



Universität der Bundeswehr München

Fakultät für Elektrotechnik

Institut für Mathematik und Datenverarbeitung

**Numerische Lösung diskret-kontinuierlicher
Vektoroptimierungsprobleme**

von
Regina Sturm

Diese Seite ist absichtlich leer.

Universität der Bundeswehr München
Fakultät für Elektrotechnik
Institut für Mathematik und Datenverarbeitung

Numerische Lösung diskret-kontinuierlicher Vektoroptimierungsprobleme

von

Dipl.-Math. Regina Sturm

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

Doktor-Ingenieur

(Dr.-Ing.)

genehmigten Dissertation

Vorsitzender	Prof. Dr. Konrad Pilzweger
1. Berichterstatter	Prof. Dr. Dr. Stefan Schäffler
2. Berichterstatter	Prof. Dr. Claus Hillermeier

Tag der Promotion: 31.10.2001

Diese Seite ist absichtlich leer.

Inhaltsverzeichnis

Symbole	viii
Einleitung	xiii
1 Grundlagen der Vektoroptimierung	1
1.1 Die Alternativenmenge	1
1.2 Die Kriterienmenge	2
1.3 Das Vektoroptimierungsproblem	2
1.4 Optimalität in der Vektoroptimierung	3
1.4.1 Effizienz	5
1.4.1.1 Eigenschaften der Effizienzmenge	7
1.4.2 Schwache Effizienz	8
1.4.3 Wesentliche Effizienz	8
1.4.4 Eigentliche Effizienz	9
2 Methoden der Vektoroptimierung	11
2.1 Methoden mit a priori Präferenzäußerung	12
2.1.1 Nutzenfunktionsmethoden	12
2.1.2 Methode der beschränkten Ziele	14
2.1.3 Lexikographische Methode	15
2.1.4 Optimierung mit Zielvorgaben	16
2.2 Methoden mit fortschreitender Präferenzäußerung	18
2.2.1 Methoden mit expliziter Trade-Off Information	19
2.2.2 Methoden mit impliziter Trade-Off Information	20
2.3 Generierungstechniken / Methoden mit a posteriori Präferenzäußerung	21
2.3.1 Deterministisches Suchen	21
2.3.1.1 Wichtungsmethode (parametrische Methode)	21
2.3.1.2 Methode der ε -Schranken	23

2.3.1.3	Methode der Gleichungsnebenbedingungen	24
2.3.1.4	Hybrider Wichtungs- und Nebenbedingungsansatz	25
2.3.1.5	Methode der gewichteten L_p -Metrik	25
2.3.1.6	Hyperbel Methode	26
2.3.1.7	Normalen-Rand Intersektion	27
2.3.1.8	Methode der Effizienzmengenschätzung	30
2.3.1.9	Stückweise Lineare Approximation	33
2.3.1.10	Methode der Vergleichszahlen	35
2.3.2	Kontinuierliches stochastisches Suchen	37
2.3.2.1	Methode der sich anpassenden Suche	37
2.3.2.2	Stochastisches Suchen nach Timmel	38
2.3.2.3	Stochastisches Verfahren nach Schäffler	41
2.3.3	Diskretes stochastisches Suchen	45
2.3.3.1	VEGA	46
2.3.3.2	Evolutionäre Strategien	46
2.3.3.3	Wichtungsansatz	47
2.3.3.4	Pareto-basierte Fitnessbewertung	48
2.3.3.5	Genetischer Mehrzielalgorithmus	49
2.3.3.6	Genetischer Algorithmus mit Nichtdominanz-Sortierung	49
2.3.3.7	Niched Pareto GA	50
2.4	Zusammenfassung	50
3	Diskret-kontinuierliche Vektoroptimierung	53
3.1	Das diskret-kontinuierliche Vektoroptimierungsproblem	54
3.2	Suche entlang von Pfaden dominierender Alternativen	55
3.2.1	Präferenzfreie Bewertung von Alternativen	55
3.2.2	Suchstrategie "Sequenzielle Suche" (SS)	56
3.2.3	Zusammenspiel von diskreten und kontinuierlichen Suchschritten	58
3.2.4	Bearbeitung der diskreten Suchschritte	63
3.2.5	Markov Ketten	65
3.2.5.1	Definition der Markov Ketten	65
3.2.5.2	Höhere Übergangswahrscheinlichkeiten	66
3.2.5.3	Eigenschaften einer Markov Kette	67
3.2.5.4	Die stationäre Verteilung	69
3.2.5.5	Der Metropolis Algorithmus	70
3.2.6	Suchstrategie "Parallele Suche" (PS)	74

