terrain Covering Modell - Abdeckung mit [indoor|32|5]. Darstellung des besten Experimentdurchlaufes.

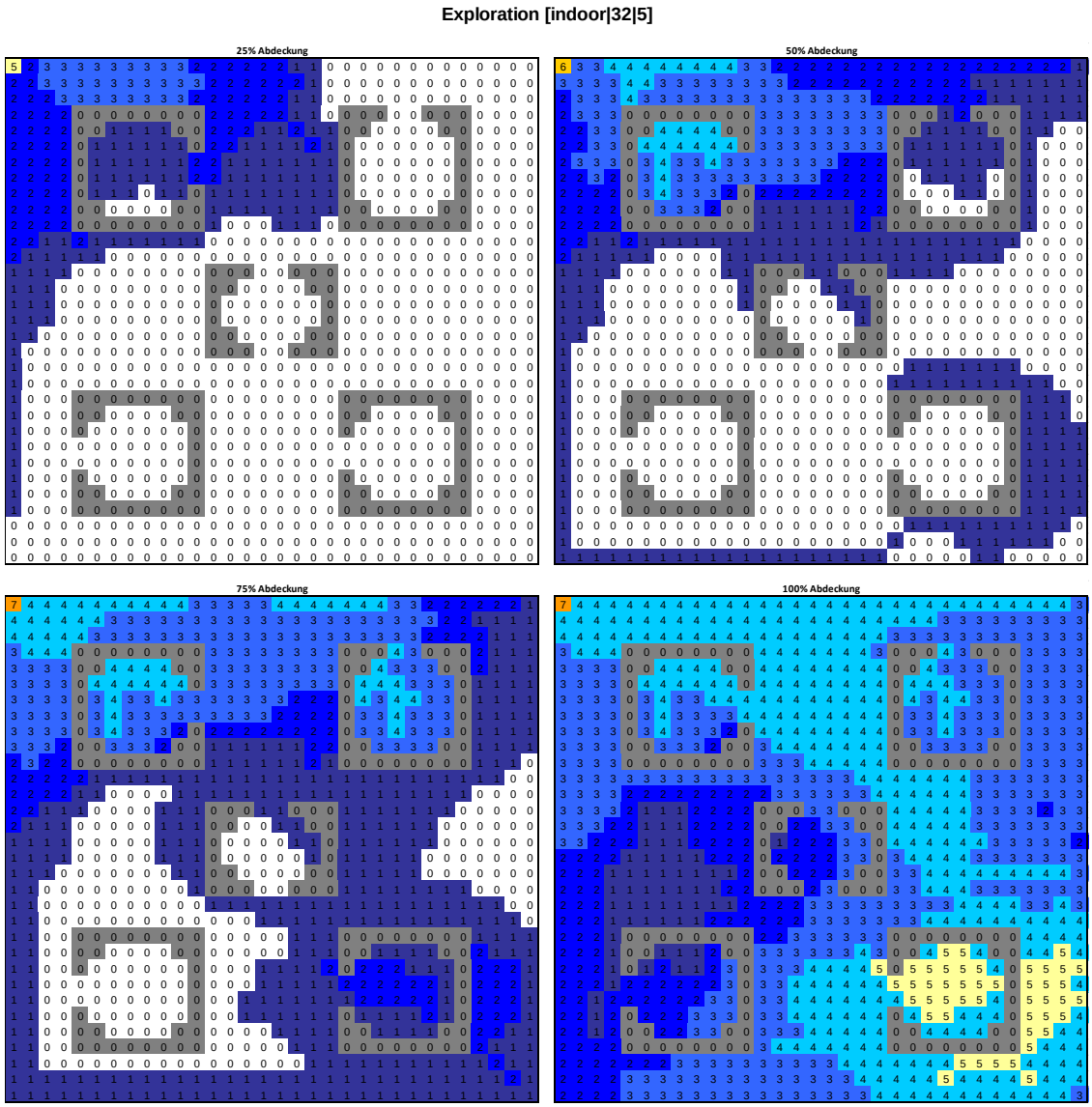


Abbildung 5.26.: NC Terrain Covering Modell - Abdeckung mit [indoor|32|5]. Darstellung des besten Experimentdurchlaufes.

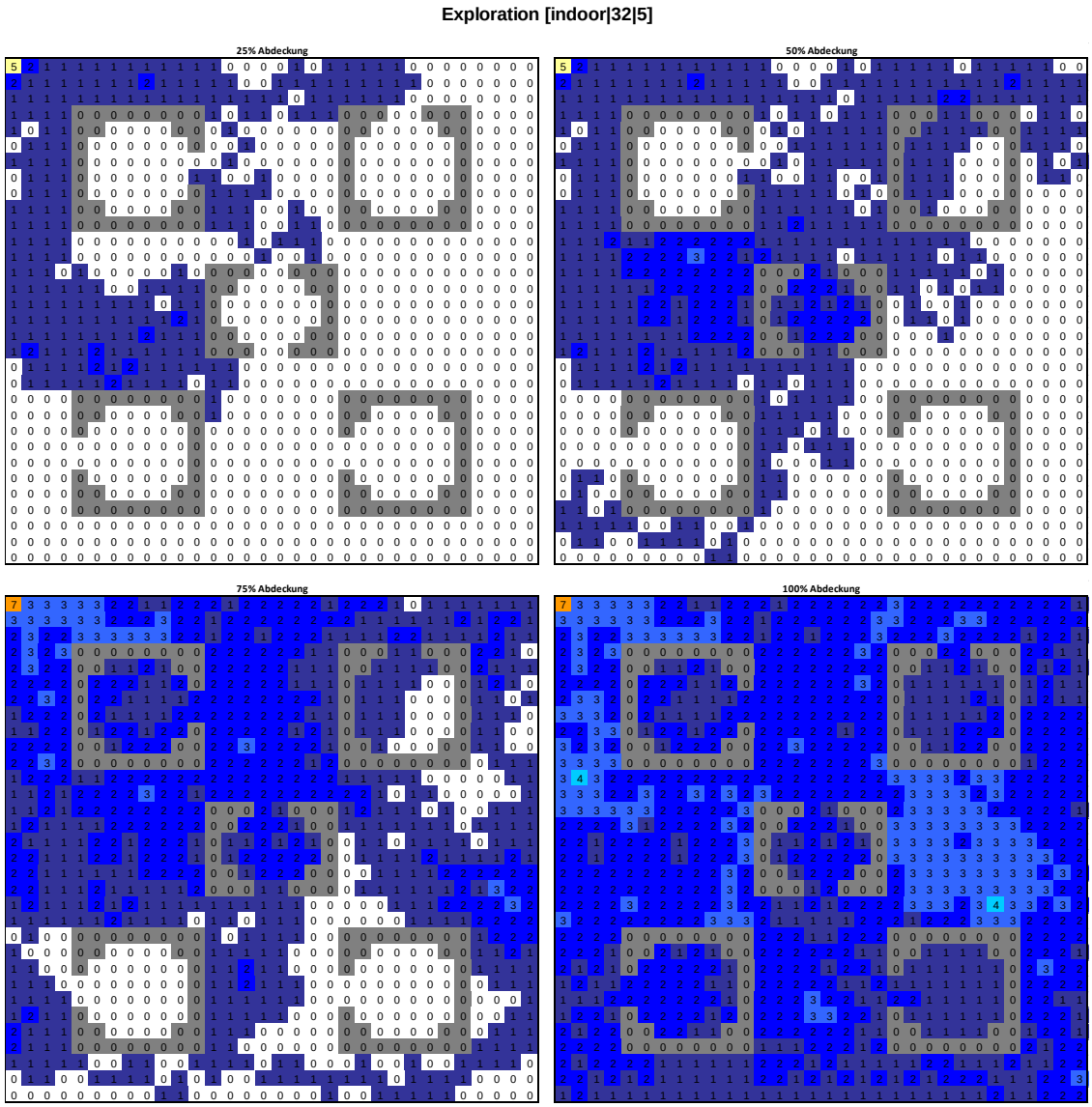


Abbildung 5.27.: NCs Terrain Covering Modell - Abdeckung mit [indoor|32|5]. Darstellung des besten Experimentdurchlaufes.

Abbildung 5.28.: Live-Experiment in der Bucht von San Francisco, 21.09.2016. Die US Küstenwache befährt vordefinierte Routen, im Sinne einer Patrouille.

Küste von San Francisco ein Live-Experiment stattgefunden. Hierbei sind verschiedener Technologien identifiziert und überprüft worden, die eine Implementierung von einem MANET, bestehend aus maritimen Plattformen (US Küstenwache, siehe Abbildung 5.28), ermöglichen. Mit Hilfe der mobilen Plattformen können Informationen gesammelt und das Situationsbewusstsein verbessert werden, um beispielsweise den Schiffsverkehr umfassender überwachen zu können. Während des Live-Experiments sind Kommunikationsprotokolle und Netzwerkflüsse sowie die Identifizierung, Klassifizierung und Nachverfolgung von mobilen Plattformen überprüft worden, um einen Informationsaustausch sicherstellen zu können.

Gemäß der Abbildung 5.29 (a) und (b) ist eine Kodierung der Bucht von San Francisco vorgenommen worden. Auf der resultierenden Systemumgebung ist eine Überprüfung der TCMs durchgeführt worden. Mit dem Anwendungsbeispiel wird simulationsbasiert der Einsatz von autonomen Agenten anhand von realen Umgebungsstrukturen aufgezeigt. Die Systemumgebung umfasst konkave und konvexe Umgebungsstrukturen. Insbesondere die engen Einbuchtungen stellen Herausforderungen dar, die von autonomen Agenten möglichst effizient abgedeckt werden müssen. Die Systemumgebung besteht aus insgesamt 1024 Zellen mit 438 nicht befahrbaren Hinderniszellen (586 frei befahrbare Zellen). Die Agenten starten mit der Exploration auf der Zelle (27,17). Die vorinitialisierten Strömungsinformationen nach [Gro98] sind in Abbildung 5.29 (c) und (d) veranschaulicht.

Zu Beginn der Analyse der TCMs ist festzuhalten, dass auf dieser Systemumgebung das NC-TCM zu verschiedenen konkreten Explorationen führt. Zuvor sind auf den untersuchten Systemumgebungen (frei/outdoor/indoor) keine der drei Fälle gemäß Abbildung 5.2 aufgetreten, in denen stochastisch eine Entscheidung herbeigeführt werden hätte müssen. Hinsichtlich der Bucht von San Francisco sind hingegen zufallsbasierte Entscheidungen zwischen jeweils zwei indifferenten Zellen notwendig, deren Befahrung zu den gleichen approximierten Kosten führt. Ferner wird auf dieser Systemumgebung von dem A*-TCM und SMO-TCM die bereits zuvor implementierte Fähigkeit benötigt, dass die Agenten selbstständig entfernte Zellen als nicht befahrbar identifizieren. Entfernte nicht befahrbare Zellen können aufgrund von Hindernisstrukturen resultieren und die entsprechende Aufgabe muss gelöscht werden. Ein Beispiel stellt unter anderem die Zelle (0,31) dar. Die Zelle kann durch die Agenten wegen der angrenzenden Hinderniszellen nicht eigenständig untersucht werden.

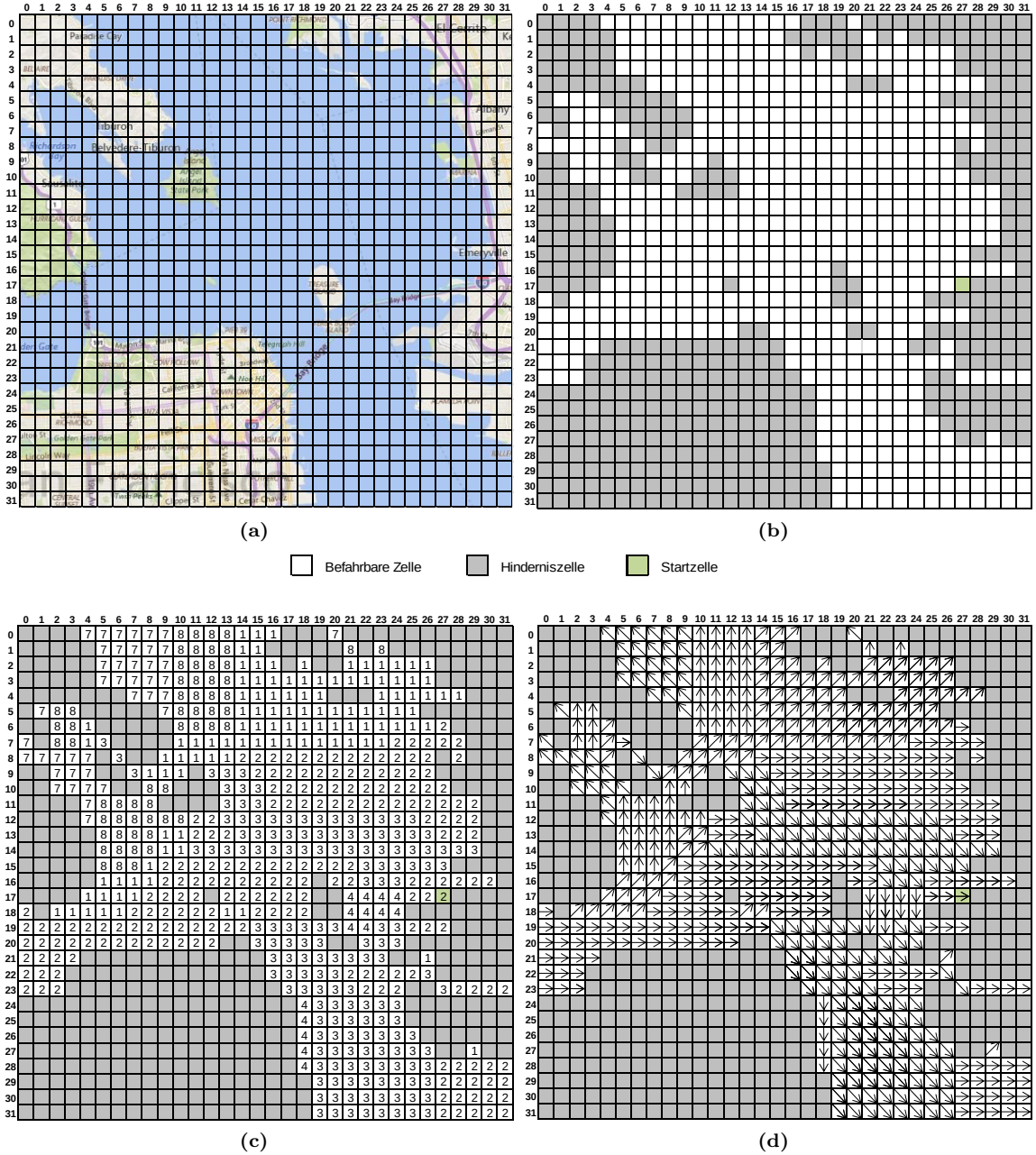


Abbildung 5.29.: Experimentinitialisierung für die CENETIX Experimentserie. Unter (a) Bing-Maps der Bucht von San Francisco [Quelle: https://www.bing.com/mapspreview], (b) kodierte Bucht von San Francisco, (c) kodierte Strömungskosten auf den Zellen und (d) vereinfachte Darstellung der Strömungskosten auf den Zellen.

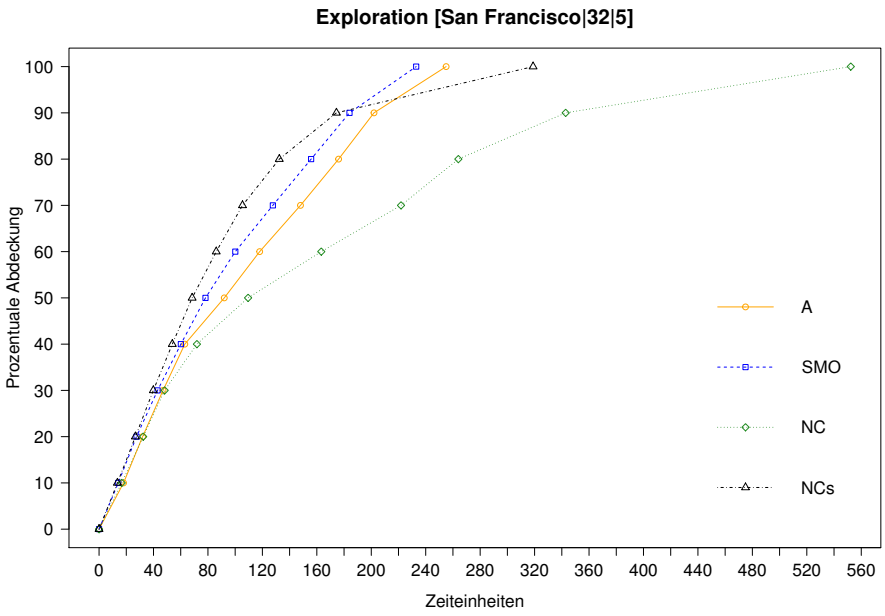


Abbildung 5.30.: Terrain Covering Modelle - Explorationsverlauf mit [San Francisco|32|5]. Darstellung der Mittelwerte über alle Experimentdurchläufe.

Die Exploration mit [San Francisco|32|5] zeigt in Abbildung 5.30, dass durchschnittlich das SMO-TCM die beste Performance hinsichtlich der totalen Explorationszeit aufweist. Der Explorationsverlauf des A*-TCM ist ähnlich und weist zirka 10 % mehr Ineffizienzen/Redundanzen auf. Das NCs-TCM zeigt wie zuvor für die ersten 90 % der Exploration die beste Performance unter den TCMs. Der Performanceunterschied fällt jedoch erheblich geringer als auf den zuvor untersuchten Systemumgebungen aus. Das NC-TCM zeigt auf dieser Systemumgebung den schlechtesten Explorationsverlauf. Erhebliche Ineffizienzen treten bereits früh ab einem Explorationsfortschritt von zirka 50 % auf. Derartig viele Ineffizienzen sind auf den zuvor experimentierten Systemumgebungen nicht aufgetreten. Angenommen es werden in einem realen Anwendung fünf autonome Unterwasserfahrzeuge gemäß dem SMO-TCM eingesetzt, die konstant zehn Kilometer pro Stunde zurücklegen, dann würde eine vollständige Exploration zirka 14.5 Stunden in Anspruch nehmen. Die Ausdehnung einer Zelle beträgt hierbei zirka 0.65 Kilometer.

Die Ineffizienzen bezüglich der Mehrfachabdeckungen sind in Abbildung 5.31 zusammengefasst. Für das SMO-TCM sind durchschnittlich 75 % der Zellen einfach oder doppelt abgedeckt worden. Eine ähnliche Abdeckung mit 70 % zeigt das A*-TCM. Hingegen führen die bereits dargestellten Ineffizienzen für das NC-TCM dazu, dass lediglich 12 % der Zellen maximal doppelt abgedeckt werden. Eine bessere Performance mit 36 % weist das NCs-TCM auf. Ferner gilt für die Abdeckungsredundanz

$$G_\Lambda(\text{SMO-TCM})=1139 < G_\Lambda(\text{A*-TCM})=1258 < G_\Lambda(\text{NCs-TCM})=1650 < G_\Lambda(\text{NC-TCM})=2767$$

Die Abdeckungsverteilung des jeweils besten Experimentdurchlaufes der TCMs kann den Abbildungen 5.32 und 5.33 entnommen werden. Aufgrund der Aufgabenreihenfolge mit stark dislozierten Aufgabenzielen zeigen das A*-TCM und SMO-TCM auch auf dieser Systemumgebung, bereits zu Beginn der Exploration, eine Abdeckung unterschiedlicher Regionen. Die dislozierte Exploration

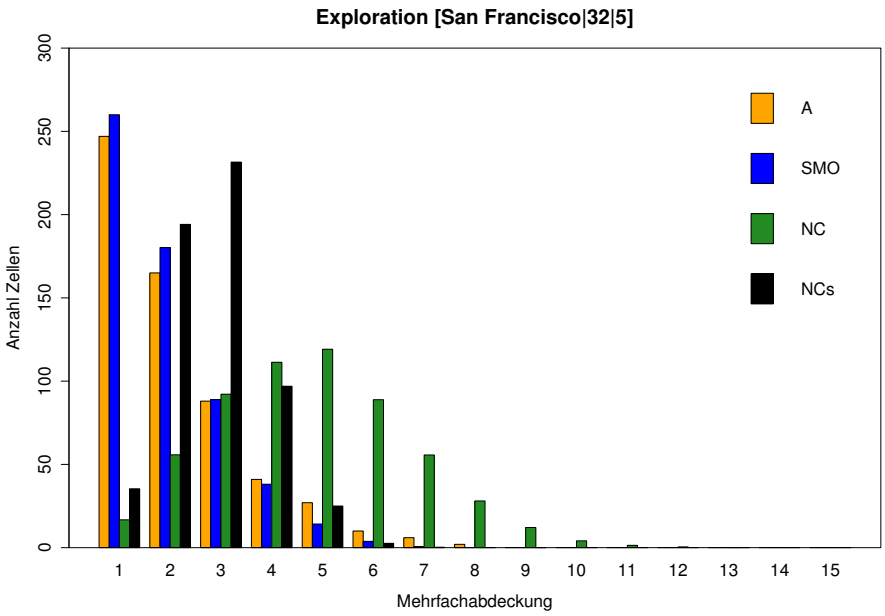
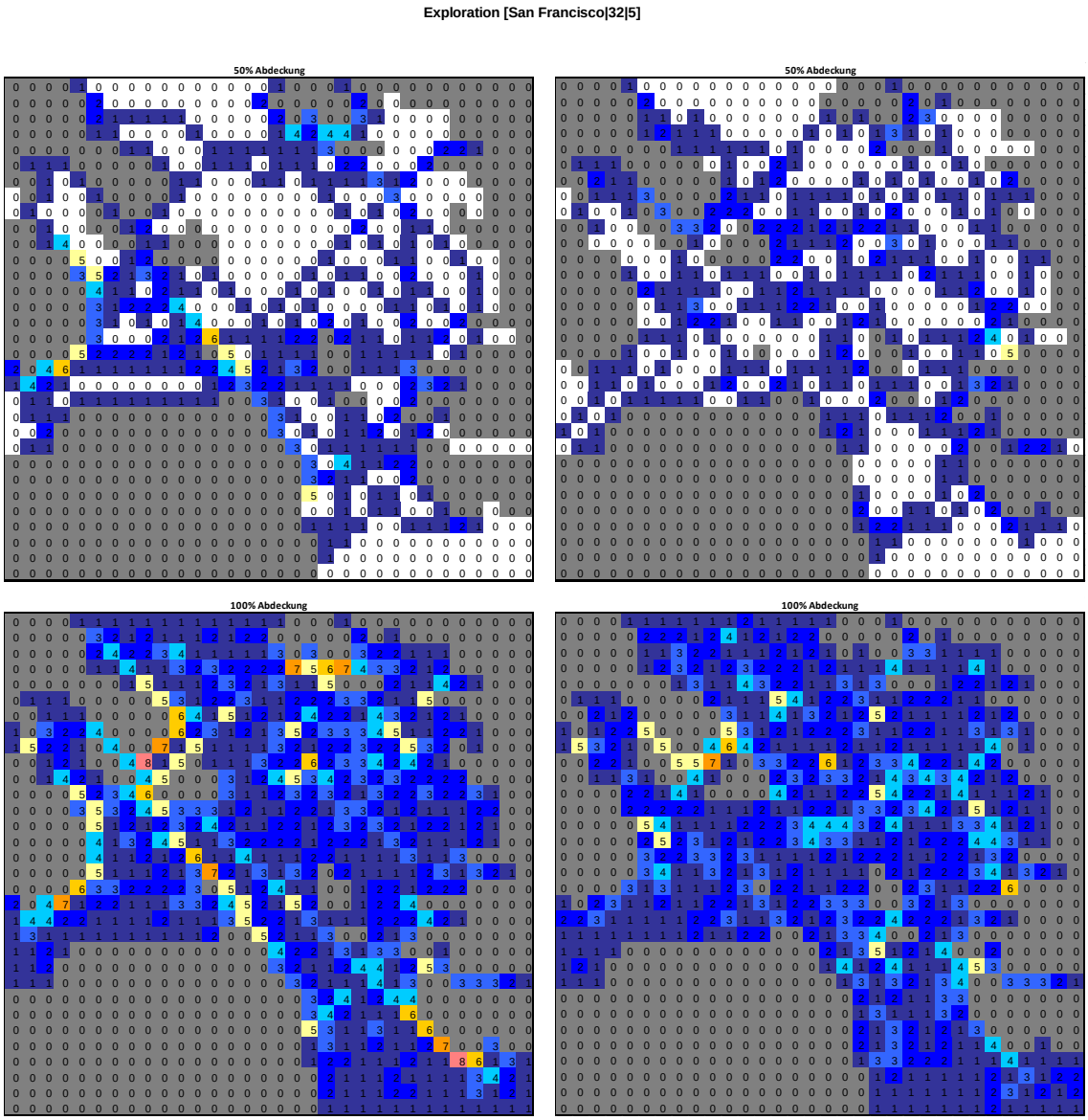


Abbildung 5.31.: Terrain Covering Modelle - Zellen mit Mehrfachabdeckungen mit [San Francisco|32|5]. Darstellung der Mittelwerte über alle Experimentdurchläufe.

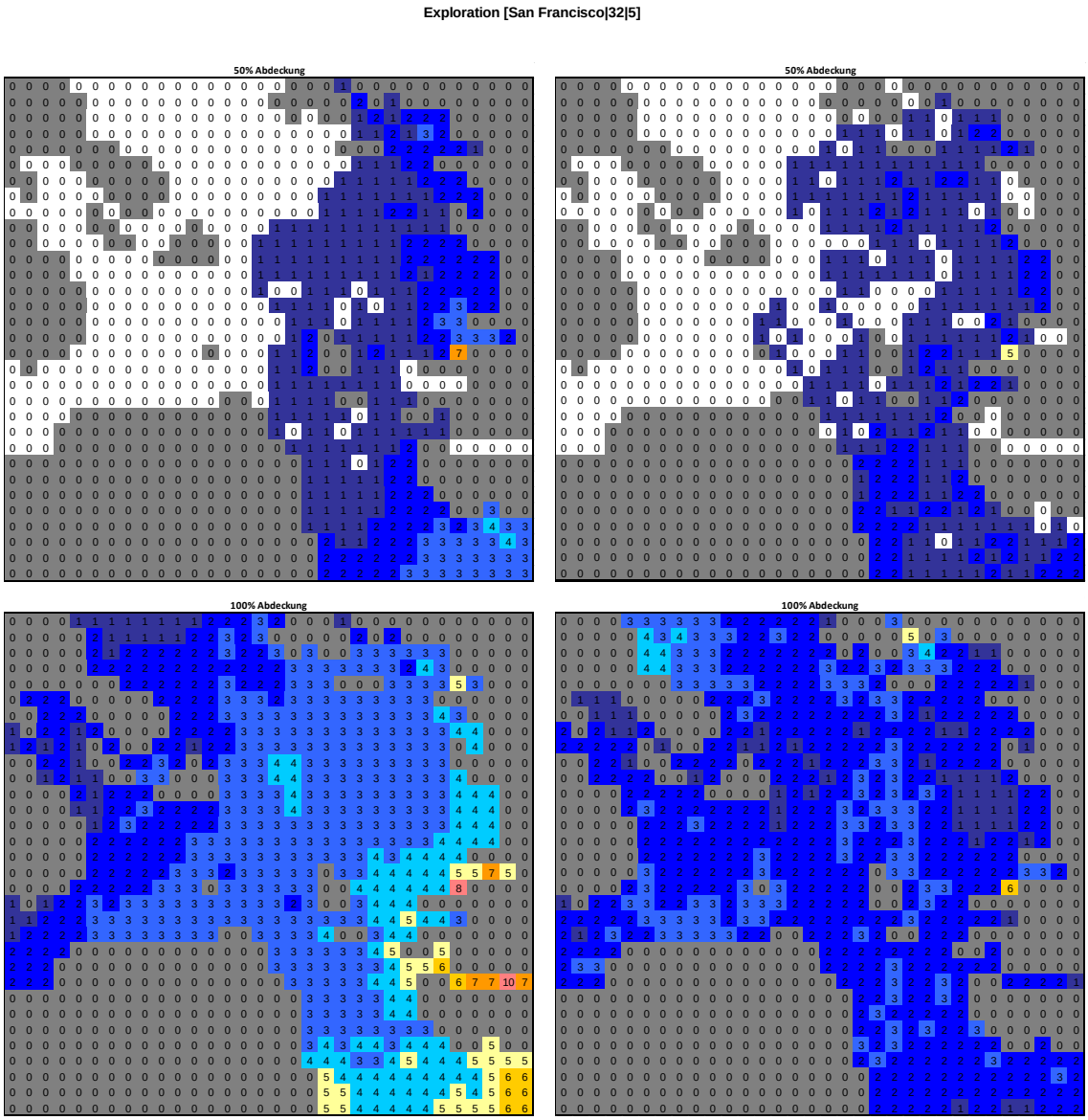
kann vor allem für Szenare der Überwachung und Aufklärung einen positiven Mehrwert leisten. Mit dieser Eigenschaft werden frühzeitig Informationen verschiedener Regionen gesammelt. Es zeigt sich, dass vor allem das NC-TCM Schwierigkeiten mit der Abdeckung enger Einbuchtungen hat. Ebenso zeigt das NCs-TCM ein vergleichbares Verhalten, das auf die Funktionsweise zurückzuführen ist. Werden die Zellen vor der Eintrittsöffnungen zuerst abgedeckt, dann ist eine spätere Exploration nur mit hohen Redundanzen zu erreichen. Größere Teilbereiche um die Eintrittsöffnung herum müssen auf das gleiche Niveau der Mehrfachabdeckungen angehoben werden. Ferner ist zu beobachten, dass das NC-TCM auch auf dieser Systemumgebung eher zusammenhängende Bereiche mit gleicher Anzahl von Mehrfachabdeckungen erzeugt.

Der totale Energieverbrauch ist in Abbildung 5.34 dargestellt. Das SMO-TCM kann die Systemumgebung mit dem geringsten Energieverbrauch abdecken. Zudem ist die Standardabweichung vergleichbar gering. Das SMO-TCM hat gegenüber dem A*-TCM die Systemumgebung 10 % schneller abgedeckt und zeigt durchschnittlich zudem einen 14 % geringeren Energieverbrauch. Die im Vergleich schlechte Exploration des NC-TCM bestätigt sich auch für den Energieverbrauch, der durchschnittlich mehr als doppelt so hoch liegt wie für das SMO-TCM. Demnach kann vermutet werden, dass die Performance für das NC-TCM eine starke Abhängigkeit von den Strömungsinformationen auf den Zellen aufweist. Im Vergleich mit dem SMO-TCM liegt der Energieverbrauch des NCs-TCM mit 42 % Mehrkosten höher als die Ineffizienzen bezüglich der totalen Explorationszeit mit 37 %.



(a) **(b)**

Abbildung 5.32.: Terrain Covering Modelle - Abdeckung mit [San Francisco|32|5]. Darstellung des besten Experimentdurchlaufes. Unter (a) A*-TCM und (b) SMO-TCM.



(a) **(b)**

Abbildung 5.33.: Terrain Covering Modelle - Abdeckung mit [San Francisco|32|5]. Darstellung des besten Experimentdurchlaufes. Unter (a) NC-TCM und (b) NCs-TCM.

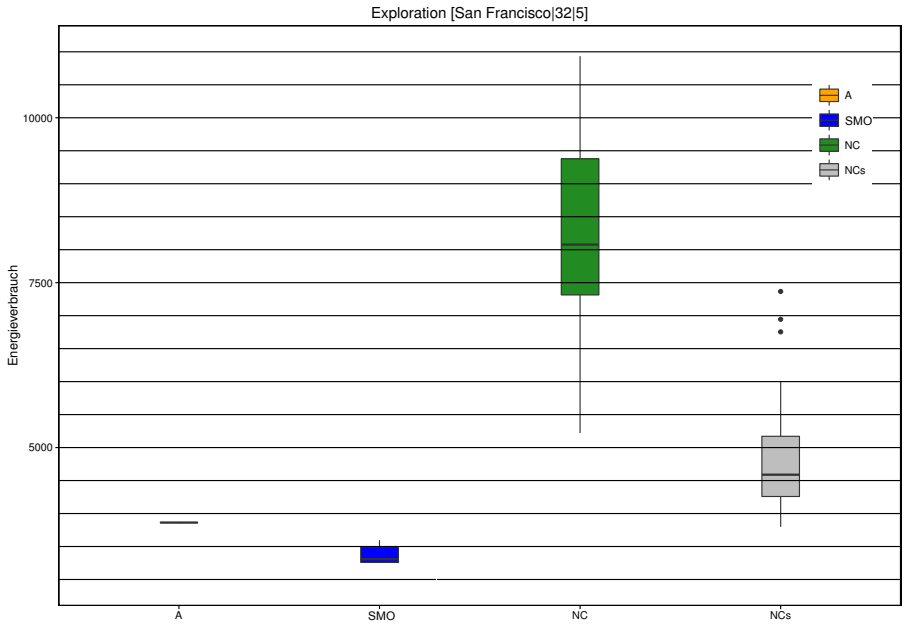


Abbildung 5.34.: Terrain Covering Modelle - Energieverbrauch mit [San Francisco|32|5]. Darstellung der Mittelwerte über alle Experimentdurchläufe.

5.3. Zusammenfassung

Grundsätzlich kann für alle vier TCMs eine vollständige Exploration der Systemumgebung garantiert werden, falls mindestens ein Agent funktionsfähig ist. Hierbei basieren die TCMs auf einem dezentralen Ansatz zur Koordination der Agenten. Ebenfalls kann eine vollständige Exploration der Systemumgebung für eine eingeschränkte oder ausgefallene Kommunikation gewährleistet werden. Nichtsdestotrotz sind Performanceunterschiede hinsichtlich der Untersuchungsgegenstände erkennbar. Resultierend können nachfolgend erste Rückschlüsse auf einen möglichen Einsatz der TCMs und Anwendungen gezogen werden.

Die Vorteile des SMO-TCM liegen vor allem in der auktionsbasierten Verteilung von Aufgaben und der optimierten Bestimmung von Routen unter Berücksichtigung von Unsicherheiten. Es ist gezeigt worden, dass die mehrkriterielle Routenplanung unter Unsicherheiten zu einer Performancesteigerung führen kann und somit zur Minimierung des Energieverbrauches beiträgt. Damit eine optimierte Routenplanung vorgenommen werden kann, müssen genügend Strömungsinformationen vorliegen. Ein kontinuierlicher Informationsaustausch nimmt daher einen besonderen Stellenwert ein. Das SMO-TCM hat im Vergleich zum NC-TCM und NCs-TCM bei einer Reduzierung der Kommunikationsreichweite unter acht Zellen eine schlechtere Performance aufgezeigt. Des Weiteren nimmt eine eingeschränkte Kommunikation einen negativen Einfluss auf die Teilnahme der Agenten an Auktionen. Aus diesem Grund können unter Umständen die Aufgabenziele nicht an die geeigneten Agenten zugeteilt werden. Insbesondere die hohe Dislozierung der Aufgabenziele entsprechend der Quadtree-Reihenfolge kann bei eingeschränkter Kommunikation zu Ineffizienzen führen. Die Ursache liegt darin begründet, dass die Agenten zwischen den dislozierten Teilregionen springen und resultierend lange Routen befahren müssen. Insbesondere zum Ende der Exploration können auf diesen Routen kaum noch unbekannte Zellen abgedeckt werden. Die gleiche Situation tritt bei einer niedrigen Anzahl von Agenten ein. Ebenfalls ist anzumerken, dass die Approximation

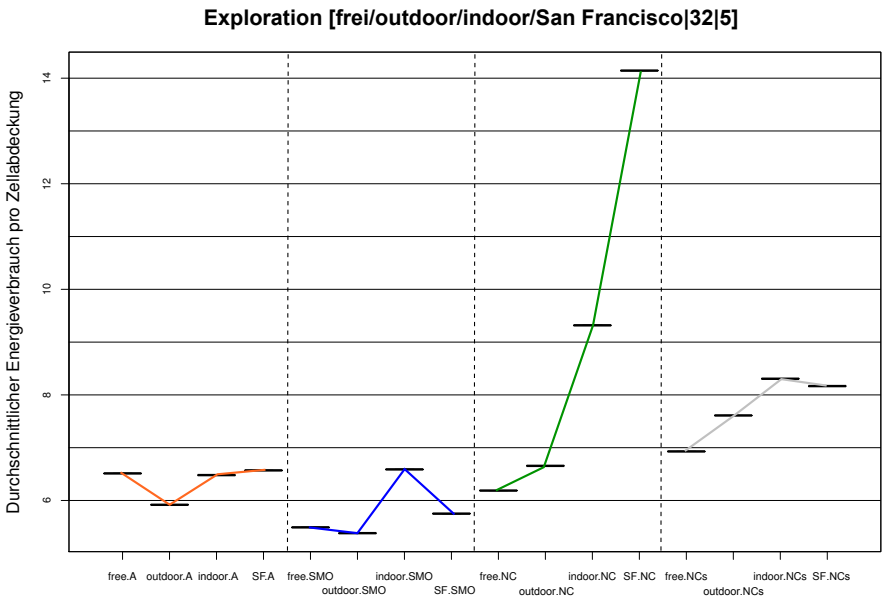


Abbildung 5.35.:]
Terrain Covering Modelle - Energieverbrauch mit [frei/outdoor/indoor/San Francisco|32|5]. Darstellung der Mittelwerte über alle Experimentdurchläufe.

der Strömungsinformationen unbekannter Zellen einen wesentlichen Einfluss auf der Performance nimmt. Wird eine falsche Approximation vorgenommen, dann führen die zuvor optimierten Routen zu ineffizienten, tatsächlichen Ausprägungen. Resultierend ist ein kontinuierlicher Informationsaustausch notwendig, um stets die aktuellen Informationen bezüglich der Strömungskosten und Hindernisse in die Optimierung einfließen lassen zu können.

Ferner zeigt das SMO-TCM eine hohe Robustheit gegenüber unterschiedlichen Hindernisstrukturen und Systemumgebungen. In Abbildung 5.35 ist der normierte, durchschnittliche Energieverbrauch veranschaulicht, der zur Abdeckung einer Zelle aufgewendet werden muss. Gegenüber den unterschiedlichen Systemumgebungen zeigt das SMO-TCM eine vergleichsweise robuste Performance. Grundsätzlich können die Ergebnisse anderer Untersuchungen in [SAB+00, ZJKK05] bestätigt werden, dass die Exploration konkaver Hindernisstrukturen besonders hohe Anforderungen an das TCM stellen.

Weitere Performancesteigerungen können vermutlich durch die Integration einer zusätzlichen Zielfunktion erreicht werden. Neben der Minimierung der approximierten Kosten und Maximierung der Eintrittswahrscheinlichkeit kann die Maximierung unbekannter Zellen auf der Route einbezogen werden. Dadurch können unter Umständen die Ineffizienzen zum Ende der Exploration vermieden werden. Des Weiteren kann dadurch die beschriebene Situation in Abbildung 5.11 untersucht und geeignet in die Routenplanung einbezogen werden. Untersuchungen müssen diesbezüglich noch durchgeführt werden. Des Weiteren kann die Neuplanung von Routen zu einer weiteren Performancesteigerung führen. Eine Neuplanung sollte immer genau dann durchgeführt werden, wenn ein Agent das MANET betritt. Unter Umständen befindet sich der neue Agent im MANET bereits schon in der Nähe von einem übernommenen Aufgabenziel.

In [DEI13] wird eine Untersuchung vorgestellt, bei der zur Laufzeit ein Wechsel zwischen verschiedenen TCMs vorgenommen wird. Es ist möglich, dass die Agenten bis zu einem vordefinierten Abbruchkriterium eine Exploration entsprechend dem NCs-TCM vornehmen. Treten vermehrt Ineffizienzen auf, wird die Exploration mit dem SMO-TCM fortgeführt.

Zusammenfassend ist eine ausreichende Kommunikationsreichweite und Anzahl von Agenten notwendig, um die Vorteile der auktionsbasierten Koordination und optimierten Routenplanung bestmöglich für die Exploration nutzen zu können. Die Einsatzmöglichkeiten des SMO-TCM können demnach vor allem in Anwendungen liegen, die eine stark unstrukturierte Systemumgebung aufweisen. Des Weiteren müssen keine vollständigen Informationen bezüglich der Systemumgebung vorliegen. Hingegen ist eine weitgehend vollständige Kommunikation förderlich. Resultierend sind vor allem Anwendungen im Bereich der autonomen Überwachung von unzugänglichen Gebieten geeignet. Beispielsweise können autonome Landfahrzeuge eingesetzt werden, die ein Naturschutzgebiet überwachen. Dabei können Informationen bezüglich der Lebensweise von Tieren und Vorgehensweisen von Wilderern gesammelt werden [Tew13]. Ein weiteres Anwendungsgebiet liegt in der Exploration von Regionen nach einer Naturkatastrophe [Pic16]. Die mobilen Luft- und Landfahrzeuge können im Verband eingesetzt werden, um beispielsweise Informationen über die humanitäre Versorgung und Infrastruktur zu sammeln [PLS+16]. Die Kommunikation kann beispielsweise in beiden Anwendungen mittels einer Satellitenverbindung sichergestellt werden [KTL+12, PDP14].

6. Zusammenfassung & Ausblick

Der Einsatz von Robotern in Form von unbemannten und autonomen Plattformen unterstützt den Menschen schon heute in unterschiedlichen Anwendungen [PwC14, WS07, Sch11, Rod05, Mah12, Tew13]. Es ist damit zu rechnen, dass sich dieser Trend in der Zukunft fortsetzt und womöglichen verstärken wird [Cha14]. Derzeit werden solche Robotersysteme vor allem in strukturierten Systemumgebungen erfolgreich eingesetzt. Die Systemumgebungen kennzeichnen sich durch eindeutige und wiederkehrende Umgebungsstrukturen. Oftmals liegen vollständige Informationen über die Systemumgebung vor. Zukünftig liegt ein großes Anwendungspotential in komplexeren Anwendungen. Aufgrund unvollständiger oder nicht eindeutiger Umgebungsinformationen nehmen die Herausforderungen bezüglich des erfolgreichen Einsatzes von autonomen Robotersystemen zu. Resultierend müssen geeignete Ansätze entwickelt werden, die beispielsweise die Robustheit, Effizienz und Flexibilität sicherstellen. Es hat sich in der Vergangenheit gezeigt, dass die Verlässlichkeit solcher Robotersysteme maßgeblich für die Ausprägung der Akzeptanz und Einführung ist.

In den meisten Anwendungen wird eine sichere und eigenständige Fortbewegung der Robotersysteme vorausgesetzt. Mit dem Online Terrain Covering Problem wird für einen oder mehrere Agenten eine optimierte Abdeckung/Befahrung einer unbekannten Systemumgebung untersucht. In Kapitel 2 dieser Dissertation ist eine umfassende Taxonomie aus bestehenden Ansätzen entwickelt worden. Die Taxonomie unterscheidet eine Vielzahl von Ansätzen bezüglich verschiedener Gestaltungsdimensionen. Die Gestaltungsdimensionen reichen von grundlegenden Optimierungsansätzen über die Ausgestaltung von Agenten und Sensoren bis hin zur Repräsentation der Systemumgebung. Auf der Grundlage der Veröffentlichungen und zugehöriger Untersuchungen sind die Auswirkungen auf reale Anwendungen und charakteristischen Eigenschaften der Modellierungsansätze erarbeitet worden. Eine geeignete Entscheidungsunterstützung bei der Auswahl von Modellierungsansätzen bietet die Möglichkeit, adaptierte und anwendungsbezogene Terrain Covering Modelle zu erstellen. Hierbei können die Gestaltungsdimensionen modular basierend auf bestehenden Terrain Covering Modellen entwickelt werden. Mit Hilfe der Entscheidungsunterstützung konnte anhand von bestehenden Terrain Covering Modellen eine Einordnung in verschiedene Anwendungsbereiche vorgenommen werden. Des Weiteren kann durch eine Klassifizierung von Terrain Covering Modellen eine bessere Vergleichbarkeit untereinander hergestellt werden.

Die Einordnung der bestehenden Terrain Covering Modelle hat gezeigt, dass derzeit nur sehr wenige Modelle eine optimierte Abdeckung unter Unsicherheiten betrachten. Insbesondere für das Online Terrain Covering Problem sind die Agenten in realen Anwendungen mit einer Vielzahl von Unsicherheiten konfrontiert. In der Dissertation sind fünf verschiedene Ebenen innerhalb von Multiagent Robotic Systems herausgearbeitet worden, in denen Fehlerquellen auftreten können. Hierbei können sich Fehler und Störungen ausgehend von der Sensorik über die Kommunikation und Kooperation akkumulieren. Ferner kann die Effizienz und Robustheit gegenüber der Aufgabenerfüllung nachhaltig gestört werden. Unsicherheiten können die Auswirkungen verstärken. Die

in der Literatur aufgeführten vier verschiedenen Typen von Unsicherheiten sind auf das Multiagent Robotic System übertragen worden. Zur Steigerung der Robustheit sind in dieser Dissertation unterschiedliche Maßnahmen beispielsweise aus den Bereichen der Robotertechnik, des Produktdesigns, der Softwareentwicklung oder Netzwerktechnik zusammengetragen worden. Die Maßnahmen sind in einem neuen Terrain Covering Modell dem SMO-TCM integriert worden.

Das Terrain Covering Problem besteht aus zwei wesentlichen Komponenten. Zum einen muss für eine Verbesserung der Effizienz und Robustheit eine geeignete Koordination und Kooperation zwischen den Agenten organisiert werden. Zum anderen müssen kontinuierlich Routen bestimmt werden, um eine optimierte Exploration der Systemumgebung vornehmen zu können. Demnach muss für reale Anwendungen eine Routenplanung unter Unsicherheiten integriert werden. Um eine optimierte Aufgabenzuteilung unter den Agenten vornehmen zu können, ist eine Vergleichbarkeit der Routen unabdingbar. Bisherige Formulierungen einer robusten Route ermöglichen nur eine bedingte Vergleichbarkeit. Die Unsicherheiten werden in den Nebenbedingungen formuliert und eine Optimierung hinsichtlich der übergeordneten Zielfunktion durchgeführt. Resultierend wird eine robuste Lösung ermittelt. Aussagen über die Robustheit der Routen untereinander sind kaum möglich. In Kapitel 4 dieser Dissertation wird daher ein neuer Lösungsansatz untersucht. Die Unsicherheiten werden als zusätzliche Zielfunktion formuliert. Die Modellierung erlaubt nunmehr eine Vergleichbarkeit aller Lösungen im Lösungsraum. Für das Routenplanungsproblem unter Unsicherheiten besteht das Ziel darin, Routen mit minimalen Kosten und maximaler Eintrittswahrscheinlichkeit zu ermitteln. Die in dieser Dissertation erstmalige Transformation des Stochastic Shortest Path Problem in das Multiobjective Shortest Path Problem (under Uncertainty) kann womöglich für eine Vielzahl von Problemstellungen unter Unsicherheiten vorgenommen werden. Zur Ermittlung der Pareto-optimalen Lösungen wird in dieser Dissertation die BiCriterionAnt Heuristik angepasst und implementiert. Es konnte an einem Beispiel gezeigt werden, dass eine umfangreiche Exploration des Lösungsraumes möglich ist. Ferner ist eine Exploitation besonders geeigneter Lösungen erreicht worden. Die Einführung einer Ordnungsrelation ermöglicht die Vergleichbarkeit der ermittelten Routen und deren konkreten Ausprägungen.