3.2.7	Reduzierung des Suchraumes	79
3.2.8	Wahrscheinlichkeitsaussagen über das Erreichen von Paretooptima . . .	82
3.2.8.1	Der einzelne Markov Prozeß	82
3.2.8.2	Der stochastische Superprozeß	83
3.2.9	Algorithmus DKMOP SS2	94
3.2.10	Algorithmus DKMOP PS	96
4	Anwendungen und Testergebnisse	99
4.1	Optimale Auslegung einer Vorwärmstrecke in einem fossil befeuerten Kraftwerk	99
4.1.1	Technische und thermodynamische Grundlagen	99
4.1.1.1	Der Wasser-/Dampfkreislauf	99
4.1.1.2	Das Prinzip der Speisewasservorwärmung	100
4.1.1.3	Auslegungsparameter der Vorwärmstrecke	102
4.1.2	Das Optimierungsproblem	102
4.1.2.1	Diskrete Variablen	102
4.1.2.2	Kontinuierliche Variablen	103
4.1.2.3	Darstellung der Variablen in einem Entscheidungsbaum . . .	105
4.1.2.4	Die Ziele	107
4.1.2.5	Die Nebenbedingungen	109
4.1.2.6	Formulierung des Optimierungsproblems	111
4.1.3	Schwierigkeiten bei der Anpassung des entwickelten Verfahrens	113
4.1.4	Eigenschaften der Markov Kette	118
4.1.4.1	Irreduzibilität der Markov Kette	119
4.1.4.2	Aperiodizität der Markov Kette	119
4.1.5	Testergebnisse	121
4.1.5.1	Suchstrategie PS im Test	122
4.1.5.2	Suchstrategie SS im Test	129
4.2	Testanwendungen	134
4.2.1	Testanwendung 1	135
4.2.1.1	Das Mehrzielproblem	135
4.2.1.2	Testergebnisse	135
4.2.2	Testanwendung 2	136
4.2.2.1	Das Mehrzielproblem	136
4.2.2.2	Testergebnisse	136
4.2.3	Testanwendung 3	137
4.2.3.1	Das Mehrzielproblem	137

4.2.3.2	Testergebnisse	137
4.2.4	Testanwendung 4	138
4.2.4.1	Das Mehrzielproblem	138
4.2.4.2	Testergebnisse	138
5	Resümee und Ausblicke	139
	Literaturverzeichnis	141
	Index	146

Symbole

Gebäuchliche Symbole

\mathbb{N}	$\{1, 2, 3, \dots\}$
\mathbb{N}_0	$\mathbb{N} \cup \{0\}$
\mathbb{Z}	$\{\dots, -2, -1, 0, 1, 2, \dots\}$
\mathbb{R}^+	Menge der positiven reellen Zahlen
\mathbb{R}	Menge der reellen Zahlen
$\mathbb{R}^n, n \in \mathbb{N}$	Menge der n -dimensionalen Vektoren reeller Zahlen
$\mathbb{R}_-, n \in \mathbb{N}$	Menge der n -dimensionalen Vektoren negativer reeller Zahlen
$\mathbb{R}_+, n \in \mathbb{N}$	Menge der n -dimensionalen Vektoren positiver reeller Zahlen
$\delta(M)$	Rand der Menge M ; für M Teilmenge einer Menge N wird hier $\delta(M)$ als die Menge aller Elemente von N , von denen jede Umgebung sowohl Elemente aus M als auch aus $N \setminus M$ enthält, verstanden
$\text{int}(M)$	das Innere der Menge M ; $\text{int}(M)$ wird hier als die Menge $M \setminus \delta(M)$ interpretiert
$ M $	Mächtigkeit der Menge M