In der Literatur zeigen Untersuchungen, dass die Lösungsgüte von Heuristiken maßgeblich auf der Definition des Heuristik-Parametersets basiert. Die Ausprägung des Heuristik-Parametersets basiert jedoch selbst auf der konkreten Probleminstanz. Für das Online Terrain Covering Problem führen die Agenten autonom eine Routenplanung durch. Resultierend nutzen die Agenten selbstständig eine Heuristik. Vor der Exploration sind jedoch keine konkreten Probleminstanzen bekannt. Es sind verschiedene Möglichkeiten zur Problemlösung aufgezeigt worden. In dieser Dissertation ist die Offline Parameterkonfiguration nach dem IRace Algorithmus implementiert worden. Dadurch ist es möglich, ein geeignetes Heuristik-Parameterset zu ermitteln, dass für einer Vielzahl von möglichen Probleminstanzen eine hohe Lösungsgüte aufweist.

In Kapitel 5 ist abschließend eine Modellierung der Systemumgebung realisiert worden. Die Systemumgebung ist in disjunkte Zellen zerlegt worden. Für jede Zelle liegen Strömungskosten vor. Zu Beginn der Exploration sind weder die Umgebungsstrukturen oder Umgebungsbegrenzungen noch die Strömungskosten bekannt. Im Verlauf der Exploration werden die Strömungskosten für die noch unbekannte Systemumgebung auf Grundlage der bereits gesammelten Informationen approximiert. Hierfür ist ein Diffusionsmodell implementiert worden. Daran anschließend ist die Implementierung des neuen Stochastic Multiobjective-Terrain Covering Modells vorgenommen worden, das die Maß-

nahmen zur Steigerung der Robustheit und das Multiobjective Shortest Path Problem (under Uncertainty) integriert. Des Weiteren sind Vergleichsmodelle implementiert worden. Unter anderem ist ein Terrain Covering Modell erstellt worden, dessen Funktionsweise bis auf die Routenplanung identisch ist. Die Routenplanung berücksichtigt keine Unsicherheiten. Damit kann ein direkter Vergleich und die Auswirkungen einer Routenplanung unter Unsicherheiten dargestellt werden. Es konnte an Beispielen gezeigt werden, dass mit dem SMO-TCM eine Effizienzsteigerung erreicht wird. Des Weiteren zeigen die Untersuchungen ein robustes Verhalten gegenüber unterschiedlichen Hindernisstrukturen. Eine weniger gute Robustheit liegt für eine geringe Kommunikationsreichweite und einer reduzierten Anzahl von Agenten vor. Der Austausch von Strömungsinformationen und bereits abgedeckter Regionen verhindert Ineffizienzen. Des Weiteren kann eine optimierte Aufgabenzuteilung nur dann erfolgen, wenn eine Kooperation zwischen den Agenten sichergestellt ist. Hervorzuheben ist jedoch, dass das SMO-TCM für jede befahrbare Systemumgebung eine vollständige Exploration garantieren kann. Hierfür muss lediglich ein Agent funktionsfähig sein. Eine vollständige Abdeckung kann auch für einen Totalausfall der Kommunikation gewährleistet werden.

Die Lösungsgüte einer implementierten Heuristik zur Bearbeitung des Multiobjective Shortest Path Problem (unter Unsicherheiten) kann gegenüber der Offline Parameterkonfiguration verbessert werden. Mit einer kontinuierlichen Analyse der Systemumgebung können charakteristische Eigenschaften ermittelt werden. Es ist möglich, die Systemumgebung entsprechend der Umgebungszerlegung in einen Graphen zu transformieren. Mit Hilfe der quantitativen Graphentheorie können Eigenschaften, wie beispielsweise die Vernetzung der Zellen ermittelt werden. Entsprechend kann angegeben werden, wie leicht jede Zelle von jeder anderen Zelle erreicht werden kann. Auf der Grundlage solcher quantifizierten Struktureigenschaften kann eine konkrete Auswahl von einem geeigneten Heuristik-Parameterset zur Laufzeit ermittelt werden.

Ein weiterer Untersuchungsbereich sollte sich mit der Auswahl einer konkreten Lösung aus der Menge Pareto-optimaler Lösungen befassen. Bei der Auswahl einer konkreten Lösung sollte der Explorationsfortschritt berücksichtigt werden. Unter Umständen ist es möglich, unterschiedliche Zielfunktionen in Abhängigkeit des Explorationsfortschrittes zu berücksichtigen. Resultierend kann ein angepassteres Verhalten erreicht werden. Es ist sogar möglich, zur Laufzeit ein Wechsel zwischen verschiedenen Terrain Covering Modellen vorzunehmen, um die individuellen Vorteile ausnutzen zu können.

Des Weiteren sollten sich an diese Dissertation weitere Untersuchungen anschließen, die unterschiedliche Koordinations- und Kooperationsformen untersuchen. Der implementiere, auktionsbasierte Ansatz bedarf nach Möglichkeit einer vollständigen Kommunikation, um eine effiziente Lösung des Terrain Covering Problems zu ermitteln. Es existieren Ansätze einer indirekten Kommunikation, die beim Ausfall der Kommunikation ein robusteres Verhalten zeigen.

Zusammenfassend ist mit dieser Dissertation ein Beitrag zum robusten Einsatz von autonomen Robotersystemen geleistet worden. Insbesondere die Ausprägung eines robusten Verhaltens in einer unstrukturierten und unbekannten Systemumgebung stellt eine der größten Herausforderung in der Implementierung von autonomen Agenten dar.

Literaturverzeichnis

[AB97] Arkin, R. C. und G. A. Bekey: *Robot Colonies.* Springer, 1997.

[AB11] Angelo, J. S. und H. J. C. Barbosa: *On Ant Colony Optimization Algorithms for Multi-objective Problems.* In: Ostfeld, A. (Herausgeber): *Ant Colony Optimization: Methods and Applications*, Seiten 53–74. Intech, 2011.

[ABC+09] Applegate, D. L., R. E. Bixby, V. Chvátal, W. Cook, D. G. Espinoza, M. Goycoolea und K. Helsgaun: *Certification of an optimal TSP tour through 85,900 cities.* Operations Research Letters, 37(1):11–15, 2009.

[AHK06] Agmon, N., N. Hazon und G. A. Kaminka: *Constructing spanning trees for efficient multi-robot coverage.* In: *Proceedings of the IEEE International Conference on Robotics and Automation*, Seiten 1698–1703, 2006.

[AL06] Adenso-Diaz, B. und M. Laguna: *Fine-Tuning of Algorithms Using Fractional Experimental Designs and Local Search.* Oper. Res., 54(1):99–114, 2006.

[ARo13] *A Roadmap for U.S. Robotics: From Internet to Robotics.* Computing Community Consortium, 2013.

[ASG07] Alaya, I., C. Solnon und K. Ghédira: *Ant Colony Optimization for Multi-Objective Optimization Problems.* In: *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence*, Seiten 450–457, 2007.

[ASM06] Allen, J. K., C. C. Seepersad und F. Mistree: *A Survey of Robust Design with Applications to Multidisciplinary and Multiscale Systems.* Journal of Mechanical Design, 128(4):832–843, 2006.

[BBC11] Bertsimas, D., D. B. Brown und C. Caramanis: *Theory and Applications of Robust Optimization.* Society for Industrial and Applied Mathematics Review, 53(3):464–501, 2011.

[BCG+08] Bird, K. J., R. R. Charpentier, D. L. Gautier, D. W. Houseknecht, T. R. Klett, J. K. Pitman, T. E. Moore, C. J. Schenk, M. E. Tennyson und C. J. Wandrey: *Circum-Arctic Resource Appraisal: Estimates of Undiscovered Oil and Gas North of the Arctic Circle.* In: *Annual Energy Outlook 2009*, Band USGS Fact Sheet 2008-3049, Washington D.C., USA, 2008. U.S. Geological Survey.

[BE96] Byrne, C. und P. Edwards: *Refinement in agent groups.* In: G.Weiss und S. Sen (Herausgeber): *Adaption and Learning in Multi-Agent Systems*, Band 1042 der Reihe *Lecture Notes in Computer Science*, Seiten 22–39. Springer Berlin Heidelberg, 1996.

[BEFW97] Borenstein, J., H. R. Everett, L. Feng und D. K. Wehe: *Mobile robot positioning: Sensors and techniques.* Journal Field Robotics, 14(4):231–249, 1997.

[Bek06] Bektas, T.: *The multiple traveling salesman problem: an overview of formulations and solution procedures.* Omega, 34(3):209–219, 2006.

[Bel58] Bellman, R.: *On a Routing Problem.* Quarterly of Applied Mathematics, 16:87–90, 1958.

[Ben04] Beni, G.: *From Swarm Intelligence to Swarm Robotics.* In: *Swarm Robotics*, Seiten 1–9. Springer, 2004.

[Beu06] Beume, N.: *Hypervolumen-basierte Selektion in einem evolutionären Algorithmus zur Mehrzieloptimierung.* Technischer Bericht TR06-2-002, Universität Dortmund, 2006.

[Beu09] Beume, N.: *Faster S-Metric Calculation by Considering Dominated Hypervolume as Klee's Measure Problem.* Evolutionary Computation, 17:477–492, 2009.

[Beu11] Beume, N.: *Hypervolume-based Metaheuristics for Multiobjective Optimization.* Doktorarbeit, Universität Dortmund, 2011.

[BF96] Borenstein, J. und L. Feng: *Measurement and correction of systematic odometry errors in mobile robots.* IEEE Transactions on Robotics and Automation, 12(6):869–880, 1996.

[BG88a] Bond, A. H. und L. Gasser: *A Survey of Distributed Artificial Intelligence.* In: *Readings in Distributed Artificial Intelligence.* Morgan Kaufmann, CA, USA, 1988.

[BG88b] Bond, A. H. und L. G. Gasser: *Readings in Distributed Artificial Intelligence.* M. Kaufmann, 1988.

[BH10] Bonet, B. und E. A. Hansen: *Heuristic Search for Planning under uncertainty.* In: Dechter, R., H. Geffner und J. Y. Halpern (Herausgeber): *Heuristics, Probability and Causality A Tribute to Judea Pearl*, Seiten 3–23. College Publication, 2010.

[Bir02] Birattari, M.: *A racing algorithm for configuring metaheuristics.* In: *Proceedings of the Genetic and Evolutionary Computation Conference*, Seiten 11–18. Morgan Kaufmann Publishers, 2002.

[BLR01] Barber, K. S., T.-H. Liu und S. Ramaswamy: *Conflict detection during plan integration for multi-agent systems.* IEEE Transactions on Systems, Man, and Cybernetics, Part B, 31(4):616–628, 2001.

[BM97] Bensaid, N. und P. Mathieu: *A Framework for Cooperation in Hierarchical Multi-Agent Systems.* Mathematical Modeling and Scientific Computing, 8, 1997.

[BM08] Bullen, H. und R. Menezes: *Models for Temporal and Spatial Terrain Coverage.* In: Badica, C. und M. Paprzycki (Herausgeber): *Advances in Intelligent and Distributed Computing*, Band 78 der Reihe *Studies in Computational Intelligence*, Seiten 239–244. Springer, 2008.

[BMF+00] Burgard, W., M. Moors, D. Fox, R. Simmons und S. Thrun: *Collaborative Multi-Robot Exploration.* In: *Proceedings of IEEE International Conference on Robotics & Automation*, 2000.

[BN68] Bellmore, M. und G. L. Nemhauser: *The Traveling Salesman Problem: A Survey.* Operations Research, 16(3):538–558, 1968.

[BNR06] Beume, N., B. Naujoks und G. Rudolph: *Mehrkriterielle Optimierung durch evolutionäre Algorithmen mit S-Metrik-Selektion.* In: *Workshop Computational Intelligence*, Band 16, Seiten 1–10. Universitätsverlag Karlsruhe, 2006.

[Bog08] Bogue, R.: *Swarm intelligence and robotics.* Industrial Robot: An International Journal, 35(6):488–495, 2008.

[Bos07] Bossel, H.: *Systems and Models: Complexity, Dynamics, Evolution, Sustainability.* Books on Demand, 2007.

[BRH00] Butler, Z. J., A. A. Rizzi und R. L. Hollis: *Complete distributed coverage of rectilinear environments.* In: *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*, 2000.

[Bri15] *Brimrose introduces flight capability with launch of HERO.* `http://www.brimrose.com/press/press_2013.html`, Abruf 14.06.2015.

[Bro92] Brooks, R. A.: *Artificial Life and Real Robots.* In: Varela, F. J. und P. Bourgine (Herausgeber): *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, Seiten 3–10, Cambridge, USA, 1992. MIT Press.

[BS03] Batalin, M. und G. S. Sukhatme: *Multi-robot Dynamic Coverage of a Planar Bounded Environment.* Technischer Bericht CRES-03-011, Center for Robotics and Embedded Systems, 2003.

[BSW08] Bachran, T., F. E. Schneider und C. Waal: *Mehrrobotersysteme in der vernetzten Operationsführung.* In: Schmidt, L., J. Grosche und C. M. Schlick (Herausgeber): *Ergonomie und Mensch-Maschine-Systeme*, Seiten 307–329. Springer Berlin Heidelberg, 2008.

[BT91] Bertsekas, D. P. und J. N. Tsitsiklis: *An Analysis of Stochastic Shortest Path Problems.* Mathematics of Operations Research, 16(3), 1991.

[But00] Butler, Z. J.: *Distributed Coverage of Rectilinear Environments.* Doktorarbeit, Carnegie Mellon University, Pennsylvania, 2000.

[BVS06] Bonomi, A., G. Vizzari und M. Sarini: *A Heterogeneous Multi-Agent System for Adaptive Web Applications.* In: *WOA*, Band 204 der Reihe *Proceedings of the CEUR Workshop*, 2006.

[BW89] Beni, G. und J. Wang: *Swarm Intelligence in Cellular Robotic Systems.* In: *Proceedings of NATO Advanced Workshop on Robots and Biological Systems*, Band 102, 1989.

[BWDA12] Biber, P., U. Weiss, M. Dorna und A. Albert: *Navigation system of the autonomous agricultural robot "BoniRob".* In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[BYBS09] Birattari, M., Z. Yuan, P. Balaprakash und T. Stützle: *F-Race and iterated F-Race: An overview.* Technischer Bericht TR/IRIDIA/2009-018, IRIDIA, Université Libre de Bruxelles, Belgium, 2009.

[Can88] Canny, J. F.: *The complexity of robot motion planning.* MIT Press, Cambridge, USA, 1988.

[Cap08] Capaccio, T.: *Iran Mining of Strait Would Pose Great Challenge, Analysis Says.* `http://www.bloomberg.com/apps/news?pid=newsarchive&sid=aUPHEkn.0f50`, 2008.

[CARL00] Choset, H., E. Acar, A. A. Rizzi und J. Luntz: *Exact cellular decompositions in terms of critical points of morse functions.* Seiten 2270–2277, 2000.

[Cas95] Castelfranchi, C.: *Guarantees for Autonomy in Cognitive Agent Architecture.* In: Wooldridge, M. J. und N. R. Jennings (Herausgeber): *Proceedings of the ECAI-94 Workshop on Agent Theories, architectures and languages: Intelligent Agents*, Band 890 der Reihe *LNAI*, Seiten 56–70. Springer-Verlag, 1995.

[Cas06] Castro, L. N. de: *Fundamentals of natural computing: basic concepts, algorithms, and applications.* Chapman & Hall/CRC, 2006.

[CATM96] Chen, W., J. K. Allen, K.-L. Tsui und F. Mistree: *A Procedure For Robust Design: Minimizing Variations Caused By Noise Factors And Control Factors.* ASME Journal of Mechanical Design, 118:478–485, 1996.

[CC95] Conte, R. und C. Castelfranchi: *Cognitive and Social Action.* Routledge, 1995.

[CEN15] CENETIX: *Center for Network Innovation and Experimentation - Mission.* `http://cenetix.nps.edu/cenetix/`, 2015.

[CFK97] Cao, Y. U., A. S. Fukunaga und A. Kahng: *Cooperative Mobile Robotics: Antecedents and Directions.* Autonomous Robots, 4(1):7–27, 1997.

[CH94] Chen, Z. und C. M. Huang: *Terrain Exploration of a Sensor-Based Robot Moving Among Unknown Obstacles of Polygonal Shape.* Robotica, 12(1):33–44, 1994.

[Cha14] Chatterjee, N.: *World Industrial Robotics Market – Opportunities and Forecasts, 2013–2020.* Technischer Bericht CO 14214, 2014.

[Cho01] Choset, H.: *Coverage for robotics - A survey of recent results.* Annals of Mathematics and Artificial Intelligence, 31(1-4):113–126, 2001.

[Cho05] Choi, H.-J.: *A Robust Design Method for Model and Propagated Uncertainty.* Doktorarbeit, Georgia Institute of Technology, 2005.

[CJ07] Czarnowski, I. und P. Jedrzejowicz: *An Agent-Based Approach to the Multiple-Objective Selection of Reference Vectors.* In: Perner, Petra (Herausgeber): *Machine Learning and Data Mining in Pattern Recognition*, Band 4571 der Reihe *Lecture Notes in Computer Science*, Seiten 117–130. Springer Berlin / Heidelberg, 2007.

[CL90] Cohen, P. R. und H. J. Levesque: *Intention is Choice with Commitment.* Artificial Intelligence, 42(2-3):213–261, 1990.

[CL94] Chen, Y.-L. und C.-C. Liu: *Multiobjective VAr planning using the goal-attainment method.* IEEE Proceedings on Generation, Transmission and Distribution, 141(3):227–232, 1994.

[CM03a] Carlson, J. und R. R. Murphy: *Reliability analysis of mobile robots.* In: *Proceedings of the IEEE International Conference on Robotics and Automation*, Band 1, Seiten 274–281, 2003.

[CM03b] Casper, J. und R. R. Murphy: *Human-robot Interactions During the Robot-assisted Urban Search and Rescue Response at the World Trade Center.* Trans. Sys. Man Cyber. Part B, 33(3):367–385, 2003.

[CM05] Carlson, J. und R. R. Murphy: *How UGVs physically fail in the field.* IEEE Transactions on Robotics, 21(3):423–437, 2005.

[CN01] Choset, H. und K. Nagatani: *Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization.* IEEE Transactions on Robotics and Automation, 17(2):125–137, 2001.

[CNN12] CNN: *Another deadly earthquake shakes northern Italy.* `http://edition.cnn.com/2012/05/29/world/europe/italy-earthquake/`, 2012.

[Cox91] Cox, I. J.: *Blanche: Position Estimation for an Autonomous Robot Vehicle.* In: Iyengar, S. S. und A. Elfes (Herausgeber): *Autonomous Mobile Robots: Control, Planning, and Architecture (Vol. 2)*, Seiten 285–292. IEEE Computer Society Press, Los Alamitos, CA, 1991.

[CP97] Choset, H. und P. Pignon: *Coverage Path Planning: The Boustrophedon Decomposition.* In: *Proceedings of the International Conference on Field and Service Robotics*, 1997.

[CPL04] Coello, C. A. C., G. T. Pulido und M. S. Lechuga: *Handling multiple objectives with particle swarm optimization.* IEEE Transactions on Evolutionary Computation, 8(3):256–279, 2004.

[ČS28] Čapek, K. und P. Selver (Übersetzung): *R. U. R.: (Rossum's Universal Robots) : a Play in Three Acts and an Epilogue.* Humphrey Milford, 1928.

[CS89] Current, J. R. und D. A. Schilling: *The covering salesman problem.* Transportation Science, 23:208–213, 1989.

[Cur81] Current, J. R.: *Multi-objective design of Transportation Networks.* Doktorarbeit, The Johns Hopkins University, Baltimore, 1981.

[CYW10] Chen, J., Y. Yang und L. Wei: *Research on the Approach of Task Decomposition in Soccer Robot System.* In: *Proceedings of the 2010 International Conference on Digital Manufacturing & Automation*, Band 2, Seiten 284–289, DC, USA, 2010. IEEE Computer Society.

[Das00] Das, I.: *Robustness optimization for constrained nonlinear programming problems.* Engineering Optimization, 32(5):585–618, 2000.

[Dc04] Dorigo, M. und E. Şahin: *Guest Editorial.* Autonomous Robots, 17(2-3):111–113, 2004.

[DCG99] Dorigo, M., G. Di Caro und L. M. Gambardella: *Ant algorithms for discrete optimization.* Artifical Life, 5:137–172, 1999.

[DD97] Das, I. und J. E. Dennis: *A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems.* Structural and Multidisciplinary Optimization, 14(1):63–69, 1997.

[Deb01] Deb, K.: *Multi-Objective Optimization Using Evolutionary Algorithms.* John Wiley & Sons, Inc., NY, USA, 2001.

[DEI13] Dervisi, M., O.B. Efremides und D. P. Iracleous: *Terrain Coverage Ant Algorithms: The Random Kick Effect.* International Journal of Advanced Computer Science and Applications, 4(10), 2013.

[DES14] Dehmer, M. und F. Emmert-Streib: *Quantitative Graph Theory: Mathematical Foundations and Applications.* Discrete Mathematics and Its Applications. Taylor & Francis, 2014.

[DFJ54] Dantzig, G. B., D. R. Fulkerson und S. M. Johnson: *Solution of a Large-Scale Traveling-Salesman Problem.* Operations Research, 2:393–410, 1954.

[DFJN97] Doran, J. E., S. Franklin, N. R. Jennings und T. J. Normant: *On cooperation in multi-agent systems.* The Knowledge Engineering Review, 12:309–314, 9 1997.

[DG97] Dorigo, M. und L. M. Gambardella: *Ant colony system: a cooperative learning approach to the traveling salesman problem.* IEEE Transactions on Evolutionary Computation, 1(1):53–66, 1997.

[DGK05] Di Marzo Serugendo, G., M.-P. Gleizes und A. Karageorgos: *Self-organization in Multi-agent Systems.* The Knowledge Engineering Review, 20(2):165–189, 2005.

[DHR03] Doerner, K., R. Hartl und M. Reimann: *Are COMPETants more competent for problem solving? – the case of a multiple objective transportation problem.* Central European Journal of Operations Research, 11(2):115–141, 2003.

[Dia04] Dias, M. B.: *Traderbots: A New Paradigm for Robust and Efficient Multirobot Coordination in Dynamic Environments.* Doktorarbeit, PA, USA, 2004.

[Dig99] Dignum, F.: *Autonomous agents with norms.* Artificial Intelligence and Law, 7(1):69–79, 1999.

[Dij59] Dijkstra, E. W.: *A note on two problems in connexion with graphs.* Numerische Mathematik, 1(1):269–271, 1959.

[DJ02] Deb, K. und S. Jain: *Running performance Metrics for evolutionary multi-objective optimization.* Technischer Bericht, Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology, 2002.

[DJMW96] Dudek, G., M. R. M. Jenkin, E. Milios und D. Wilkes: *A taxonomy for multi-agent robotics.* Autonomous Robots, 3(4):375–397, 1996.

[DJW93] Dudek, G., M. R. M. Jenkin und D. Wilkes: *A taxonomy for swarm robots.* In: *Proceeding of the IEEE/RSJ International Conference on Intelligent Robotic Systems*, Seiten 441–447, 1993.

[DL89] Durfee, E. H. und V. R. Lesser: *Distributed Artificial Intelligence (Vol. 2).* Kapitel Negotiating task decomposition and allocation using partial global planning, Seiten 229–243. Morgan Kaufmann Publishers Inc., CA, USA, 1989.

[DM06] Dimopoulos, Y. und P. Moraitis: *Multi-Agent Coordination and Cooperation through Classical Planning.* In: *Proceedings of the IEEE Conference on Intelligent Agent Technology*, Seiten 398–402. IEEE Computer Society, 2006.

[DMC96] Dorigo, M., V. Maniezzo und A. Colorni: *Ant System: Optimization by a Colony of Cooperating Agents.* Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 26(1):29–41, 1996.

[Dor92] Dorigo, M.: *Optimization, Learning and Natural Algorithms (in Italian).* Doktorarbeit, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy, 1992.

[DS01] Dias, M. B. und A. Stentz: *A Market Approach to Multirobot Coordination.* Technischer Bericht CMU-RI-TR-01-26, Robotics Institute, PA, USA, 2001.

[DS03a] Dias, M. B. und A. Stentz: *TraderBots: A Market-Based Approach for Resource, Role, and Task Allocation in Multirobot Coordination.* Technischer Bericht CMU-RI -TR-03-19, Robotics Institute, PA, USA, 2003.

[DS03b] Dorigo, M. und T. Stützle: *The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances.* In: Hillier, F. S., F. Glover und G. Kochenberger (Herausgeber): *Handbook of Metaheuristics*, Band 57 der Reihe *International Series in Operations Research & Management Science*, Seiten 250–285. Springer New York, 2003.

[DS04] Dorigo, M. und T. Stützle: *Ant Colony Optimization.* Bradford Company, MA, USA, 2004.

[Dye02] Dyer, F. C.: *The biology of the dance language.* Annual Review of Entomology, 47:917–949, 2002.

[EHL+06] Evert, F. K. van, G. W. A. M. van der Heijden, L. A. P. Lotz, G. Polder, A. Lamaker, A. de Jong, M. C. Kuyper, E. J. K. Groendijk, J. J. Neeteson und T. van der Zalm: *A Mobile Field Robot with Vision-Based Detection of Volunteer Potato Plants in a Corn Crop.* Weed Technology, 20(4):853–861, 2006.

[EJ02] Excelente-Toledo, C. B. und N. R. Jennings: *Learning to select a coordination mechanism.* In: *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 3*, Seiten 1106–1113, New York, NY, USA, 2002. ACM.

[Ele10] Electrolux Group: *History 2000–2009.* `http://group.electrolux.com/en/history-2000-2009-770/`, 2010.

[Elf87] Elfes, A.: *Sonar-based real-world mapping and navigation.* IEEE Journal of Robotics and Automation, 3(3):249–265, 1987.

[Ell07] Elliot, M.: *Stigmergic Collaboration: The Evolution of Group Work.* M/C Journal, 9(2), 2007.

[ER93] Ephrati, E. und J. S. Rosenschein: *Multi-Agent Planning as a Dynamic Search for Social Consensus.* In: Bajcsy, R. (Herausgeber): *Proceedings of the International Joint Conferences on Artificial Intelligence*, Seiten 423–431. Morgan Kaufmann, 1993.

[FBKT00] Fox, D., W. Burgard, H. Kruppa und S. Thrun: *A Probabilistic Approach to Collaborative Multi-Robot Localization.* Autonomous Robots, 8(3):325–344, 2000.

[FC12] Fowlkes, W. Y. und C. M. Creveling: *Engineering Methods for Robust Product Design: Using Taguchi Methods in Technology and Product Development.* Engineering Process Improvement. Prentice Hall, 2012.

[Fer94] Ferrell, C.: *Failure Recognition and Fault Tolerance of an Autonomous Robot.* Adaptive Behavior, 2(4):375–398, März 1994.

[Fer99] Ferber, J.: *Multi-agent systems: an introduction to distributed artificial intelligence.* Addison-Wesley, 1999.

[FIN04] Farinelli, A., L. Iocchi und D. Nardi: *Multirobot systems: a classification focused on coordination.* IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 34(5):2015–2028, 2004.

[Flo62] Floyd, R. W.: *Algorithm 97: Shortest Path.* Communications of the ACM, 5(6):345, 1962.

[For56] Ford, L. R.: *Network Flow Theory.* Report P-923, The Rand Corporation, 1956.

[FP00] Fiorini, P. und E. Prassler: *Cleaning and Household Robots: A Technology Survey.* Autonomous Robots, 9(3):227–235, 2000.

[Fra16] Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme IAIS: *NIFTi - Mobile Roboter sollen Rettungskräfte im Einsatz unterstützen.* `http://www.iais.fraunhofer.de/nifti.html`, Abruf 02.04.2016.

[FTL07] Ferranti, E., N. Trigoni und M. Levene: *Brick and Mortar: an on-line multi-agent exploration algorithm.* In: *IEEE International Conference on Robotics and Automation*, Seiten 761–767, 2007.

[Gal88] Galliers, J. R.: *A Theoretical Framework for Computer Models of Cooperative Dialogue, Acknowledging Multi-Agent Conflict.* Doktorarbeit, Open University, England, 1988.

[GCW97] Gen, M., R. Cheng und Q. Wang: *Genetic algorithms for solving shortest path problems.* In: *Proceedings of the IEEE International Conference on Evolutionary Computation*, Seiten 401–406, 1997.

[GF05] Ge, S. S. und C.-H. Fua: *Complete Multi-Robot Coverage of Unknown Environments with Minimum Repeated Coverage.* In: *Proceedings of the IEEE International Conference on Robotics and Automation*, Seiten 715–720, 2005.

[GHA+08] Ghoul, S. E., A. S. Hussein, M. S. Abdel-Wahab, U. Witkowski und U. Rückert: *A Modified Multiple Depth First Search Algorithm for Grid Mapping Using Mini-Robots Khepera.* Journal of Computing Science and Engineering, 2(4):321–338, 2008.

[GK94] Genesereth, M. R. und S. P. Ketchpel: *Software agents.* Communication ACM, 37(7), 1994.

[GKM10] Goerzen, C., Z. Kong und B. Mettler: *A Survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance.* Journal on Intelligence Robotics Systems, 57(1-4):65–100, 2010.

[GM04a] Gerkey, B. und M. Mataric: *Are (explicit) multi-robot coordination and multi-agent coordination really so different?* In: *Proceedings of the AAAI Spring Symposium on Bridging the Multi-Agent and Multi-Robotic Research Gap*, Seiten 1–3, CA, USA, 2004.

[GM04b] Gerkey, B. P. und M. J. Mataric: *A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems.* The International Journal of Robotics Research, 23(9):939–954, 2004.

[GM07] Gabrel, V. und C. Murat: *Robust shortest path problems.* In: Paschos, V. und B. Roy (Herausgeber): *Robustness in OR-DA*, Seiten 71–93. Université Paris IX - Dauphine, 2007. Annales du Lamsade No7.

[GNR+12] Golden, B., Z. Naji-Azimi, S. Raghavan, M. Salari und P. Toth: *The Generalized Covering Salesman Problem.* INFORMS Journal on Computing, 24(4):534–553, 2012.

[Goe14] Goebel, J.: *Öl-Katastrophen schädigen Fischherzen.* `http://www.welt.de/wissenschaft/article126157972/Oel-Katastrophen-schaedigen-Fischherzen.html`, 2014.

[GR01] Gabriely, Y. und E. Rimon: *Spanning-tree based coverage of continuous areas by a mobile robot.* Annals of Mathematics and Artificial Intelligence, 31(1-4):77–98, 2001.

[Gra59] Grassé, P.-P.: *La reconstruction du nid et les coordinations interindividuelles chez Bellicositermes natalensis et Cubitermes sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs.* Insectes Sociaux, 6(1):41–80, 1959.

[Gri01] Gribble, S. D.: *Robustness in Complex Systems.* In: *HotOS*, Seiten 21–26. IEEE Computer Society, 2001.

[Gro98] Grove, K.: *Tides and currents in San Francisco Bay.* `http://geosci.sfsu.edu/courses/geol102/ex6.html`, 1998.

[GS86] Gavish, B. und K. Srikanth: *An optimal solution method for large-scale multiple traveling salesmen problems.* Operations Research: the journal of the Operations Research Society of America, 34:698–717, 1986.

[GSKP07] Groen, F. C. A., M. T. J. Spaan, J. R. Kok und G. Pavlin: *Real World Multi-agent Systems: Information Sharing, Coordination and Planning.* In: Cate, B. D. und H. W. Zeevat (Herausgeber): *Logic, Language, and Computation*, Band 4363 der Reihe *Lecture Notes in Computer Science*, Seiten 154–165. Springer Berlin Heidelberg, 2007.

[Gui11] Guizzo, E.: *Japan Earthquake: Robots Help Search For Survivors.* `http://spectrum.ieee.org/automaton/robotics/industrial-robots/japan-earthquake-robots-help-search-for-survivors`, 2011.

[GWD12] Guruprasad, K. R., Z. Wilson und P. Dasgupta: *Complete Coverage of an Initially Unknown Environment by Multiple Robots using Voronoi Partition.* In: *Proceedings of the International Conference on Advances in Control and Optimization in Dynamical Systems*, 2012.

[GZKT08] Grift, T., Q. Zhang, N. Kondo und K.C. Ting: *A review of automation and robotics for the bio-industry.* Journal of Biomechatronics Engineering, 1(1):37–54, 2008.

[Har14] Harlan, C.: *Search teams will send unmanned submarine to look for missing Malaysia Airlines flight.* `http://www.washingtonpost.com/world/search-teams-will-send-unmanned-submarine-to-look-for-missing-malaysia-airlines-flight/2014/04/14/5047f2e4-c392-11e3-bcec-b71ee10e9bc3\_story.html`, 2014.

[HAS12] Habib, A., M. S. Alam und N. H. Siddique: *Optimizing coverage performance of multiple random path-planning robots.* Paladyn, 3(1):11–22, 2012.

[HCV09] Hoog, J. de, S. Cameron und A. Visser: *Role-Based Autonomous Multi-Robot Exploration.* In: *Proceedings of the International Conference on Advanced Cognitive Technologies and Applications*, 2009.

[Hel93] Helton, J. C.: *Uncertainty and sensitivity analysis techniques for use in performance assessment for radioactive waste disposal.* Reliability Engineering & System Safety, 42(2-3):327–367, 1993.

[HF56] Hall, A. D. und R. E. Fagen: *Definition of System.* In: *General systems I: Yearbook of the Society for the Advancement of General Systems Theory.* The Society, 1956.

[HJ98] Hansen, M. P. und A. Jaszkiewicz: *Evaluating the quality of approximations to the non-dominated set.* Technischer Bericht IMM-REP-1998-7, 1998.

[HK05] Hazon, N. und G. A. Kaminka: *Redundancy, Efficiency and Robustness in Multi-Robot Coverage*, 2005.

[HM79] Hwang, C. L. und A. Masud: *Multiple Objective Decision Making - Methods and Applications. A State-of-the-Art Survey.* Springer-Verlag Berlin Heidelberg, 1979.

[HNR68] Hart, P. E., N. J. Nilsson und B. Raphael: *A Formal Basis for the Heuristic Determination of Minimum Cost Paths.* IEEE Transactions on Systems Science and Cybernetics, 4(2):100–107, 1968.

[HQF07] He, F., H. Qi und Q. Fan: *An evolutionary algorithm for the multi-objective shortest path problem.* In: *Proceedings of the Conference on Intelligent Systems and Knowledge Engineering*, 2007.

[HR12] Hutchinson, D. und L. Robbins: *Arctic Expedition Reaches 88.5 Degrees North Latitude-Fourth Joint U.S.-Canada Survey for Purpose of Delineating Extended Continental Shelf.* Sound Waves – Coastal and Marine Research news from Across the USGS, 2012.

[HS99] Herlihy, M. und N. Shavit: *The Topological Structure of Asynchronous Computability.* 46(6):858–923, 1999.

[HS07] H.-G. Beyer und B. Sendhoff: *Robust optimization - A comprehensive survey.* Computer Methods in Applied Mechanics and Engineering, 196(33-34):3190–3218, 2007.

[HTL96] Hert, S., S. Tiwari und V. Lumelsky: *A Terrain-Covering Algorithm for an AUV.* Autonomous Robots, 3:91–119, 1996.

[Hua01] Huang, W. H.: *Optimal line-sweep-based decompositions for coverage algorithms.* In: *Proceedings of IEEE International Conference on Robotics and Automation*, Band 1, Seiten 27–32, 2001.

[HZ91] Hou, E. S. H. und D. Zheng: *Hierarchical path planning with hexagonal decomposition.* In: *Proceedings of the IEEE International Conference on Decision Aiding for Complex Systems*, Band 2, Seiten 1005–1010, 1991.

[IMM01] Iredi, S., D. Merkle und M. Middendorf: *Bi-Criterion Optimization with Multi Colony Ant Algorithms.* In: *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, EMO '01, Seiten 359–372, London, UK, 2001. Springer-Verlag.

[INS00] Iocchi, L., D. Nardi und M. Salerno: *Reactivity and Deliberation: A Survey on Multi-Robot Systems.* In: Hannebauer, M., J. Wendler und E. Pagello (Herausgeber): *Balancing Reactivity and Social Deliberation in Multi-Agent Systems*, Band 2103 der Reihe *Lecture Notes in Computer Science*, Seiten 9–34. Springer, 2000.

[IRG98] Isukapalli, S. S., A. Roy und P. G. Georgopoulos: *Stochastic Response Surface Methods (SRSMs) for Uncertainty Propagation: Application to Environmental and Biological Systems.* Risk Analysis, 18(3):351–363, 1998.

[IS89] Ioannidis, Y. E. und T. K. Sellis: *Conflict Resolution of Rules Assigning Values to Virtual Attributes.* In: *Proceedings of the International Conference on Management of Data*, Seiten 205–214, NY, USA, 1989. ACM.

[Jam06] Jamali, M.: *Learning to Solve Stochastic Shortest Path Problems.* Technischer Bericht No. 2006-02, Web Intelligence Web Laboratory, Sharif University of Technology, Theran, 2006.

[Jen93] Jennings, N. R.: *Commitments and conventions: The foundation of coordination in multi-agent systems.* The Knowledge Engineering Review, 8(3):223–250, 1993.

[Jen02] Jen, E.: *Stable or robust? What's the difference?*, 2002.

[Jet13] Jet Propulsion Laboratory, California Institute of Technology: *NASA Rover Confirms Mars Origin of Some Meteorites.* `http://www.jpl.nasa.gov/news/news.php?release=2013-298`, 2013.

[Jet14] Jet Propulsion Laboratory, California Institute of Technology: *Mars Exploration Rovers.* `http://mars.jpl.nasa.gov/mer/technology/is_autonomous_mobility.html`, Abruf 07.09.2014.

[JOZF14] Jiang, S., Y. S. Ong, J. Zhang und L. Feng: *Consistencies and Contradictions of Performance Metrics in Multiobjective Optimization.* IEEE Transactions on Cybernetics, 44(12):2391–2404, 2014.

[JRBM02] Jones, H. L., S. M. Rock, D. Burns und S. Morris: *Autonomous Robots in SWAT Applications: Research, Design, and Operations Challenges.* In: *Proceedings of the Symposium for the Association of Unmanned Vehicle Systems International*, FL, USA, 2002. AUVSI, AUVSI.

[JSW98] Jennings, N. R., K. P. Sycara und M. J. Wooldridge: *A Roadmap of Agent Research and Development.* Autonomous Agents and Multi-Agent Systems, 1(1):7–38, 1998.

[JT08] Jiao, L. und Z. Tang: *A Visibility-based Algorithm for Multi-robot Boundary Coverage.* International Journal on Advanced Robotic Systems, 5(1):63–68, 2008.

[KAS12] Karapinar, S., D. Altan und S. Sariel-Talay: *A Robust Planning Framework for Cognitive Robots.* 2012.

[KC02] Knowles, J. und D. Corne: *On metrics for comparing nondominated sets.* In: *Proceedings of the Congress on Evolutionary Computation, 2002*, Band 1, Seiten 711–716, 2002.

[KF95] Kit, E. und S. Finzi: *Software Testing in the Real World: Improving the Process.* ACM Press/Addison-Wesley Publishing Co., NY, USA, 1995.

[Kha86] Khatib, O.: *Real-time Obstacle Avoidance for Manipulators and Mobile Robots.* International Journal of Robotics Research, 5(1):90–98, 1986.

[Kiu89] Kiureghian, A. Der: *Measures of Structural Safety under Imperfect State of Knowledge.* Journal of Structural Engineering, 115(5):1119–1139, 1989.

[KL01] Koenig, S. und Y. Liu: *Terrain coverage with ant robots: a simulation study.* In: *Proceedings of the fifth international conference on Autonomous agents*, Seiten 600–607, New York, USA, 2001. ACM.

[Kle64] Klee, V. L.: *A 'String Algorithm' for Shortest Path in Directed Networks.* Operations Research, 12(3):428–432, 1964.

[KNFU84] Kawamura, N., K. Namikawa, T. Fujiura und M. Ura: *Study on agricultural robot (Part 1).* Journal of the JSAM, 46(3):353–358, 1984.

[KPY01] Karasan, O. E., M. C. Pinar und H. Yaman: *The Robust Shortest Path Problem with Interval Data.* Technischer Bericht, Bilkent University, Department of Industrial Engineering, Ankara, 2001.

[KRS04] Kumar, V., D. Rus und S. Singh: *Robot and sensor networks for first responders.* IEEE Pervasive Computing, 3(4):24–33, 2004.

[KSL01] Koenig, S., B. Szymanski und Y. Liu: *Efficient and inefficient ant coverage methods.* Annals of Mathematics and Artificial Intelligence, 31(1):41–76, 2001.

[KSLO94] Kavraki, L., P. Svestka, J. Latombe und M. Overmars: *Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces.* Technischer Bericht, CA, USA, 1994.

[KTH01] Koenig, S., C. A. Tovey und W. Halliburton: *Greedy Mapping of Terrain.* In: *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, Seiten 3594–3599, 2001.

[KTL+12] Kruijff, G.-J. M., V. Tretyakov, T. Linder, F. Pirri, M. Gianni, P. Papadakis, M. Pizzoli, A. Sinha, E. Pianese, S. Corrao, F. Priori, S. Febrini und S. Angeletti: *Rescue robots at earthquake-hit Mirandola, Italy: A field report.* In: *Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Seiten 1–8, 2012.

[KY97] Kouvelis, P. und G. Yu: *Robust Discrete Optimization and Its Applications.* Nonconvex Optimization and Its Applications. Springer, 1997.

[KYA07] Karlsson, M., F. Ygge und A. Andersson: *Market-Based Approaches to Optimization.* Computational Intelligence, 23(1):92–109, 2007.

[LB92] Lewis, M. A. und G. A. Bekey: *The Behavioral Self-Organization of Nanorobots Using Local Rules.* In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots ans Systems*, 1992.

[LC98] Lim, J. H. und D. W. Choo: *Sonar Based Systematic Exploration Method for an Autonomous Mobile Robot Operating in an Unknown Environment.* Robotica, 16(6):659–667, 1998.

[LGCM10] Legriel, J., C. Le Guernic, S. Cotton und O. Maler: *Approximating the Pareto Front of Multi-criteria Optimization Problems.* In: *TACAS'10*, Seiten 69–83, 2010.

[LIDLSB11] López-Ibáñez, M., J. Dubois-Lacoste, T. Stützle und M. Birattari: *The irace package, Iterated Race for Automatic Algorithm Configuration.* Technischer Bericht TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium, 2011.

[Lim92] Lim, W. Y.: *Self-awareness in mobile robots.* Band 1613, Seiten 58–67, 1992.

[LIS10] López-Ibáñez, M. und T. Stützle: *Proceedings of the 7th International Conference in Swarm Intelligence*, Kapitel Automatic Configuration of Multi-Objective ACO Algorithms, Seiten 95–106. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[LIS12] López-Ibáñez, M. und T. Stützle: *An experimental analysis of design choices of multi-objective ant colony optimization algorithms.* Swarm Intelligence, 6(3):207–232, 2012.

[LK98] Laszlo, A. und S. Krippner: *Systems Theories: Their Origins, Foundations, and Development.* Systems Theories and A Priori Aspects of Perception, Seiten 47–74, 1998.

[LL91] Lin, C.-T. und C. S. G. Lee: *Fault-tolerant reconfigurable architecture for robot kinematics and dynamics computations.* IEEE Transactions on Systems, Man and Cybernetics, 21(5):983–999, 1991.

[LL03] López, F. L. y und M. Luck: *Modelling Norms for Autonomous Agents.* In: *Proceedings of Fourth Mexican International Conference on Computer Science*, Seiten 238–245. IEEE Computer Society, 2003.

[LLH11] Li, P., S.-H. Lee und H.-Y. Hsu: *Review on fruit harvesting method for potential use of automatic fruit harvesting systems.* Procedia Engineering, 23(0):351–366, 2011.

[LNNW98] Lee, L. C., H. S. Nwana, D. T. Ndumu und P. De Wilde: *The Stability, Scalability and Performance of Multi-agent Systems.* BT Technology Journal, 16(3):94–103, 1998.

[Lou83] Loui, R. P.: *Optimal paths in graphs with stochastic or multidimensional weights.* Communication, 26(9):670–676, 1983.

[LTB+10] Linder, T., V. Tretyakov, S. Blumenthal, P. Molitor, D. Holz, R. R. Murphy, S. Tadokoro und H. Surmann: *Rescue robots at the Collapse of the municipal archive of Cologne City: A field report.* In: *Proceedings of the IEEE International Workshop on Safety Security and Rescue Robotics*, Seiten 1–6, 2010.

[LWS07] Liu, W., A. F. T. Winfield und J. Sa: *Modelling Swarm Robotic System: A Case Study in Collective Foraging.* In: *Proceedings of the Conference on Towards Autonomous Robotic Systems*, Seiten 25–32, 2007.

[LY02] Luo, C. und S. X. Yang: *A real-time cooperative sweeping strategy for multiple cleaning robots.* Seiten 660–665, 2002.

[LZZ02] Li, C.-S., C.-Q. Zhang und Z.-L. Zhang: *An agent-based intelligent system for information gathering from World Wide Web environment.* In: *Proceedings of the International Conference on Machine Learning and Cybernetics*, Band 4, Seiten 1852–1857, 2002.

[Mah12] Mahadevan, K.: *The Deepwater Horizon Oil Spill.* `http://www.mote.org/index.php?src=gendocs&ref=Gulf\%20of\%20Mexico\%20Oil\%20Spill&category=Marine\%20Policy\%20Institute`, 2012.