Spezielle Symbole

$R := C \times D \subseteq \mathbb{R}^m, m \in \mathbb{N}$	Alternativenmenge, zulässiger Bereich des diskret-kontinuierlichen Vektoroptimierungsproblems
$C \subseteq \mathbb{R}^{m_1}, m_1 \in \mathbb{N}$	kompakte Menge, "kontinuierlicher" Teil von R
$D \subseteq \mathbb{R}^{m_2}, m_2 \in \mathbb{N}$	endliche Menge, "diskreter" Teil von R
$x_C \in C$	erste m_1 Komponenten von $x \in R$, kontinuierliche Komponenten
$x_D \in D$	letzte m_2 Komponenten von $x \in R$, diskrete Komponenten
$F : R \mapsto \mathbb{R}^n, x \mapsto F(x), n \in \mathbb{N}, n > 1$	zu optimierende diskret-kontinuierliche Mehrzielfunktion
$F(R) + (\mathbb{R}_+^n \cup \{0\})$	$\{F(x) + v \mid x \in R, v \in \mathbb{R}_+^n \cup \{0\}\}$
$f_i : R \mapsto \mathbb{R}, x \mapsto f_i(x),$ $i = 1, \dots, n$	skalare Zielfunktion, Komponente von F
$\nabla f_i, i = 1, \dots, n$	Gradient der Funktion f_i
$F^{xD} : C \mapsto \mathbb{R}^n, x_C \mapsto F(x_C, x_D)$	zu minimierende kontinuierliche Mehrzielfunktion des kontinuierlichen Subproblems, das aus Festlegen der diskreten Komponenten x_D und Variation der kontinuierlichen Komponenten x_C entsteht

$f_i^{xD} : C \mapsto \mathbb{R}, x_C \mapsto f_i(x_C, x_D),$ $i = 1, \dots, n$	skalare Zielfunktion, Komponente von F^{xD}
$L^{xD} \subseteq C$	Lösungsmenge der "Minimierung" von F^{xD} über C
\bar{L}^{xD}	numerisch erzeugte Approximation von L^{xD}
$E^{xD} \subseteq R$	Menge aller Konkatenationen der $y \in L^{xD}$ mit x_D , d.h. aller $\{(y, x_D) y \in L^{xD}\}$
\bar{E}^{xD}	numerisch erzeugte Approximation von E^{xD}
$R_D \subseteq R$	$R_D := \cup_{x_D \in D} E^{xD}$
\bar{R}_D	numerisch erzeugte Approximation von R_D bzw. eines Ausschnittes von R_D
$x \prec y$	x dominiert y
$x \succ y$	x wird von y dominiert, x ist inferior zu y
$x \asymp y$	x dominiert nicht y und y dominiert nicht x
S	Menge der Zustände einer Markov Kette und Suchraum des Metropolis Algorithmus, der von der Markov Kette modelliert wird, insbesondere $ S < \infty$
$\{X_t\}_{t \in \mathbb{N}_0}$	stochastischer Prozeß aus diskreten Zufallsvariablen mit Realisierungen in S ; Markov Kette, die insbesondere (zeit)homogen ist
$\{x_t\}_{t \in \mathbb{N}_0}$	Realisierung der Markov Kette, $x_t \in S \quad \forall t \in \mathbb{N}_0$
$P_{ij}, i, j \in S$	Übergangswahrscheinlichkeit der Markov Kette, $P_{ij} := P\{X_t = j X_{t-1} = i\} \quad \forall t \in \mathbb{N}, i, j \in S$
P_{trans}	Matrix der Übergangswahrscheinlichkeiten aller Punktepaare $i, j \in S$, $P_{trans} := (P_{ij})_{i, j \in S}$
P_i	Wahrscheinlichkeit, mit der $i \in S$ der Anfangszustand der Markov Kette ist, $P_i := P\{X_0 = i\}$
P_{init}	Vektor der Wahrscheinlichkeiten aller Anfangszustände, $P_{init} := (P_i)_{i \in S}$
$P_{ij}^n, n \in \mathbb{N}, n > 1, i, j \in S$	höhere Übergangswahrscheinlichkeit der Markov Kette, $P_{ij}^n := P\{X_{t+n} = j X_t = i\} \quad \forall t \in \mathbb{N}_0, i, j \in S$
P_{trans}^n	Matrix der höheren Übergangswahrscheinlichkeiten aller Punktepaare $i, j \in S$, $P_{trans}^n := (P_{ij}^n)_{i, j \in S}$
$P_i^n, n \in \mathbb{N}, n > 1, i \in S$	Wahrscheinlichkeit, Zustand $i \in S$ nach n Übergängen in der Markov Kette zu erreichen (höhere absolute Wahrscheinlichkeit), $P_i^n := \sum_{j \in S} P_j P_{ji}^n$
P_{abs}^n	Vektor aller höheren absoluten Wahrscheinlichkeiten, $P_{abs}^n := (P_i^n)_{i \in S}$
$f_{ij}^n, n \in \mathbb{N}_0, i, j \in S$	Wahrscheinlichkeit, $j \in S$ von $i \in S$ aus nach n Übergängen in der Markov Kette erstmalig zu erreichen
$f_{ij}, i, j \in S$	Wahrscheinlichkeit, $j \in S$ von $i \in S$ aus im Laufe der Experimentenfolge einer Markov Kette zu erreichen, $f_{ij} := \sum_{n=1}^{\infty} f_{ij}^n$
$\mu_i \leq \infty, i \in S$	mittlere Wiederkehrzeit eines rekurrenten Zustandes $i \in S$, $\mu_i := \sum_{n=1}^{\infty} n f_{ii}^n$