[Mal14] *The search for flight MH370.* `http://www.bbc.com/news/world-asia-26514556`, 2014.

[Mar84] Martins, E. Q. V.: *On a multicriteria shortest path problem.* European Journal of Operational Research, 16(2):236–245, 1984.

[MB08] Menezes, R. und H. Bullen: *A Study of Terrain Coverage Models.* In: *Proceedings of the 2008 ACM Symposium on Applied Computing*, SAC '08, Seiten 1964–1968, NY, USA, 2008. ACM.

[MG08] Montemanni, R. und L. M. Gambardella: *Robust shortest path problems with uncertain costs.* Technischer Bericht IDSIA-03-08, Instituto Dalle Molle di studi sull intelligenza artficiale, Manno, 2008.

[Mie99] Miettinen, K.: *Nonlinear Multiobjective Optimization*, Band 12 der Reihe *International Series in Operations Research and Management Science.* Kluwer Academic Publishers, 1999.

[MK93] Musliner, D. und D. Kortenkamp: *MICE and the Science of Vacuuming.* In: *Proceedings of AAAI Fall Symposium Series: Instantiating Real-World Agents*, Seiten 101–104, 1993.

[MKP+08] Murphy, R. R., J. Kravitz, K. Peligren, J. Milward und J. Stanway: *Preliminary report: Rescue robot at Crandall Canyon, Utah, mine disaster.* In: *Proceedings of the IEEE International Conference on Robotics and Automation*, Seiten 2205–2206. IEEE, 2008.

[MLB07] Maimone, M. W., P. C. Leger und J. J. Biesiadecki: *Overview of the Mars Exploration Rovers Autonomous Mobility and Vision Capabilities.* In: *IEEE International Conference on Robotics and Automation (ICRA)*, 2007.

[MM08] Mason, J. und R. Menezes: *Autonomous algorithms for terrain coverage metrics, classification and evaluation.* In: *IEEE Congress on Evolutionary Computation*, Seiten 1641–1648, Juni 2008.

[MMN11] Mahdavi, I., N. Mahdavi-Amiri und S. Nejati: *Algorithms for biobjective shortest path problems in fuzzy networks.* Iranian Journal of Fuzzy Systems, 8(4):9–37, 2011.

[Mur04] Murphy, R. R.: *Trial by fire [rescue robots].* IEEE Robotics Automation Magazine, 11(3):50–61, 2004.

[Mur14] Murphy, R. R.: *Disaster Robotics.* MIT Press, 2014.

[MY07] Micire, M. J. und H. A. Yanco: *Improving disaster response with multi-touch technologies.* In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Seiten 2567–2568, 2007.

[Nar89] Narasimhan, G.: *A Note on the Hamiltonian Circuit Problem on Directed Path Graphs.* Information Processing Letters, 32(4):167–170, 1989.

[NDDP09] Nallusamy, R., K. Duraiswamy, R. Dhanalakshmi und P. Parthiban: *Optimization of Non-Linear Multiple Traveling Salesman Problem Using K-Means Clustering, Shrink Wrap Algorithm and Meta-Heuristics.* International Journal of Nonlinear Science, 8(4):480–487, 2009.

[Neh03] Nehrlich, W.: *Diskrete Mathematik – Basiswissen für Informatiker.* Fachbuchverlag Leipzig, 2003.

[Neu14] Neuman, S.: *Search For Flight MH370 Reportedly Largest In History.* `http://www.npr.org/blogs/thetwo-way/2014/03/17/290890377/search-for-flight-mh370-reportedly-largest-in-history`, 2014.

[NL04] Nimis, J. und P. C. Lockemann: *Robust multi-agent systems: The transactional conversation approach.* In: *Proceedings of the 1st International Workshop on Safety and Security in Multiagent Systems*, 2004.

[NU00] Nodine, M. H. und A. Unruh: *Constructing Robust Conversation Policies in Dynamic Agent Communities.* In: Dignum, F. und M. Greaves (Herausgeber): *Issues in Agent Communication*, Band 1916 der Reihe *Lecture Notes in Computer Science*, Seiten 205–219. Springer, 2000.

[NZES05] Ngatchou, P., A. Zarei und M. A. El-Sharkawi: *Pareto Multi Objective Optimization.* In: *Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems*, Seiten 84–91, 2005.

[OCPZ03] Oh, J. S., Y. H. Choi, J. B. Park und Y. F. Zheng: *Navigation of cleaning robots using triangular-cell map for complete coverage.* In: *Proceedings of the IEEE International Conference on Robotics and Automation*, Seiten 2006–2011, 2003.

[OM04] Ollero, A. und J. I. Maza: *Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms.* Seiten 211–220, 2004.

[OYKP09] Ozkan, M., A. Yazici, M. Kapanoglu und O. Parlaktuna: *Hierarchical oriented genetic algorithms for coverage path planning of multi-robot teams with load balancing.* In: *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, Seiten 451–458. ACM, 2009.

[Pap77] Papadimitriou, C. H.: *The Euclidean travelling salesman problem is NP-complete.* Theoretical Computer Science, 4(3):237–244, 1977.

[Par96] Pareto, V.: *Cours D'Economie Politique; Volume I and II.* F. Rouge, Lausanne, 1896.

[Par94] Parker, L. E.: *Heterogeneous Multi-Robot Cooperation.* Doktorarbeit, Massachusetts Institute of Technology, 1994.

[Par98] Parker, L. E.: *ALLIANCE: an architecture for fault tolerant multirobot cooperation.* IEEE Robotics and Automation, 14(2):220–240, 1998.

[Par08] Parker, L. E.: *Distributed Intelligence: Overview of the Field and its Application in Multi-Robot Systems.* Journal of Physical Agents, 2(1):5–14, 2008.

[PDP14] Preuss, M., M. Dehmer und S. Pickl: *An Improved Graph-Based Decision Support Approach by employing Graph Measures.* In: *Proceedings of the Networking and Electronic Commerce Research Conference*, Seiten 81–84, 2014.

[PDPH14] Preuss, M., M. Dehmer, S. Pickl und A. Holzinger: *On Terrain Coverage Optimization by Using a Network Approach for Universal Graph-Based Data Mining and Knowledge Discovery.* In: Ślęzak, D., A.-H. Tan, J. F. Peters und L. Schwabe (Herausgeber): *Brain Informatics and Health*, Band 8609 der Reihe *Lecture Notes in Computer Science*, Seiten 564–573. Springer International Publishing, 2014.

[Pea14] Pearlman, J.: *MH370 search becomes most expensive aviation hunt in history, yet still no clues.* `http://www.telegraph.co.uk/news/worldnews/asia/malaysia/10863605/MH370-search-becomes-most-expensive-aviation-hunt-in-history-yet-still-no-clues.html`, 2014.

[PHM03] Proud, R. W., J. J. Hart und R. B. Mrozinski: *Methods for determining the level of autonomy to design into a human spaceflight vehicle: a function specific approach.* Technischer Bericht, NASA Johnson Space Center, TX, USA, 2003.

[Pic16] Pickl, S.: *Persönliche Kommunikation*, Durchführung 18.03.2016.

[Pis00] Piskulic, G.: *Military Police/Engineer Urban Robot (URBOT) Concept Experimentation Program Report, Engineer Module.* 2000.

[PLS+16] Pickl, S., A. Lechleuthner, F. Schultmann, W. Raskob und H. Bracker: *Risiken und Kosten der terroristischen Bedrohungen des schienengebundenen öffentlichen Personenverkehrs.* `http://rikov.w3.rz.unibw-muenchen.de`, Abruf 06.09.2016.

[PMA05] Pires, D. F., A. G. Martins und C. H. Antunes: *A multiobjective model for VAR planning in radial distribution networks based on tabu search.* IEEE Transactions on Power Systems, 20(2):1089–1094, 2005.

[Pre11] Preuss, M.: *Terrain Coverage - Modelle und Algorithmen.* Masterarbeit, Universität der Bundeswehr München, 2011.

[Pre14] Preuss, M.: *A Multi-Objective Online Terrain Coverage Approach.* In: Huisman, D., I. Louwerse und A. P. M. Wagelmans (Herausgeber): *Operations Research Proceedings 2013*, Seiten 347–353. Springer International Publishing, 2014.

[PS90] Pirzadeh, A. und W. Snyder: *A unified solution to coverage and search in explored and unexplored terrains using indirect control.* In: *Proceedings of the IEEE International Conference on Robotics and Automation*, Band 3, Seiten 2113–2119, 1990.

[PSP93] Parkinson, A., C. Sorensen und N. Pourhassan: *A General Approach for Robust Optimal Design.* Journal of Mechanical Design, 115(1):74–80, 1993.

[PwC14] *PwC and Zpryme survey and analysis: 2014 Disruptive Manufacturing Innovations Survey.* Technischer Bericht, PricewaterhouseCoopers International, 2014.

[RDE+04] Rudnick, D. L., R. E. Davis, C. C. Eriksen, D. M. Fratantoni und M. J. Perry: *Undersea gliders for ocean research.* Marine Technology Society Journal, 38(2):73–84, 2004.

[RLNC04] Rekleitis, I. M., V. Lee-Shue, A. P. New und H. Choset: *Limited Communication, Multi-robot Team Based Coverage.* In: *Proceedings of the IEEE International Conference on Robotics and Automation*, Seiten 3462–3468. IEEE, 2004.

[RNRC08] Rekleitis, I. M., A.P. New, E. S. Rankin und H. Choset: *Efficient Boustrophedon Multi-Robot Coverage: an algorithmic approach.* Annals of Mathematics and Artificial Intelligence, 52(2-4):109–142, 2008.

[Rob13] Roberts, J.: *Autonomous robotic systems.* `http://www.csiro.au/Organisation-Structure/Divisions/Computational-Informatics/Autonomous-robotic-systems.aspx`, 2013.

[Rod05] Rodriguez, R. V.: *Small, Unmanned Aircraft Search for Survivors in Katrina Wreckage.* `http://www.nsf.gov/news/news\_summ.jsp?cntn\_id=104453`, 2005.

[ROE09] Ray, P., M. O'Rourke und D. Edwards: *The ontological status of autonomous underwater vehicle fleets.* In: *Proceedings of OCEANS, MTS/IEEE Biloxi on Marine Technology for Our Future: Global and Local Challenges*, 2009.

[Rya07] Ryan, A. J.: *A Multidisciplinary Approach to Complex Systems Design.* Doktorarbeit, The University of Adelaide, 2007.

[SAB+00] Simmons, R. G., D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun und H. L. S. Younes: *Coordination for Multi-Robot Exploration and Mapping.* In: *Proceedings of the Innovative Applications of Artificial Intelligence Conference*, Seiten 852–858, 2000.

[Sah04] Sahinidis, N. V.: *Optimization under uncertainty: state-of-the-art and opportunities.* Computers & Chemical Engineering, 28(6-7):971 – 983, 2004. {FOCAPO} 2003 Special issue.

[Sar07] Sariel, S.: *An Integrated Planning, Scheduling and Execution Framework for Multi-Robot Cooperation and Coordination.* Doktorarbeit, Istanbul Technical University, 2007.

[SB05] Sariel, S. und T. Balch: *Real Time Auction Based Allocation of Tasks for Multi-Robot Exploration Problem in Dynamic Environments.* In: *AAAI Workshop on Integrating Planning into Scheduling*, Seiten 27–33, 2005.

[SB08] Senthilkumar, K. S. und K. K. Bharadwaj: *Spanning tree based terrain coverage by multi robots in unknown environments.* In: *Proceedings of the Annual IEEE India Conference*, Band 1, Seiten 120–125, 2008.

[SB10] Senthilkumar, K. S. und K. K. Bharadwaj: *Multi-Robot Terrain Coverage by Constructing Multiple Spanning Trees Simultaneously.* International Journal of Robotics and Automation, 25(3), 2010.

[SBE08] Sariel, S., T. Balch und N. Erdogan: *Naval Mine Countermeasure Missions.* IEEE Robotics Automation Magazine, 15(1):45–52, 2008.

[Sch95] Schott, J. R.: *Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization.* Diplomarbeit, Massachusetts Institute of Technology, 1995.

[Sch11] Schmidt, F.: *Auch im Schwarm autonom.* http://www.dw-world.de/dw/article/0,15046352,00.html, 2011.

[SDS78] Smith, R. G., R. Davis und Stanford Computer Science Department: *Distributed Problem Solving: The Contract Net Approach.* Reports of the Stanford University. Defense Technical Information Center, 1978.

[SH00] Stützle, T. und H. H. Hoos: *MAX-MIN Ant System.* Future Generation Computer Systems, 16(9):889–914, 2000.

[Sha12] Shaw, D.: *Disaster drones: How robot teams can help in a crisis.* http://www.bbc.com/news/technology-18581883, 2012.

[She92] Sheridan, T. B.: *Telerobotics, Automation, and Human Supervisory Control.* MIT Press, 1992.

[Shi55] Shimbel, A.: *Structure in communication nets.* In: *Proceedings of the Symposium on Information Networks*, Seiten 199–203, 1955.

[SIH91] Sundaresan, S., K. Ishii und D. R. Houser: *Design Optimization for Robustness Using Performance Simulation Programs.* AMSE Journal, 32(1):249–256, 1991.

[Sin11] Singh, M. P.: *Commitments in Multiagent Systems: Some History, Some Confusions, Some Controversies, Some Prospects.* In: Paglieri, F., L. Tummolini, R. Falcone und M. Miceli (Herausgeber): *Essays in Honor of Cristiano Castelfranchi*, Seiten 601–626. College Publications, 2011.

[SIO+95] Sugie, H., Y. Inagaki, S. Ono, H. Aisu und T. Unemi: *Placing Objects with Multiple Mobile Robots - Mutual Help Using Intention Inference.* In: *Proceedings of the International Conference on Robotics and Automation*, Seiten 2181–2186. IEEE Computer Society, 1995.

[SJL08] Sotzing, C. C., N. Johnson und D. M. Lane: *Improving Multi-AUV Coordination with Hierarchical Blackboard-Based Plan Representation.* In: *Proceedings of the 27th Workshop of the UK Planning and Scheduling Special Interest Group*, Seiten 110–117, 2008.

[SK04] Svennebring, J. und S. Koenig: *Building Terrain-Covering Ant Robots: A Feasibility Study.* Autonomous Robots, 16(3):313–332, 2004.

[SKSM08] Sawhney, R., K. M. Krishna, K. Srinathan und M. Mohan: *On Reduced Time Fault Tolerant Paths for Multiple UAVs Covering a Hostile Terrain.* In: *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 3*, International Conference on Autonomous Agents and Multiagent Systems, Seiten 1171–1174. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[SL08a] Shoham, Y. und K. Leyton-Brown (Herausgeber): *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations.* Cambridge University Press, 2008.

[SL08b] Sotzing, C. C. und D. M. Lane: *Improving the Coordination Efficiency of Multiple AUV Operations Using Prediction of Intent.* In: *Proceedings of the 3rd SEAS DTC Technical Conference*, 2008.

[SLID11] Stützle, T., M. López-Ibáñez und M. Dorigo: *A Concise Overview of Applications of Ant Colony Optimization.* In: Cochran, J. J. (Herausgeber): *Wiley Encyclopedia of Operations Research and Management Science*, Band 2, Seiten 896–911. John Wiley & Sons, 2011.

[SPA15] SPARC: *Robotics 2020 Multi-Annual Roadmap for Robotics in Europe.* Horizon 2020 Call ICT–2016 (ICT–25 & ICT–26), 2015.

[SR97] Su, J. und J. E. Renaud: *Automatic differentiation in robust optimization.* AIAA Journal, 35(6):1072–1079, 1997.

[SSVO09] Siciliano, B., L. Sciavicco, L. Villani und G. Oriolo: *Robotics: Modelling, Planning and Control.* Advanced Textbooks in Control and Signal Processing. Springer, 2009.

[STCR04] Saltelli, A., S. Tarantola, F. Campolongo und M. Ratto: *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models.* Halsted Press, NY, USA, 2004.

[Ste05] Steuer, D.: *Statistische Eigenschaften der Multikriteriellen Optimierung mittels Wünschbarkeiten.* Doktorarbeit, Universität Dortmund, 2005.

[Suh01] Suh, N. P.: *Axiomatic Design: Advances and Applications.* MIT-Pappalardo series in mechanical engineering. Oxford University Press, 2001.

[Suh05] Suh, N. P.: *Complexity: theory and applications.* MIT-Pappalardo series in mechanical engineering. Oxford University Press, 2005.

[SVVW80] Silver, E. A., R. Victor, V. Vidal und D. de Werra: *A tutorial on heuristic methods.* European Journal of Operational Research, 5(3):153–162, 1980.

[Syc98] Sycara, K. P.: *The Many Faces of Agents.* Artificial Intelligence Magazine, 19(2):11–12, 1998.

[SYTX06] Sheng, W., Q. Yang, J. Tan und N. Xi: *Distributed multi-robot coordination in area exploration.* Robotics and Autonomous Systems, 54(12):945–955, 2006.

[SZVM11] Salih, N. K., T. Zang, G. K. Viju und A. A. Mohamed: *Autonomic Management for Multi-agent Systems.* International Journal of Computer Science Issues, 8(1), 2011.

[TB96] Thrun, S. und A. Buecken: *Learning Maps for Indoor Mobile Robot Navigation.* Technischer Bericht CMU-CS-96-121, Computer Science Department, PA, USA, 1996.

[TCM01] Tessier, C., L. Chaudron und H. J. Müller: *Conflicting Agents: Conflict Management in Multi-Agent Systems.* Multiagent Systems, Artificial Societies, and Simulated Organizations. Springer, 2001.

[Tew13] Tews, A.: *Stealth robots: spying on wildlife just got real.* `http://theconversation.com/stealth-robots-spying-on-wildlife-just-got-real-11994`, 2013.

[The13] The National Museum of Australia: *Robot tours.* `http://www.nma.gov.au/engage-learn/robot-tours`, 2013.

[Thr98] Thrun, S.: *Learning Metric-Topological Maps for Indoor Mobile Robot Navigation.* Artificial Intelligence, 99(1):21–71, 1998.

[Thr03] Thrun, S.: *Exploring artificial intelligence in the new millennium.* Kapitel Robotic mapping: a survey, Seiten 1–35. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[Thu79] Thuente, D. J.: *Information Linkage between Applied Mathematics and Industry.* Kapitel Two algorithms for shortest paths through multiple criteria networks, Seiten 567–574. Academic Press Inc., 1979.

[TOT10] TOTAL S.A: *Exploration and Production in conditions of extreme cold.* Paris, Frankreich, 2010.

[Tri08] Trianni, V.: *Multi-Robot Systems, Swarm Robotics and Self-Organisation.* In: *Evolutionary Swarm Robotics*, Band 108 der Reihe *Studies in Computational Intelligence*, Seiten 23–46. Springer Berlin Heidelberg, 2008.

[TS10] Tichy, P. und R. J. Staron: *Multi-Agent Technology for Fault Tolerant and Flexible Control.* In: Srinivasan, D. und L. Jain (Herausgeber): *Innovations in Multi-Agent Systems and Applications – 1*, Band 310 der Reihe *Studies in Computational Intelligence*, Seiten 223–246. Springer Berlin Heidelberg, 2010.

[TSS+06] Tichy, P., P. Slechta, R. J. Staron, F. P. Maturana und K. H. Hall: *Multiagent technology for fault tolerance and flexible control.* IEEE Transactions on Systems, Man, and Cybernetics, 36(5):700–704, 2006.

[UMN97] Ulrich, I. R., F. Mondada und J.-d. Nicoud: *Autonomous vacuum cleaner.* Robotics and Autonomous Systems, 19:4–233, 1997.

[Vel99] Veldhuizen, D. A. Van: *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations.* Doktorarbeit, OH, USA, 1999.

[Vel10] Velagapudi, P.: *Information Sharing for Distributed Planning.* In: *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, Band 1, Seiten 1667–1668, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.

[Vid10] Vidal, J. M.: *Fundamentals of Multiagent Systems with NetLogo Examples.* 2010.

[VL00] Veldhuizen, D. A. Van und G. B. Lamont: *On measuring multiobjective evolutionary algorithm performance.* In: *Proceedings of the Congress on Evolutionary Computation, 2000*, Band 1, Seiten 204–211, 2000.

[VR95] VanderHeide, J. und N. S. V. Rao: *Terrain coverage of an unknown room by an autonomous mobile robot.* Technischer Bericht ORNL/TM-13117, Oak Ridge National Laboratory, TN, USA, 1995.

[VS08] Visser, A. und B. A. Slamet: *Including communication success in the estimation of information gain for multi-robot exploration.* In: *6th International Symposium and Workshops on Modeling and Optimization in Mobile Ad Hoc and Wireless Networks*, Seiten 680–687, 2008.

[War62] Warshall, S.: *A Theorem on Boolean Matrices.* ACM, 9(1):11–12, Januar 1962.

[Wei99] Weiss, G. (Herausgeber): *Multiagent systems: a modern approach to distributed artificial intelligence.* MIT Press, Cambridge, USA, 1999.

[Wei13] Weiss, G. (Herausgeber): *Multiagent Systems.* MIT Press, 2013.

[WHH+08] Witkowski, U., M. El Habbal, S. Herbrechtsmeier, A. Tanoto, J. Penders, L. Alboul und V. Gazi: *Ad-hoc Network Communication Infrastructure for Multi-robot Systems in Disaster Scenarios.* In: *Proceedings of IARP/EURON Workshop on Robotics for Risky Interventions and Environmental Surveillance*, 2008.

[Wil75] Williamson, O. E.: *Markets and hierarchies : analysis and antitrust implications – a study in the economics of internal organization.* Free Press, 1 Auflage, 1975.

[Wil11] Willsher, K.: *Air France 447 search finds bodies at bottom of Atlantic.* `http://www.theguardian.com/world/2011/apr/04/air-france-447-search-bodies`, 2011.

[Wis11] Wise, J.: *How Air France 447's Missing Wreckage Was Found - and Why It Took So Long.* `http://www.popularmechanics.com/technology/engineering/robots/how-air-france-447s-missing-wreckage-was-found-5583302`, 2011.

[WJ95] Wooldridge, M. J. und N. R. Jennings: *Intelligent Agents: Theory and Practice.* Knowledge Engineering Review, 10(2):115–152, 1995.

[WLB96] Wagner, I. A., M. Lindenbaum und A. M. Bruckstein: *Smell as a Computational Resource - A Lesson We Can Learn from the Ant.* In: *ISTCS*, Seiten 219–230, 1996.

[WLB99] Wagner, I., M. Lindenbaum und A. M. Bruckstein: *Distributed Covering by Ant-Robots using Evaporating Traces.* IEEE Transactions on Robotics and Automation, 15(5):918–933, 1999.

[WM04] Wong, S. C. und B. A. MacDonald: *Complete Coverage by Mobile Robots Using Slice Decomposition Based on Natural Landmarks.* In: *Proceedings of The Pacific Rim International Conference on Artificial Intelligence*, Seiten 683–692, 2004.

[Woo09] Wooldridge, M. J.: *An Introduction to MultiAgent Systems.* Wiley Publishing, 2nd Auflage, 2009.

[WS07] Wada, K. und T. Shibata: *Living With Seal Robots – Its Sociopsychological and Physiological Influences on the Elderly at a Care House.* IEEE Transactions on Robotics, 23(5):972–980, 2007.

[WSS11] Wong, C. Y., G. Seet und S. K. Sim: *Multiple–Robot Systems for USAR: Key Design Attributes and Deployment Issues.* International Journal of Advanced Robotic Systems, 2011.

[XLF08] Xu, Z., Y. Li und X. Feng: *Constrained Multi-objective Task Assignment for UUVs Using Multiple Ant Colonies System.* In: *Proceedings of the ISECS International Colloquium on Computing, Communication, Control, and Management*, Band 1, Seiten 462–466, 2008.

[YAB+13] Yaghoubi, S., N. A. Akbarzadeh, S. S. Bazargani, S. S. Bazargani, M. Bamizan und M. I. Asl: *Autonomous Robots for Agricultural Tasks and Farm Assignment and Future Trends in Agro Robots.* International Journal of Mechanical & Mechatronics Engineering, 13(3), 2013.

[Yam04] Yamauchi, B.: *PackBot: A Versatile Platform for Military Robotics.* In: *Proceedings of SPIE 5422*, Seiten 228–237, 2004.

[YJA13] Yan, Z., N. Jouandeau und A. Ali Cherif: *A Survey and Analysis of Multi-Robot Coordination.* International Journal of Advanced Robotic Systems, 10(399), 2013.

[YJL08] Y.-S. Jung, K.-W. Lee und B.-H. Lee: *Recent Advances in Multi Robot Systems*, Kapitel Advances in Sea Coverage Methods Using Autonomous Underwater Vehicles (AUVs), Seiten 70–100. 2008.

[YLW+07] Yan, J., C. Li, Z. Wang, L. Deng und D. Sun: *Diversity Metrics in Multi-objective Optimization: Review and Perspective.* In: *IEEE International Conference on Integration Technology, 2007*, Seiten 553–557, 2007.

[Yuh00] Yuh, J.: *Design and Control of Autonomous Underwater Robots: A Survey.* Autonomous Robots, 8(1):7–24, 2000.

[ZBT07] Zitzler, E., D. Brockhoff und L. Thiele: *The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators via Weighted Integration.* In: *Proceedings of the 4th International Conference on Evolutionary Multi-criterion Optimization*, Seiten 862–876, Berlin, Heidelberg, 2007. Springer-Verlag.

[ZDT00] Zitzler, E., K. Deb und L. Thiele: *Comparison of Multiobjective Evolutionary Algorithms: Empirical Results.* Evolutionary Computation, 8(2):173–195, 2000.

[ZFM05] Zang, C., M. Friswell und J. Mottershead: *A review of robust optimal design and its application in dynamics.* Computers & Structures, 83(4-5):315–326, 2005.

[Zit99] Zitzler, E.: *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications.* Doktorarbeit, ETH Zurich, Switzerland, 1999.

[ZJKK05] Zheng, X., S. Jain, S. Koenig und D. Kempe: *Multi-robot forest coverage.* In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Seiten 3852–3857, 2005.

[ZSDT02] Zlot, R., A. Stentz, M. B. Dias und S. Thayer: *Multi-robot exploration controlled by a market economy.* In: *Proceedings of the IEEE International Conference on Robotics and Automation*, Band 3, Seiten 3016–3023, 2002.

[ZT98] Zitzler, E. und L. Thiele: *Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study.* Seiten 292–301. Springer, 1998.

[ZTB10] Zitzler, E., L. Thiele und J. Bader: *On Set-Based Multiobjective Optimization.* IEEE Transactions on Evolutionary Computation, 14(1):58–79, 2010.

[ZTL+03] Zitzler, E., L. Thiele, M. Laumanns, C. M. Fonseca und V. Grunert da Fonseca: *Performance Assessment of Multiobjective Optimizers: An Analysis and Review.* IEEE Transactions on Evolutionary Computation, 7(2):117–132, 2003.

Anhang

Nichts in der Welt steht einzeln
und irgend ein Wirksames muß nicht als Ende,
sondern als Anfang betrachtet werden.
Johann Wolfgang von Goethe (1749–1832)

A. Mehrkriterielle Optimierung

Im Rahmen der MOO werden zwei übergeordnete Phasen durchlaufen. In der ersten Phase werden Pareto-optimale Lösungen erzeugt, die zusammengefasst die Paretomenge darstellen. Anschließend muss der DM eine geeignete Lösung aus der Paretomenge auswählen. In beiden Phasen kann der DM unterstützt werden. In dieser Dissertation wird gezeigt, wie mit Heuristiken die Bestimmung einer approximierten Paretomenge unterstützt werden kann. Des Weiteren wird gezeigt, dass unterschiedliche Faktoren bezüglich der Heuristiken, wie beispielsweise die Ausprägung und Implementierung von Designkomponenten oder die Definition des Parametersets einen Einfluss auf die Güte der Paretomenge nehmen kann. Ferner kann ein Vergleich der Performance unterschiedlicher Heuristiken bei der Auswahl und Definition geeigneter Implementierungen unterstützen. Aus diesem Grund muss ein Vergleich von verschiedenen Paretomengen ermöglicht werden. Weiterhin wird der DM bei der Bestimmung einer geeigneten Paretomenge aus einer Menge von verschiedenen Paretomengen unterstützt. Im nachfolgenden werden verschiedene Metriken beschrieben, die einen Vergleich von Paretomengen anhand deren Abbilder im Zielfunktionsraum, den PFs, ermöglichen [HJ98].

Um die Güte einer PF und der zugehörigen Paretomenge bestimmen zu können, werden in der Literatur verschiedene Metriken vorgeschlagen. Die Metriken bewerten hierbei verschiedene quantitative und qualitative Eigenschaften einer PF. Die nachfolgenden Ausführungen beziehen sich grundsätzlich auf PFs, die für das selbe MOOP zulässig sind und folglich im selben zulässigen Zielfunktionsraum liegen. Grundsätzlich lassen sich unäre und binäre Metriken unterscheiden. Mittels unärer Metriken $M_u : PF \rightarrow \mathbb{R}$ werden Eigenschaften einer PF auf eine reelle Zahl abgebildet und bewertet. Die Abbilder und dementsprechend die Ausprägung einer bestimmten Eigenschaft unterschiedlicher PFs können gemäß einer Ordnungsrelation anschließend miteinander verglichen werden. Hingegen ermöglichen binäre Metriken $M_b : (PF_A, PF_B) \rightarrow \mathbb{R}$ ausschließlich einen paarweisen Vergleich von zwei PFs [Beu06]. In Tabelle 6.1 werden die verschiedenen Metriken gegenübergestellt.

Um einen Vergleich von PFs durchzuführen, definieren Hansen und Jaszkiewicz in [HJ98] das Konzept der Überlegenheit. Demnach ist allgemein die Paretofront PF_A gegenüber der Paretofront PF_B überlegen (besser) genau dann, wenn der DM bezüglich der Präferenzannahmen eine bessere Pareto-optimale Lösung in PF_A findet. Des Weiteren liegt eine Überlegenheit von PF_A vor genau dann, wenn die Lösungen in PF_A nicht schlechter sind als in PF_B und mindestens eine Lösung

in PF_B dominiert wird. Demzufolge basiert das Konzept der Überlegenheit auf der Dominanzrelation für Mengen. Hierbei werden die Definitionen in 4.2.1 und 4.2.1 für einzelne Lösungen auf Mengen von Lösungen überführt. Dementsprechend resultieren wie für die Dominanzrelation auch für die Überlegenheitsrelation verschiedene Ausprägungsgrade, die für zwei Paretofronten PF_A und PF_B gegeben sein können. Nachfolgend wird das Konzept der Überlegenheit in Bezug auf die Dominanzrelation für Mengen angegeben [HJ98]:

Definition 6.1 (Schwache Überlegenheit)**.** : *Die Paretofront PF_A ist PF_B schwach überlegen (Schreibweise: PF_A $\boldsymbol{O}_W$ PF_B) genau dann, wenn $PF_A \neq PF_B$ und die Menge der nicht dominierten Lösungen in $PF_A \cup PF_B$ der PF_A entspricht.*

Resultierend gilt für PF_A $\mathbf{O}_W$ PF_B, dass für jede Lösung in PF_B eine dominierte oder gleiche Lösung in PF_A existiert und mindestens eine Lösung in PF_A nicht in PF_B enthalten ist.

Definition 6.2 (Starke Überlegenheit)**.** : *Die Paretofront PF_A ist PF_B stark überlegen (Schreibweise: PF_A $\boldsymbol{O}_S$ PF_B) genau dann, wenn die Menge der nicht dominierten Lösungen in $PF_A \cup PF_B$ der PF_A entspricht und die Differenzmenge aus PF_B und der Menge der nicht dominierten Lösungen in $PF_A \cup PF_B$ ungleich der leeren Menge ist.*

Demnach gilt für PF_A $\mathbf{O}_S$ PF_B, dass für jede Lösung in PF_B eine schwach dominierte Lösung in PF_A existiert und mindestens eine Lösung in PF_B durch eine Lösung in PF_A stark dominiert wird.

Definition 6.3 (Vollständige Überlegenheit)**.** : *Die Paretofront PF_A ist PF_B vollständig überlegen (Schreibweise: PF_A $\boldsymbol{O}_C$ PF_B) genau dann, wenn die Menge der nicht dominierten Lösungen in $PF_A \cup PF_B$ der PF_A entspricht und die Schnittmenge aus PF_B und der Menge der nicht dominierten Lösungen in $PF_A \cup PF_B$ ungleich der leeren Menge ist.*

Für die vollständige Überlegenheit von PF_A über PF_B muss jede Lösung in PF_B von mindestens einer Lösung in PF_A strikt dominiert werden.
Zusammenfassend gilt $\mathbf{O}_C \subset \mathbf{O}_S \subset \mathbf{O}_W$. Des Weiteren erfüllen die Überlegenheitsrelationen die Eigenschaften der Transitivität und Asymmetrie. Die Überlegenheitsrelationen sind in der Abbildung A.1 veranschaulicht. In [KC02] wird angeführt, dass nicht alle Metriken mit den Überlegenheitsrelationen übereinstimmen.

Die nachfolgenden Metriken sind zu den beschrieben Überlegenheitsrelationen in unterschiedlicher Ausprägung kompatibel. Hansen und Jaszkiewicz definieren in [HJ98] die (schwache) Kompatibilität für eine Metrik in Bezug auf zwei approximierte Paretofronten PF_A und PF_B wie folgt:

Definition 6.4 (Schwache Kompatibilität)**.** : *Eine Metrik $\mathfrak{M}$ ist schwach kompatibel zu einer Überlegenheitsrelation $\boldsymbol{O}_\Phi$ mit $\Phi = \{W, S, C\}$, wenn für jedes beliebige Paar (PF_A, PF_B) mit PF_A $\boldsymbol{O}$ PF_B, $\mathfrak{M}$ zu einer Bewertung führt derart, dass PF_A nicht schlechter als PF_B ist.*

Definition 6.5 (Kompatibilität)**.** : *Eine Metrik $\mathfrak{M}$ ist kompatibel zu einer Überlegenheitsrelation $\boldsymbol{O}_\Phi$ mit $\Phi = \{W, S, C\}$, wenn für jedes beliebige Paar (PF_A, PF_B) mit PF_A $\boldsymbol{O}$ PF_B, $\mathfrak{M}$ zu einer Bewertung führt derart, dass PF_A besser als PF_B ist.*

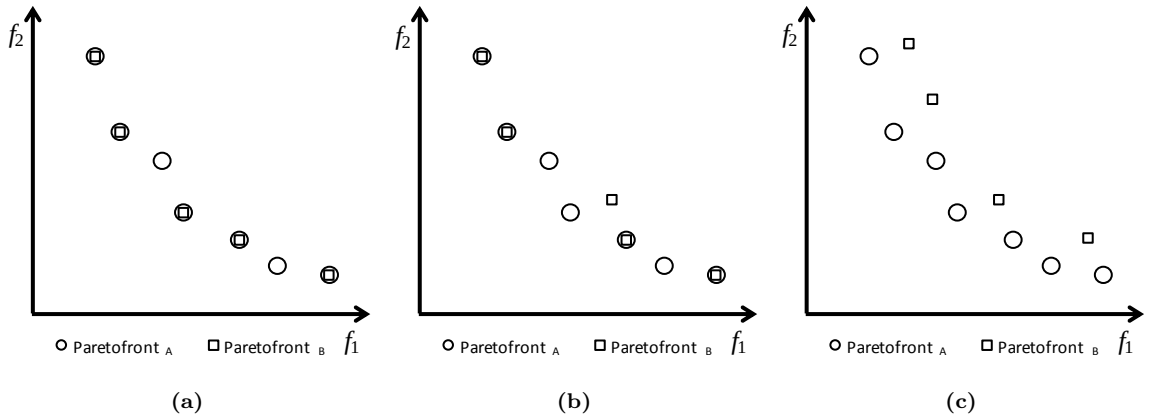


Abbildung A.1.: Überlegenheitsrelationen bezüglich zwei Paretofronten, am Beispiel von einem bikriteriellen Minimierungsproblem. Unter (a) PF_A $\mathbf{O}_W$ PF_B, (b) PF_A $\mathbf{O}_S$ PF_B, (c) PF_A $\mathbf{O}_C$ PF_B (angelehnt an [HJ98], Seiten 7-8).

Demnach ist die Bewertung eine Metrik für PFs nur dann korrekt, wenn sie sich gegenüber den Überlegenheitsmetriken mindestens (schwach) kompatibel verhält. Ferner ist die Vollständigkeit einer Metrik wie folgt angegeben:

Definition 6.6 (Vollständigkeit)**.** : *Eine Metrik $\mathfrak{M}$ ist vollständig zu einer Überlegenheitsrelation $\boldsymbol{O}_\Phi$ mit $\Phi = \{W, S, C\}$, wenn für jedes Paar (PF_A, PF_B) die Überlegenheitsrelation PF_A $\boldsymbol{O}$ PF_B impliziert, dass $\mathfrak{M}$ zu einer Bewertung führt derart, dass PF_A besser als PF_B ist. Entsprechend ist die Bewertung mittels $\mathfrak{M}$ mit der Ordnung PF_A $\boldsymbol{O}$ PF_B kompatibel.*

Ein intuitiver Ansatz zum Vergleich von zwei PFs wird mit der binären $\mathfrak{C}$-Metrik in [ZT98, ZDT00] eingeführt. Mit der $\mathfrak{C}$-Metrik wird das Ziel verfolgt, eine Bewertung der Überlegenheit einer PF anzugeben. Für zwei Paretofronten PF_A, PF_B bildet die $\mathfrak{C}$-Metrik eine Funktion in das Intervall $[0, 1]$ ab. Die $\mathfrak{C}$-Metrik ist wie folgt definiert:

$$\mathfrak{C}(PF_A, PF_B) = \frac{|\{f(\mathbf{x'}) \in PF_B; \exists f(\mathbf{x}) \in PF_A : f(\mathbf{x}) \preceq f(\mathbf{x'})\}|}{|PF_B|} \tag{6.1}$$

Nimmt die $\mathfrak{C}$-Metrik einen Wert von eins an, dann werden alle Lösungen in PF_B von PF_A dominiert oder sind gleich. Entsprechend bedeutet $\mathfrak{C}(PF_A, PF_B) = 0$, dass keine Lösung in PF_B dominiert wird. Für die $\mathfrak{C}$-Metrik gilt nicht zwangsläufig $\mathfrak{C}(PF_A, PF_B) = 1 - \mathfrak{C}(PF_B, PF_A)$. Aus diesem Grund müssen immer beide Überlegenheiten überprüft werden. Des Weiteren werden mit der $\mathfrak{C}$-Metrik beide Eigenschaften der Konvergenz und Diversität in die Betrachtung der Überlegenheit einbezogen. Dennoch können keine qualitative Aussagen über die Konvergenz oder Diversität getroffen werden, weil jede Lösung der betrachteten PF einen gleichgroßen Anteil zur Bewertung beiträgt, ungeachtet der Nähe zur Referenzmenge oder allgemein der Lage im Lösungsraum [HJ98].

Gemäß Deb [Deb01] kann eine Unterscheidung von Metriken in drei Kategorien vorgenommen werden: Metriken der Konvergenz, Metriken der Diversität und Metriken, die beide Eigenschaften gleichzeitig abbilden. Die vorgeschlagenen Kategorien spiegeln die drei übergeordneten Ziele der MOO wider [ZDT00]: Minimierung der Distanz zur wahren PF, geeignete Verteilung der Paretooptimalen Lösungen entlang der PF und Maximierung der Ausdehnung der PF. Zusätzlich wird in

[JOZF14] eine vierte Kategorie vorgeschlagen: Metriken der Kapazität.

- **Metriken der Konvergenz:**
  Mit den Metriken der Konvergenz wird die Annäherung der ermittelten PF zu der wahren PF beziehungsweise einer approximierten Referenzmenge von Pareto-optimalen Lösungen oder einem Referenzpunkt bewertet. Für eine Betrachtung von nicht Pareto-optimalen Lösungen in einer approximierten Referenzmenge ist zu beachten, dass die Möglichkeit einer falschen Bewertung resultiert. Des Weiteren nimmt die Verteilung von Lösungen der Referenzmenge im Lösungsraum einen Einfluss auf die Bewertung [HJ98]. Mit einer Minimierung der Distanz geht in der Regel eine Steigerung der Güte einher.

  In [ZT98] wird im Rahmen der MOO die unäre $\mathfrak{S}$-Metrik eingeführt. Hierbei entspricht die $\mathfrak{S}$-Metrik dem Lebesgue-Maß ι, das einem geometrischen Objekt eine Größe zuordnet. Entsprechend wird einer PF das dominierte Hypervolumen bezüglich eines Referenzpunktes zugeordnet, dessen Größe wiederum den Wert der Metrik widerspiegelt. Für das bikriterielle MOOP entspricht das Hypervolumen der Vereinigung aller aufgespannten Flächeninhalte definiert über einen Referenzpunkt $\mathbf{r}$ und den Pareto-optimalen Lösungen. Resultierend definiert sich das dominierte Hypervolumen beziehungsweise $\mathfrak{S}$-Metrik einer Paretofront PF_A zu [Beu11]:

$$\begin{aligned}\mathfrak{S}(PF_A, \mathbf{r}) &= \iota(\{\mathbf{u} \in \mathbb{R}^k | \exists f(\mathbf{x}) \in PF_A : f(\mathbf{x}) \preceq \mathbf{u} \preceq \mathbf{r}\}) \\ &= \iota(\bigcup_{i=1}^{|PF_A|} [f(\mathbf{x}_i), \mathbf{r}])\end{aligned} \tag{6.2}$$

  Es gilt für $\mathbb{R}^k$, dass $k \geq 2$ ist. Das k-dimensionale Intervall $[f(\mathbf{x}_i), \mathbf{r}]$ beschreibt das Hypervolumen, welches der dominierte Raum von $\mathbf{x}_i$ ist und durch $\mathbf{r}$ begrenzt wird. Dieser dominierte Raum von $\mathbf{x}_i$ wird auch als $\mathfrak{S}$-Metrikbeitrag bezeichnet. Die $\mathfrak{S}$-Metrik ist eine monotone Funktion, da eine Hinzunahme einer Pareto-optimalen Lösung in keinem Fall zu einer Verschlechterung führt. Des Weiteren ist das Hypervolumen der approximierten PF durch das der wahren PF beschränkt. Entsprechend existiert ein Maximalwert der $\mathfrak{S}$-Metrik. Ferner nimmt die $\mathfrak{S}$-Metrik eine Differenzierung der Überlegenheitsrelationen vor, insofern der Referenzpunkt derart gewählt wird, dass jede betrachtete PF einen positiven $\mathfrak{S}$-Metrikwert aufweist. Entsprechend wird jeder überlegenen PF einen besseren $\mathfrak{S}$-Metrikwert zugeordnet. Aus diesen Gründen gilt die $\mathfrak{S}$-Metrik als eine der fairsten Metrik, um die Approximation an die wahre PF zu bewerten [ZTL+03, Beu06]. Jedoch ist zu beachten, dass die Ordnung der PFs stark abhängig von der Lage des Referenzpunktes ist [KC02, YLW+07].

  Erweiterungen der $\mathfrak{S}$-Metrik werden in [ZBT07, ZTB10] angegeben. Hierbei wird eine Verallgemeinerung des Referenzpunkt durch eine Referenzmenge eingeführt, welche beispielsweise die wahre PF des zugehörigen MOOP darstellen kann. Des Weiteren wird ein gewichtetes Hypervolumen eingeführt, um bestimmte Regionen des Zielfunktionraumes gemäß der Präferenzen des DM zu gewichten.

  Eine weitere Konvergenzmetrik die $\mathfrak{K}$-Metrik wird in beschrieben [DJ02]. Ähnlich der Erweiterung der $\mathfrak{S}$-Metrik in [ZTB10] wird die Distanz der approximierten PF zu einer Referenzmenge beziehungsweise wahren PF bestimmt. Für jede Lösung $f(\mathbf{x}) \in PF_A$ wird mittels einer

Distanzmetrik, hier die kleinste normalisierte euklidische Distanz $d(f(\mathbf{x}))$, die Entfernung zu der wahren oder Referenz-Paretofront PF_R wie folgt bestimmt:

$$d(f(\mathbf{x}_i)) \min_{j=1}^{|PF_R|} \sqrt{\sum_{k=1}^{\#\text{Dimensionen}} \left(\frac{f_k(\mathbf{x}_i) - f_k(\mathbf{x}_j)}{f_k^{\max} - f_k^{\min}} \right)^2} \tag{6.3}$$

Mit den Parametern $f_k^{\min}$ und $f_k^{\max}$ werden die minimalen beziehungsweise maximalen Ausprägungen der k-ten Zielfunktion in PF_R angegeben. Die Konvergenzmetrik $\mathfrak{K}(PF_A)$ wird für eine Paretofront PF_A mittels der Mittelwerte über alle $d(f(\mathbf{x}_i))$ der nicht dominierten Lösungen G in PF_A gebildet:

$$\mathfrak{K}(PF_A) = \frac{\sum_{i=1}^{|G|} d(f(\mathbf{x}_i))}{|PF_A|} \tag{6.4}$$

Zusammenfassend hat die Wahl und allgemeiner die Lage der Referenzmenge oder des Referenzpunktes einen Einfluss auf die Bewertung der betrachteten PFs. Insofern die wahre PF unbekannt ist, kann eine Mehrfachbewertung mit unterschiedlichen Referenzmengen und -punkten sinnvoll sein. Neben der Konvergenz und Annäherung an eine Referenzmenge, kann auch die Maximierung der Distanz von bekannten, schlechten Lösungen betrachtet werden [SVVW80].

- **Metriken der Diversität:**
  Eine geeignete Verteilung der Pareto-optimalen Lösungen entlang der PF spiegelt eine gute Abdeckung des Lösungsraumes wider. Oftmals wird eine gleichmäßige Verteilung angestrebt, welche problemspezifisch abweichen kann. Des Weiteren ist eine maximale Ausdehnung der PF erwünscht, um einen möglichst großen Bereich mit Pareto-optimalen Lösungen im Lösungsraumes abzudecken. Beide Eigenschaften zusammen führen zu einer PF mit hoher Diversität. Die Bewertung der Diversität basiert auf einem definierten Distanzmaß.

  Die $\mathfrak{S}$-Metrikbeiträge [ZT98, BNR06] der Pareto-optimalen Lösungen entlang einer Paretofront PF_A spiegeln die Verteilung der Lösungen im Zielfunktionsraum wider. Entsprechend ist der $\mathfrak{S}$-Metrikbeitrag für eine Pareto-optimale Lösung $f(\mathbf{x}) \in PF_A$ durch

$$\mathfrak{S}_{\text{Beitrag}}(f(\mathbf{x}), PF_A, \mathbf{r}) = \mathfrak{S}(PF_A, \mathbf{r}) - \mathfrak{S}(PF_A \backslash \{f(\mathbf{x})\}, \mathbf{r}) \tag{6.5}$$

  definiert. Entsprechend wird das zugehörige Hypervolumen einer Lösung durch die k-Hyperebenen der direkten Nachbarlösungen definiert. Die Lage des Referenzpunktes führt zu einer Maximierung (Minimierung) der $\mathfrak{S}$-Metrik, um eine höhere Güte und somit Annäherung an die wahre PF zu erreichen. Dabei wirkt sich die Anzahl von Lösungen in PF_A grundsätzlich positiv auf die Güte aus. Des Weiteren kann über die Standardabweichung der $\mathfrak{S}$-Metrikbeiträge ein Rückschluss auf die Verteilung der Lösungen entlang der PF gezogen werden. Eine ungleichmäßige Verteilung führt zu einer höheren Standardabweichung.

  Die in [Sch95] beschriebene Schott-Metrik misst, wie gleichmäßig die Pareto-optimalen Lösungen im Lösungsraum verteilt sind. Es wird vorgeschlagen, die Metrik in Verbindung mit Konvergenzmetriken einzusetzen. Die Metrik kann auf einen höherdimensionalen Lösungs-

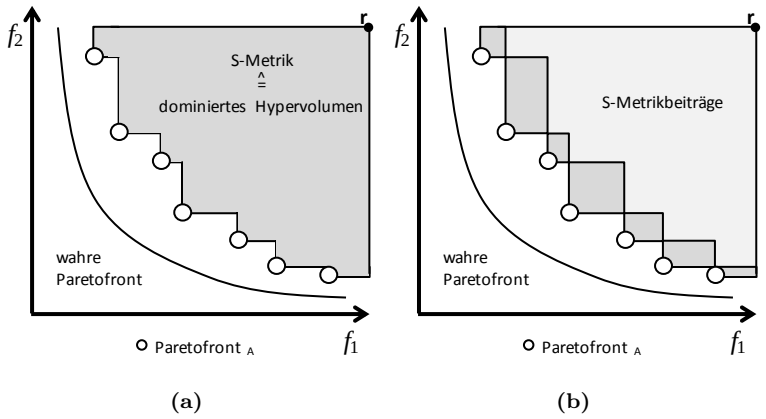


Abbildung A.2.: Konvergenz- und Diversitätsmetrik. Unter (a): $\mathfrak{S}$-Metrik. Unter (b): $\mathfrak{S}$-Metrikbeiträge (angelehnt an [BNR06], Seite 3).

raum erweitert werden. Schott definiert die Metrik

$$SS(f(\mathbf{x}), PF_A) = \sqrt{\frac{1}{n-1} \sum_{(i+1)} n(\overline{d} - d_i)^2} \tag{6.6}$$

mit $d_i = \min_j(|f_{k=1}(\mathbf{x}_i) - f_{k=1}(\mathbf{x}_j)| + |f_{k=2}(\mathbf{x}_i) - f_{k=2}(\mathbf{x}_j)|)$. Somit beschreibt d_i für jede Pareto-optimale Lösung die geringste Distanz über alle anderen Lösungen in PF_A. Hingegen beschreibt $\overline{d}$ den Mittelwert über alle d_i. Ferner gilt $i, j = 1 \ldots n$ mit $n = |PF_A|$. Nimmt die Metrik einen Wert von null an, dann sind alle Lösungen abstandsgleich zu einander im Lösungsraum verteilt.

In [Zit99] wird eine Diversitätsmetrik M_2 angegeben, die versucht eine Messung der Verteilung Pareto-optimaler Lösungen im Lösungsraum vorzunehmen. Hierfür wird eine ϵ-Umgebung eingeführt, die die Nachbarschaft von Lösungen definiert. Auf der Grundlage der ϵ-Umgebung wird überprüft, wieviele Lösungen jeweils in den Nachbarschaften der Lösungen liegen. Die M_2-Metrik ist ebenfalls abhängig von der Distanzmetrik $\| \bullet \|$, welche auf den Lösungen definiert ist. Daraus folgt, dass die M_2-Metrik für eine Paretofront PF_A einen konkreten Wert im Intervall $[0, |PF_A|]$ annimmt.

$$M_2(PF_A) = \frac{1}{|PF_A| - 1} \sum_{f(\mathbf{x}_i) \in PF_A} |\{f(\mathbf{x}_j) \in PF_A | \, \|f(\mathbf{x}_i) - f(\mathbf{x}_j)\| > \delta\}| \tag{6.7}$$

Demnach nimmt die Metrik größere Werte für PFs eines Optimierungsproblems mit höherer Diversität an. Ebenfalls wird in [Zit99] eine Metrik M_3 angegeben, die die maximale Ausdehnung jeder Dimension einer PF bewertet. Die M_3 Metrik wird für PF_A und eine Distanzmetrik $\| \bullet \|$ durch

$$M_3(PF_A) = \sqrt{\sum_{k=1}^{\#\text{Dimensionen}} \max\{\|f_k(\mathbf{x}_i) - f_k(\mathbf{x}_j)\| \, | f(\mathbf{x}_i), f(\mathbf{x}_j) \in PF_A\}} \tag{6.8}$$

beschrieben.

- **Metriken der Konvergenz und Diversität:**
  Mit Hilfe dieser Klasse von Metriken können die Eigenschaften der Konvergenz und Diversität gleichzeitig bewertet werden, um die Güte einer PF anzugeben. In [Sch95] wird die MPFE[1]-Metrik

$$MPFE(PF_A, PF_R) = \max_{i=1}^{|PF_A|} \sqrt{\min_{j=1}^{|PF_R|} \sum_{k=1}^{\#\text{Dimensionen}} |f_k(x_j) - f_k(x_i)|^2} \tag{6.9}$$

  beschrieben. Hierbei wird die größte Distanz über alle Lösungen in PF_A zu der jeweils am nächstgelegenen Lösung in PF_R ermittelt. Die MPFE-Metrik tendiert eher dazu, Extremlösungen in die Bewertung einzubeziehen, anstatt die gesamte PF zu untersuchen [JOZF14].

  Des Weiteren kann auch die $\mathfrak{S}$-Metrik dieser Klasse zugeordnet werden. Zur Bestimmung der $\mathfrak{S}$-Metrik und resultierend des Hypervolumens muss die Summe über alle $\mathfrak{S}$-Metrikbeiträge ermittelt werden. Hierbei nimmt zum einen das Hypervolumen zu, umso mehr sich die Paretofront PF_A der wahren PF annähert. Zum anderen lässt ein größeres Hypervolumen vermuten, dass die Lösungen in PF_A räumlich gleichmäßiger entlang der PF verteilt sind [JOZF14].

- **Metriken der Kapazität:**
  Die Metriken der Kapazität messen die Anzahl an Pareto-optimalen Lösungen innerhalb der PF, beziehungsweise das Verhältnis Pareto-optimaler Lösungen einer Referenz-PF. Grundsätzlich wird das Ziel verfolgt, möglichst viele Pareto-optimale Lösungen entlang der PF zu ermitteln. Vor allem in Nutzung mit Metriken der Konvergenz und Diversität kann eine umfassende Bewertung der PFs vorgenommen werden.

  In [VL00] führen Veldhuizen et al. unter anderem die ONVG[2]-Metrik

$$\text{ONVG}(PF_A) = |PF_A| \tag{6.10}$$

  ein. Mit dieser sehr intuitiven Metrik wird die Anzahl $|\bullet|$ Pareto-optimaler Lösungen angegeben. Die Metrik ist unabhängig und ermittelt für eine Menge von PFs eine vollständige Ordnung. Jedoch kann die Metrik ohne Betrachtung weiterer Eigenschaften zu missverständlichen Ergebnissen hinsichtlich der Qualität einer PF führen. Ein Beispiel stellt sich derart dar, dass $\text{ONVG}(PF_A) = 1$ und diese eine Pareto-optimale Lösung alle Lösungen in PF_B dominiert und somit „besser“ ist, obwohl $\text{ONVG}(PF_A) < \text{ONVG}(PF_B)$ gilt [KC02]. Eine erweiterte ONVGR[3]-Metrik misst das Verhältnis von Pareto-optimalen Lösungen zu einer Referenz-PF.

$$\text{ONVGR}(PF_A, PF_R) = \frac{|PF_A|}{|PF_R|} \tag{6.11}$$

  Mit der ER[4]-Metrik führt Veldhuizen in [Vel99] eine Metrik ein, die eine Bewertung bezüglich der Ähnlichkeit von zwei PFs im eingeschränkten Sinne zulässt. Hierbei wird überprüft, welche Pareto-optimalen Lösungen in beiden PFs enthalten sind $|PF_A \cap PF_R|$ und in das

[1] Engl. Maximum Pareto Front Error
[2] Engl. Overall Non-dominated Vector Generation
[3] Engl. Overall Non-dominated Vector Generation Ratio
[4] Engl. Error Ratio

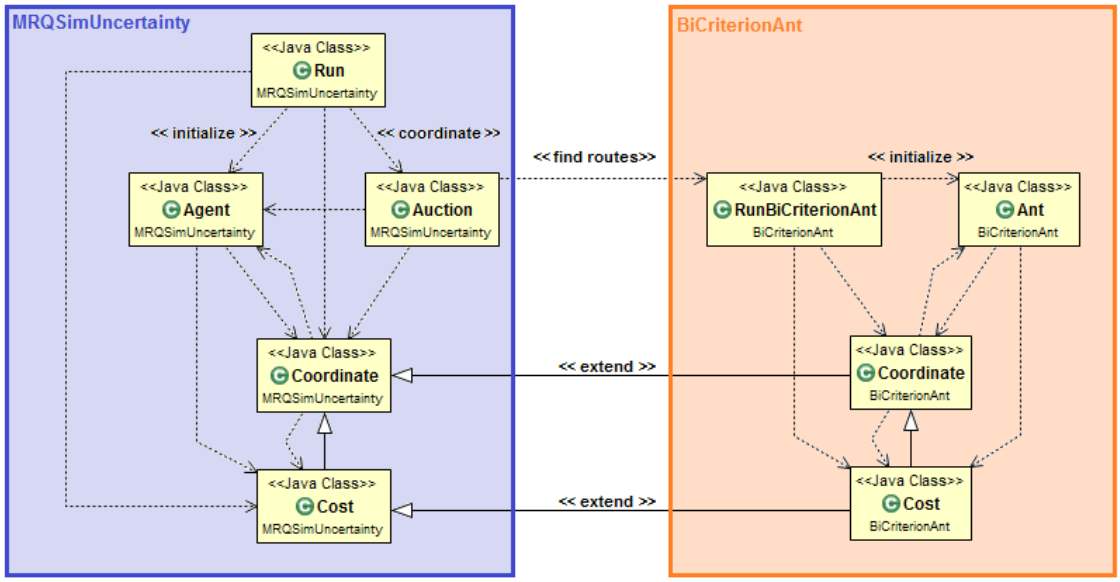


Abbildung B.3.: Implementierung des SMO-TCM. Beziehungen zwischen den Klassen der Module *MRQ-SimUncertainty* und *BiCriterionAnt*.

Verhältnis der Anzahl der Pareto-optimalen Lösungen der Referenz-PF gesetzt.

$$\mathrm{ER}(PF_A, PF_R) = 1 - \frac{|PF_A \cap PF_R|}{|PF_R|} \tag{6.12}$$

Grundsätzlich kann die $\mathfrak{C}$-Metrik ebenfalls dieser Klasse zugeordnet werden und misst im paarweisen Vergleich die Anzahl der dominierenden Lösungen einer Paretofront PF_A gegenüber einer weiteren Paretofront PF_B. Demnach ist die $\mathfrak{C}$-Metrik unabhängig von einer Referenz-PF [JOZF14].

B. Programmiertechnische Umsetzung

Die Implementierung der Algorithmen ist in Java erfolgt und gliedert sich in drei Module auf. Zusätzlich ist eine *R*-Schnittstelle integriert. Mit Hilfe von *R* wird das *IRace* [LIS10, LIDLSB11] durchgeführt, um ein optimiertes, robustes Heuristik-Parameterset für die BiCriterionAnt Heuristik ermitteln zu können. Zusätzlich ist das Framework *JExcelAPI* integriert und bietet somit die Möglichkeit vordefinierte Experimente durchführen zu lassen und die Ergebnisse der Experimente in einer Excel-Datei zu speichern. Eine Gesamtübersicht der Module sowie Beziehungen und der Aufruf unter den Modulklassen ist in der Abbildung B.3 veranschaulicht.

Das Modul *BiCriterionAnt* enthält die Umsetzung des in Kapitel 4.3 beschriebenen Algorithmus bezüglich des MOSPP. Der Programmaufbau ist in die nachfolgenden sechs Klassen unterteilt und in der Abbildung B.4 veranschaulicht:

- **Coordinate:**
  Die Klasse *Coordinate* beschreibt eine Zelle in der Systemumgebung. Hierbei wird die horizontale und vertikale Lage einer Zelle mit Hilfe der zugehörigen *(x,y)*-Koordinate beschrieben. Darauf aufbauend kann die direkte Nachbarschaft einer konkreten Zelle ermittelt werden.

Metrik	Klasse	Komplexität	Bemerkungen
$\mathfrak{C}$-Metrik [ZT98, ZDT00]	Überlegenheit (Kapazität)	$O(Dim\|PF_A\| * \|PF_B\|)$	- Ausschließlich paarweiser Vergleich von zwei PFs - Keine Bewertung hinsichtlich dem Grad/Ausmaß der Überlegenheit einer PF - Keine Bewertung hinsichtlich der Konvergenz und Diversität einer PF - Intuitiver Ansatz
$\mathfrak{S}$-Metrik [ZT98]	Konvergenz	$O(\|PF_A\|^{Dim-1})$	- Vergleich mehrerer PFs - $\mathfrak{S}$-Metrik ist monoton - Bewertung anhand eines für die untersuchten PFs allgemeingültigen Referenzpunktes - Ordnung der PFs ist abhängig von der Lage des Referenzpunktes
$\mathfrak{K}$-Metrik [DJ02]	Konvergenz	$O(Dim\|PF_A\| * \|PF_R\|)$	- Vergleich mehrerer PFs - $\mathfrak{K}$-Metrik ist nicht monoton - Bewertung anhand einer Referenz-PF - Bewertung ist stark abhängig von der Referenz-PF - Mehrfachbewertungen können nötig sein
$\mathfrak{S}$-Metrikbeitrag [ZT98]	Diversität	$O(\|PF_A\|^{Dim-1})$	- Vergleich mehrerer PFs - Basiert auf der $\mathfrak{S}$-Metrik - Bewertung des Anteils einer Lösung der PF am Hyperraum - Detailanalyse entlang der PF möglich
SS-Metrik [Sch95]	Diversität	$O(Dim\|PF_A\|^2)$	- Vergleich mehrerer PFs - SS-Metrik ist nicht monoton - Bewertung der Abstände der Lösungen zueinander innerhalb einer PF - Mehrwert durch ergänzende Nutzung mit anderen Metriken
M_2-Metrik [Zit99]	Diversität	$O(Dim\|PF_A\|^2)$	- Vergleich mehrerer PFs - Bewertung bezüglich der Anzahl der Lösungen die innerhalb einer ϵ-Umgebung liegen - Bewertung abhängig von der Definition der ϵ-Umgebung und der Distanzmetrik
M_3-Metrik [Zit99]	Diversität	$O(Dim\|PF_A\|^2)$	- Vergleich mehrerer PFs - Bewertung der maximalen Ausdehnung der PF entlang jeder Dimension - Ausreißerlösungen können zu Fehlbewertungen führen - Mehrwert durch ergänzende Nutzung mit der M_2-Metrik
MPFE-Metrik [Zit99]	Diversität	$O(Dim\|PF_A\| * \|PF_R\|)$	- Vergleich mehrerer PFs - Bewertung anhand einer Referenz-PF - Tendiert dazu sich auf Extremlösungen zu konzentrieren
ONVG-Metrik [VL00]	Kapazität	gering	- Vergleich mehrerer PFs - Bewertung der Anzahl Pareto-optimaler Lösungen innerhalb der PF - Bedingte Aussage über die Güte einer PF - Mehrwert durch ergänzende Nutzung mit anderen Metriken

Tabelle 6.1.: Zusammenfassung der Metriken für die Bewertung einer Pareto-Front (erweitert nach [Pre11]).

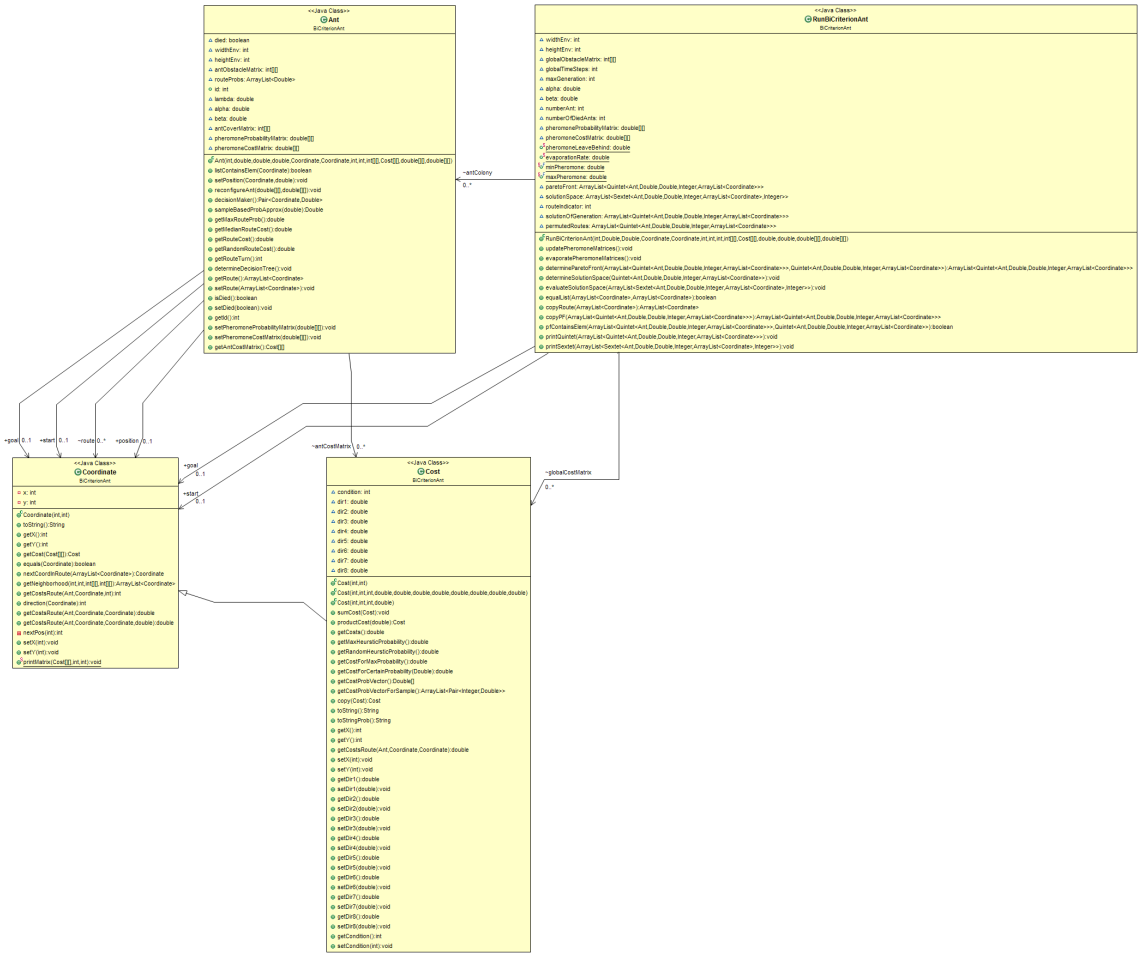


Abbildung B.4.: Implementierung des *BiCriterionAnt* Moduls und Veranschaulichung der Beziehungen zwischen den Klassen.

Zusätzlich enthält ein Objekt der Klasse *Coordinate* ein Verweis auf ein Objekt der Klasse *Cost*, um einen Vektor zur Beschreibung der Eintrittswahrscheinlichkeiten bezüglich der acht möglichen Strömungsrichtungen vorzunehmen. Daraus resultierend kann zum einen die Position einer Ameise in der Systemumgebung mittels eines Objektes der Klasse *Coordinate* beschrieben werden. Zum anderen kann eine Route durch eine Folge von Objekten der Klasse *Coordinate* beschrieben werden. Hierbei kann die nächste zu befahrende Zelle einer Route ermittelt werden. Die (geschätzten) Gesamtkosten einer Route können in Verbindung mit der Klasse *Cost* bestimmt werden.

- **Cost:**
  Mit der Klasse *Cost* wird die Beschreibung der Eintrittswahrscheinlichkeiten für die acht möglichen Strömungsrichtungen in einem Vektor beschrieben. Hierbei sind elementare Rechenoperationen auf Vektoren implementiert. Zusätzlich können die (geschätzten) Kosten ermittelt werden, die für die Befahrung einer Zelle resultieren. Diese Informationen wird von der Klasse *Ant* genutzt, um eine Routenplanung durchzuführen. Die (geschätzten) Kosten sowie die zugehörige Eintrittswahrscheinlichkeit der Kosten für eine Route werden insbesondere bei der Bestimmung eines Auktionsgebotes im Rahmen einer optimierten Koordination von Agenten im TCM benötigt.

- **Ant:**
  Eine Ameise wird durch allgemeine Attribute wie Position, Start- und Zielzelle, Hindernismatrix, Kostenmatrix und der befahrenen Route beschrieben. Ziel einer Ameise besteht in der Bestimmung einer Route zwischen der Start- und Zielzelle. Die Hindernismatrix stellt eine Binärmatrix dar und beschreibt, welche der Zellen in der Systemumgebung durch ein Hindernis blockiert oder frei sind. Die Kostenmatrix hingegen ist eine Zusammenfassung der Zellen zugehörigen Objekte der Klasse *Cost*. Somit dient die Kostenmatrix zur individuellen Berechnung der approximierten Fahrtkosten und Angabe der Eintrittswahrscheinlichkeiten zwischen zwei Zellen. Des Weiteren werden Attribute zugewiesen, die für den BiCriterionAnt Algorithmus benötigt werden. Jede Ameise verfügt über eine eindeutige Kennung und den zugehörigen Gewichtungsfaktor λ, um eine dislozierte Suche im Lösungsraum zu ermöglichen. Die Gewichtungsfaktoren α sowie β tragen maßgeblich zur Bewertung beziehungsweise Auswahl einer Lösungskomponente bei. Hierbei werden die Pheromon- und Heuristikwerte auf den Lösungskomponenten gewichtet. Dazu wird jeweils eine Pheromonmatrix bezüglich der Eintrittswahrscheinlichkeit und der Kosten verwaltet. Die heuristischen Informationen werden aus der Kostenmatrix ermittelt.

  Die *Ant* Klasse ermöglicht das Setzen von Ameisen auf eine neue Zelle sowie einen DM, um eine Zelle aus der Nachbarschaft gemäß den ermittelten Auswahlwahrscheinlichkeiten bestimmen zu können. Des Weiteren kann nach jeder Generation die Pheromonmatrix und allgemeine Eigenschaften der Ameise, wie das Zurücksetzen auf die Startzelle, konfiguriert werden. Zur Bestimmung der Pareto-optimalen Lösungen stehen die Funktionalitäten zur Bestimmung der Kosten, Eintrittswahrscheinlichkeit und Anzahl an Richtungsänderungen einer Route zur Verfügung. Hierbei wird sich auf die bereitgestellten Funktionalitäten der Klassen *Cost* und *Coordinate* abgestützt.

- **RunBiCriterionAnt:**
  Mit Hilfe der Klasse *RunBiCriterionAnt* wird der Ablauf einer Routenoptimierung gemäß

dem BiCriterionAnt Algorithmus gesteuert. Hierzu werden allgemeine Attribute, wie Start- und Zielzelle, Größe der Systemumgebung, Hindernisse und Strömungsinformationen, vorgegeben. Des Weiteren werden spezifische Attribute für den BiCriterionAnt Algorithmus definiert. Für die Heuristik werden die maximale Anzahl von Generationen, die Gewichtungsparameter α und β, die Anzahl von Ameisen, die Definition einer Kolonie, die Evaporationsrate sowie die obere und untere Grenze der Pheromonmenge auf einer Lösungskomponente definiert. Des Weiteren werden die Pheromonmatrizen, PF und Gesamtlösungsmenge initialisiert.

Auf der Grundlage der Initialisierung wird eine Optimierung des MOSPP durchgeführt. Hierbei werden die Klassen *Coordinate*, *Cost* und *Ant* eingebunden. Nach jeder Generation werden die gesamte Lösungsmenge und die aktuelle PF ermittelt. Auf dieser Grundlage wird eine Aktualisierung der Pheromonmatrizen, einschließlich einer Pheromonevaporation, durchgeführt und an die Ameisen übergeben.

- **Experiment:**
  Die Klasse *Experiment* ermöglicht die Durchführung von einem Experiment mit mehreren Durchläufen und verschiedenen Instanzen. Für jeden Aufruf der Klasse *RunBiCriterionAnt* werden die Ergebnisse der implementierten Metriken ermitteln und in einer Excel-Datei gespeichert. Die Verknüpfung mit Excel wird über die *JExcelAPI* realisiert.

- **RExperiment:**
  Die Klasse *RExperiment* ist eine Erweiterung der Klasse *Experiment*, um einen externen Aufruf und Instanziierung von Experimenten aus *R* heraus zu ermöglichen. Die Schnittstelle bietet somit die Möglichkeit die *R*-Implementierung von *IRace* [LIDLSB11] für die Durchführung einer Optimierung des Heuristik-Parametersets durchzuführen. Es werden die Ergebnisse zu einem Experiment an *IRace* zurückgegeben und verarbeitet.

Das Modul *MRQSimUncertainty* ist ähnlich dem Modul *BiCriterionAnt* aufgebaut und setzt das in Kapitel 5 beschriebene TCM zur vollständigen Abdeckung einer Systemumgebung mittels autonomer Agenten um. Hierbei ist eine Systemumgebung entsprechend den Implementierungen *Coordinate* und *Cost* des Moduls *BiCriterionAnt* umgesetzt. Zur Koordination und Steuerung der Agenten wird zusätzlich eine Klasse *Auction* eingeführt. In der Abbildung B.5 werden die Abhängigkeiten zwischen den Klassen des Moduls *MRQSimUncertainty* veranschaulicht.

- **Coordinate:**
  Die Klasse *Coordinate* definiert eine Zelle in der Systemumgebung. Mit der horizontalen und vertikalen Lage kann eine eindeutige Positionsbestimmung von Agenten in der Systemumgebung vorgenommen werden. Des Weiteren können Ziele in der Systemumgebung definiert werden, die durch Agenten abgedeckt werden sollen. Die Klasse *Coordinate* ermöglicht zu einer bestimmten Zelle die Nachbarschaft der acht angrenzenden Zellen zu ermitteln. Ferner wird dadurch das Scannen der Nachbarschaft durch einen Agenten ermöglicht. Hierbei werden Informationen über Hindernisse und die Strömung gesammelt. Auf der Grundlage der Strömungsinformationen, die mit der Klasse *Cost* definiert sind, können anschließend die (approximierten) Fahrtkosten zwischen zwei angrenzende Zellen bestimmt werden. Auf der Definition der Fahrtkosten können ebenfalls die (approximierten) Gesamtkosten mit zugehöriger Eintrittswahrscheinlichkeit für eine Route angegeben werden.

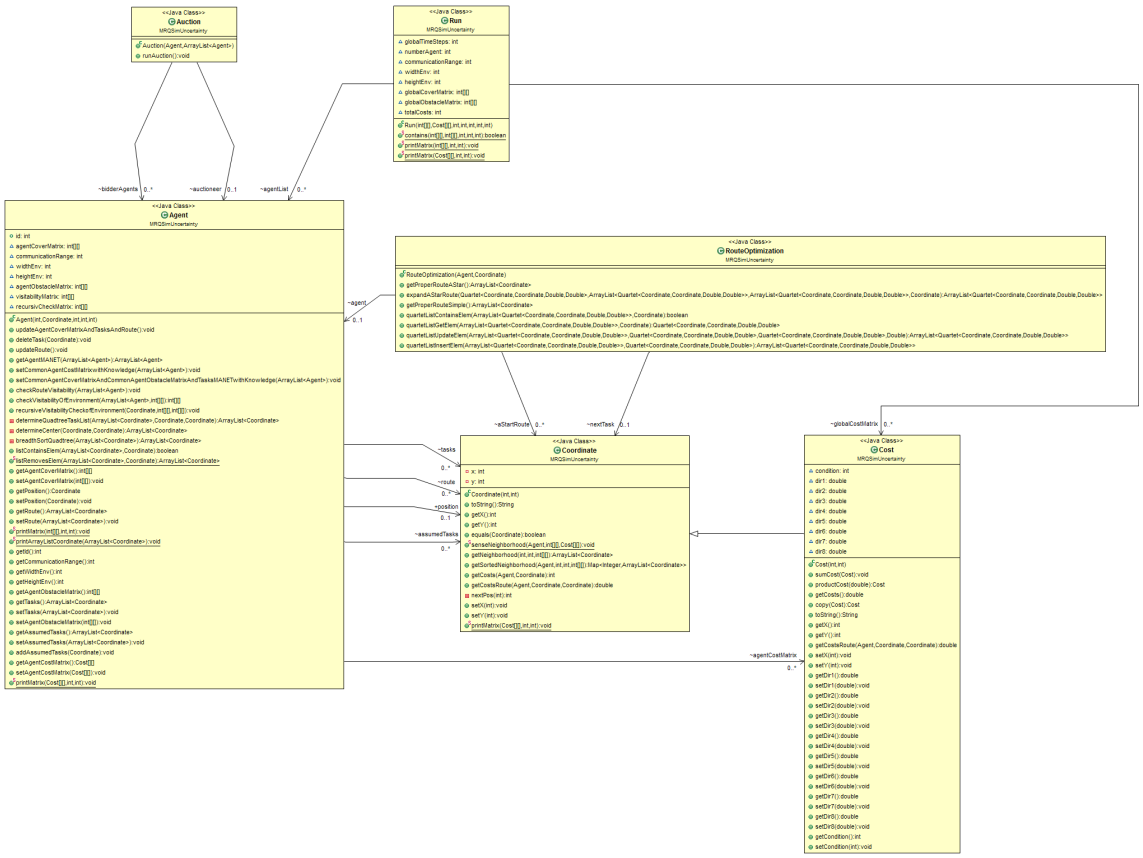


Abbildung B.5.: Implementierung des *MRQSimUncertainty* Moduls und Veranschaulichung der Beziehungen zwischen den Klassen.

- **Cost:**
  Die Klasse *Cost* stellt wie im Modul *BiCriterionAnt* die Beschreibung der Eintrittswahrscheinlichkeiten für die acht möglichen Strömungsrichtungen bereit. Ebenfalls werden elementare Rechenoperationen auf Vektoren implementiert. Zusätzlich können die (geschätzten) Kosten ermittelt werden, die für die Befahrung einer Zelle resultieren. Diese Informationen werden vor allem zur Bestimmung eines Auktionsgebotes, im Rahmen einer optimierten Koordination von Agenten im TCM, von der Klasse *Auction* benötigt.

- **Agent:**
  Ein Agent verfügt über eine eindeutige Kennung. Des Weiteren werden die Matrizen für die Abdeckung, Hindernisse und Strömungsrichtungen vorinitialisiert. Die bereits durchgeführten und noch zu erfüllenden Aufgaben werden jeweils getrennt verwaltet. Eine Aufgabe stellt hierbei den Besuch einer bestimmten Zelle dar. Die offenen Aufgaben werden in einer vordefinierten Reihenfolge in einem Stapelspeicher verwaltet. Für einen Agenten ist das übergeordnete Ziel einer vollständigen Abdeckung genau dann erfüllt, wenn der Stapelspeicher an offenen Aufgaben leer ist. Für eine vollständige Abdeckung kann ein Agent in jedem Zeitschritt auf der Grundlage der Kommunikationsreichweite das MANET bestimmen. Zum Informationsaustausch sind die nachfolgenden Funktionalitäten wie Aktualisierung der Abdeckungs-, Hindernis- und Strömungsrichtungsmatrix sowie der Abgleich der Aufgabenspeicher implementiert. Es besteht ebenfalls die Möglichkeit, eine Überprüfung der Befahrbarkeit aller Zellen durchzuführen. Zellen können aufgrund von Hindernissen eingeschlossen sein und somit außerhalb der befahrbaren Systemumgebung liegen.

- **Auction:**
  Mit der Klasse *Auction* ist der in Kapitel 5 beschriebene auktionsbasierte Algorithmus zur Koordination und Steuerung der autonomen Agenten implementiert. Die Klasse wird genau dann aufgerufen, wenn ein konkretes Objekt der Klasse *Agent* keine aktuelle Aufgabe aufweist. Dieser Agent tritt im Rahmen der Auktion als Auktionator auf und kann ebenfalls Gebote für die Übernahme einer Aufgabe abgeben. Eine Aktualisierung und Abgleich der verfügbaren Informationen in dem zugehörigen MANET ist zu diesem Zeitpunkt bereits erfolgt. Die Auktion wird ebenfalls für die Agenten des aktuellen MANET durchgeführt. Nach jeder Auktion wird die entsprechende Aufgabe an den Agenten mit dem besten Gebot zugewiesen. Es werden solange Auktionen durchgeführt bis der Auktionator eine Auktion gewonnen hat.

- **Run:**
  Die Klasse *Run* definiert den vollständigen Ablauf des in Kapitel 5 beschriebenen TCM. Hierzu wird eine vollständige Beschreibung der Systemumgebung vorgenommen. Neben der Festlegung der maximalen Ausdehnung, der Hindernisse und Strömungsinformationen wird auf dieser Grundlage eine Aufgabenreihenfolge bestimmt. Die Definition einer Systemumgebung wird über eine standardisierte Excel-Datei eingelesen. Hingegen basiert die Aufgabenreihenfolge auf einer Quadtree Zerlegung, die jeder Zelle genau eine Aufgabe zuweist. Des Weiteren werden die Anzahl der Agenten und deren allgemeine Eigenschaften Kommunikationsreichweite und Startzelle definiert sowie die Zuweisung der Aufgabenreihenfolge vorgenommen.

  Auf der Grundlage der Initialisierung wird eine optimierte Abdeckung, einer für die autonomen Agenten unbekannte Systemumgebung, ermittelt. Die Koordinierung und Optimierung basiert auf einem marktbasierten Ansatz. Hierbei werden die Klassen *Agent*, *Auction* und *RouteOptimization* eingebunden. Ebenso wird das Modul *BiCriterionAnt* zur Ermittlung einer optimierten Routen zur Erfüllung einer Aufgabe eingebunden. In jedem Zeitschritt wird eine Aktualisierung der individuellen Aufgaben und Hindernis- sowie Strömungsinformationen durchgeführt.

- **ExperimentMain:**
  Mit der Klasse *ExperimentMain* wird die Durchführung von einem Experiment mit mehreren Durchläufen und verschiedenen Instanzen organisiert. Die Ergebnisse entsprechend der implementierten Metriken werden für jeden Aufruf der Klasse *Run* ermittelt und geeignet in einer Excel-Datei dokumentiert. Die Verknüpfung mit Excel erfolgt auf dem *JExcelAPI* Framework.

C. IRace Ergebnisse

Nachfolgend ist das R-Logfile zu dem IRace Experiment beigefügt. Das Logfile umfasst die Initialisierung der Parameterbereiche entsprechend der Tabelle 4.1 aus Kapitel 4.3.3.2 sowie vier Iterationen. Es werden insgesamt 30 verschiedene Systemumgebungen überprüft. Die Ergebnisse der 1000 Einzelexperimente im Rahmen des gesamten IRace Experiments, bei denen systematisch Parameterinitialisierungen auf den Systemumgebungen überprüft worden sind, können der CD entnommen werden.

```
> #install.packages("irace")
> #install.packages("rJava")
>
> library(irace)
> library(rJava)
>
> .jinit()
> .jaddClassPath("/home/mpreuss/R/RExperimentMainBiCriterionNeu.jar")
> obj=.jnew("BiCriterionAnt/RExperimentMainBiCriterion")
>
> ## First we provide an implementation of the function to be optimized:
>
> biCritAnt<-function(maxGeneration, numberAnt, alpha, beta, pheromoneLeaveBehind, evaporationRate, instance)
+ {
+   result<- .jcall(obj,"D","runBiCritAnt", as.integer(maxGeneration),
+                                 as.integer(numberAnt),
+                                 as.double(alpha),
+                                 as.double(beta),
+                                 as.double(pheromoneLeaveBehind),
+                                 as.double(evaporationRate),
+                                 #.jarray(instance,dispatch=FALSE))
+                                 as.integer(instance,dispatch=FALSE))
+   return(result)
+ }
>
> .jcall(obj,"S","createAllInstances")
[1] "done"
>
> ## We generate 100 instances (in this case, weights):
> weights <-  floor(runif(n=100,min=0,max=101))
>
> ## On this set of instances, we are interested in optimizing four
> ## parameters of the BiCritAnt algorithm: alpha, beta, pheromonLeaveBehind (plb)
> ## and pheromonEvaporationRate (per). We setup the parameter space as follows:
>
> parameters.table <-
+ '
+ alpha "" r (1, 2)
+ beta "" r (1, 2)
+ plb "" i (35, 80)
+ per "" r (0.25, 0.6)
+ '
>
> ## We use the irace function readParameters to read this table:
> parameters <- readParameters(text = parameters.table)
>
> ## Next, we define the function  that will evaluate each candidate
> ## configuration on a single instance.
>
> ## maxGeneration, numberAnt, alpha, beta, pheromoneLeaveBehind, evaporationRate, instance
> hook.run <- function(experiment, config = list())
+ {
+   instance<-experiment$instance
+   candidate<-experiment$candidate
+
+   resultJava<-biCritAnt(
+     25,   #or    as.integer(candidate["generations"]),
+     30,   #or    as.integer(candidate["ants"]),
+     as.double(candidate["alpha"]),
+     as.double(candidate["beta"]),
+     as.double(candidate["plb"]),
+     as.double(candidate["per"]),
+     as.integer(instance))
+   return(resultJava)
+ }
>
```

```
> result <- irace(tunerConfig = list(
+   hookRun = hook.run,
+   instances = weights[1:100],
+   maxExperiments = 1000,
+   logFile = ""),
+   parameters = parameters)
>

# 2016-08-10 21:04:43 CEST: Initialization
# nbIterations: 4
# minNbSurvival: 4
# nbParameters: 4
# seed: 521637208
# confidence level: 0.95
# remainingBudget: 1000
# mu: 5

# 2016-08-10 21:04:43 CEST: Iteration 1 of 4
# experimentsUsedSoFar: 0
# timeUsedSoFar: 0
# timeEstimate: 0
# remainingBudget: 1000
# currentBudget: 250
# nbCandidates: 41

                    Markers:
                      x No test is performed.
                      - The test is performed and some candidates are discarded.
                      = The test is performed but no candidate is discarded.

+-+----------+---------+---------+---------+---------+---------+
| |     Task|    Alive|     Best| Mean best| Exp so far|Run Time|
+-+----------+---------+---------+---------+---------+---------+
|x|        1|       41|       13|  -0.5956|       41|01:07:57|
|x|        2|       41|       13|  -0.5639|       82|02:06:54|
|x|        3|       41|        1|  -0.5572|      123|01:37:49|
|x|        4|       41|       13|  -0.5645|      164|00:30:16|
|=|        5|       41|       13|  -0.5556|      205|00:28:10|
|=|        6|       41|       13|  -0.5505|      246|01:10:42|
+-+----------+---------+---------+---------+---------+---------+

Selected candidate:        13 mean value:    -0.5505

Description of the selected candidate:
  .ID. alpha  beta plb   per .PARENT.
13   13 1.4574 1.3547  38 0.4681       NA

# Elite candidates:
   alpha  beta plb   per
13 1.4574 1.3547  38 0.4681
1  1.9831 1.1887  80 0.4294
15 1.3705 1.0358  36 0.3563
37 1.0621 1.1058  35 0.3476

# 2016-08-11 04:06:34 CEST: Iteration 2 of 4
# experimentsUsedSoFar: 246
# timeUsedSoFar: 0
# timeEstimate: 0
# remainingBudget: 754
# currentBudget: 251
# nbCandidates: 35

                    Markers:
                      x No test is performed.
                      - The test is performed and some candidates are discarded.
                      = The test is performed but no candidate is discarded.
```

```
+-+----------+---------+---------+---------+---------+------+
| |     Task|    Alive|     Best| Mean best| Exp so far|Run Time|
+-+----------+---------+---------+---------+---------+------+
|x|        1|       35|       22|  -0.5711|       35|01:27:19|
|x|        2|       35|       14|  -0.5516|       70|00:53:07|
|x|        3|       35|        3|  -0.5507|      105|01:03:53|
|x|        4|       35|       31|    -0.54|      140|01:31:02|
|=|        5|       35|       22|  -0.5304|      175|01:15:10|
|=|        6|       35|       14|  -0.5419|      210|00:36:33|
|=|        7|       35|       14|  -0.5445|      245|00:26:58|
+-+----------+---------+---------+---------+---------+------+

Selected candidate:        14 mean value:   -0.5445

Description of the selected candidate:
  .ID. alpha  beta plb   per .PARENT.
51   51 1.6589 1.1964  36 0.4079       13

# Elite candidates:
   alpha  beta plb   per
51 1.6589 1.1964  36 0.4079
71 1.2308 1.2504  40 0.4063
68 1.3890 1.3701  36 0.3635
13 1.4574 1.3547  38 0.4681
```

 # 2016-08-11 11:20:42 CEST: Iteration 3 of 4

```
# experimentsUsedSoFar: 491
# timeUsedSoFar: 0
# timeEstimate: 0
# remainingBudget: 509
# currentBudget: 254
# nbCandidates: 31

                    Markers:
                      x No test is performed.
                      - The test is performed and some candidates are discarded.
                      = The test is performed but no candidate is discarded.

+-+----------+---------+---------+---------+---------+------+
| |     Task|    Alive|     Best| Mean best| Exp so far|Run Time|
+-+----------+---------+---------+---------+---------+------+
|x|        1|       31|       25|  -0.5813|       31|01:27:02|
|x|        2|       31|       13|  -0.5914|       62|00:31:45|
|x|        3|       31|       22|  -0.5705|       93|00:58:03|
|x|        4|       31|       22|  -0.5814|      124|00:51:50|
|=|        5|       31|       22|  -0.5789|      155|00:31:16|
|=|        6|       31|       30|  -0.5485|      186|00:46:59|
|=|        7|       31|       30|  -0.5392|      217|00:34:11|
|=|        8|       31|       31|  -0.5144|      248|00:46:24|
+-+----------+---------+---------+---------+---------+------+

Selected candidate:        31 mean value:   -0.5144

Description of the selected candidate:
  .ID. alpha  beta plb   per .PARENT.
99   99 1.687 1.1217  41 0.4161       51

# Elite candidates:
   alpha  beta plb   per
99 1.6870 1.1217  41 0.4161
85 1.4956 1.2746  37 0.3350
90 1.7271 1.2173  38 0.3792
80 1.3202 1.3121  43 0.3914
```

```
# 2016-08-11 17:48:15 CEST: Iteration 4 of 4
# experimentsUsedSoFar: 739
# timeUsedSoFar: 0
# timeEstimate: 0
# remainingBudget: 261
# currentBudget: 261
# nbCandidates: 29

                    Markers:
                      x No test is performed.
                      - The test is performed and some candidates are discarded.
                      = The test is performed but no candidate is discarded.

+-+----------+--------+--------+--------+--------+--------+
| |     Task|    Alive|     Best| Mean best| Exp so far|Run Time|
+-+----------+--------+--------+--------+--------+--------+
|x|        1|       29|       18|          -0.6012|       29|01:07:34|
|x|        2|       29|       17|          -0.6247|       58|00:38:38|
|x|        3|       29|       17|          -0.6016|       87|00:33:28|
|x|        4|       29|       17|          -0.5651|      116|00:40:13|
|=|        5|       29|       22|          -0.5547|      145|00:19:42|
|=|        6|       29|       17|          -0.5461|      174|00:35:00|
|=|        7|       29|       17|          -0.5567|      203|00:22:41|
|=|        8|       29|        6|          -0.5601|      232|00:42:18|
|=|        9|       29|        6|          -0.5601|      261|00:25:11|
+-+----------+--------+--------+--------+--------+--------+

Selected candidate:        6 mean value:    -0.5601

Description of the selected candidate:
   .ID.  alpha beta plb   per .PARENT.
101  101 1.5442 1.25  36 0.3361       85

# Elite candidates:
    alpha   beta plb   per
101 1.5542 1.2500  36 0.3361
112 1.5065 1.3086  40 0.3400
116 1.6560 1.1049  41 0.4155
110 1.5281 1.2346  38 0.3405
# 2016-08-11 23:13:05 CEST: Stopped because budget is exhausted
# Iteration: 5
# nbIterations: 4
# experimentsUsedSoFar: 1000
# timeUsedSoFar: 0
# timeEstimate: 0
# remainingBudget: 0
# currentBudget: 261
# number of elites: 4
# nbCandidates: 29
# mu: 5
>
> ## We can print the best configurations found by irace as follows:
> candidates.print(result)
    alpha   beta plb   per
101 1.5542 1.2500  36 0.3361
112 1.5065 1.3086  40 0.3400
116 1.6560 1.1049  41 0.4155
110 1.5281 1.2346  38 0.3405
```

Erklärung

Hiermit versichere ich, dass ich die vorliegende Dissertation selbstständig angefertigt habe. Die aus fremden Quellen und Hilfsmitteln direkt oder indirekt übernommenen Gedanken und Zitate sind als solche kenntlich gemacht. Es wurden keine anderen als die in der Dissertation angegebenen Quellen und Hilfsmittel benutzt. Diese Dissertation wurde weder einer anderen Prüfungsbehörde vorgelegt noch veröffentlicht. Eine Veröffentlichung vor Abschluss des Promotionsverfahrens werde ich nicht nicht vornehmen. Die Bestimmungen der Promotionsordnung sind mir bekannt. Die vorgelegte Dissertation ist von Professor Dr. Stefan Pickl betreut worden.

Neubiberg, den 8. November 2017