$q_i, i \in S$	Wahrscheinlichkeit, $i \in S$ unabhängig vom Anfangszustand der Markov Kette nach unendlich vielen Übergängen zu erreichen, $q_i := \lim_{n \rightarrow \infty} P_{ji}^n \quad \forall j \in S$ (falls existent)
q	stationäre Verteilung der Markov Kette, $q := (q_i)_{i \in S}$ (falls existent)
IMA	idealisierter Metropolis Algorithmus mit unendlich vielen Übergängen, modelliert durch $\{X_t\}_{t \in \mathbb{N}_0}$
NMA	numerische Version des IMA mit endlich vielen ($\bar{t} \in \mathbb{N}$) Übergängen, modelliert durch $\{X_t\}_{0 \leq t \leq \bar{t}}, \bar{t} \in \mathbb{N}_0$
$Z : S \mapsto \mathbb{R}, s \mapsto Z(s)$	Zielfunktion, auf deren Werten der durch $\{X_t\}_{t \in \mathbb{N}_0}$ bzw. $\{X_t\}_{0 \leq t \leq \bar{t}}$ modellierte Metropolis Algorithmus IMA bzw. NMA arbeitet, hier $S \subseteq \mathbb{R}_D \times \mathbb{R}_D$ und $Z : S \mapsto \{min, mid, max\} \subset \mathbb{R}, (x, y) \mapsto Z[(x, y)], min < mid < max$
$G_{ij}, i, j \in S$	im Vorhinein festgelegte Generierungswahrscheinlichkeit des Metropolis Algorithmus (IMA und NMA), Wahrscheinlichkeit, daß $j \in S$ von $i \in S$ aus als nächster Standpunktkandidat vorgeschlagen wird
\mathfrak{G}	Matrix der Generierungswahrscheinlichkeiten $\mathfrak{G} := (G_{ij})_{i, j \in S}$; \mathfrak{G} definiert die Nachbarschaftsstruktur des Metropolis Algorithmus (IMA und NMA)
$N_i \subseteq S, i \in S$ $c \in \mathbb{R}^+$	Nachbarschaft eines Punktes $i \in S, N_i := \{j \in S G_{ij} > 0\}$ Kontrollparameter des Metropolis Algorithmus (IMA und NMA), der die Akzeptanzwahrscheinlichkeiten für Verschlechterungen im Zielfunktionswert steuert
$A_{ij}(c), i, j \in S$	Wahrscheinlichkeit des Metropolis Algorithmus (IMA und NMA), vom Standpunkt $i \in S$ aus den (mit Wahrscheinlichkeit G_{ij} vorgeschlagenen) Punkt $j \in S$ bei gegebenem $c \in \mathbb{R}^+$ als neuen Standpunkt zu akzeptieren, $A_{ij}(c) := \exp\left(\frac{-(Z(j)-Z(i))^+}{c}\right)$
$\mathfrak{A}(c)$	Matrix aller Akzeptanzwahrscheinlichkeiten, $\mathfrak{A}(c) := (A_{ij}(c))_{i, j \in S}$
$P_{ij}(c), i, j \in S$	Übergangswahrscheinlichkeit P_{ij} in einer Markov Kette, die für einen festen Kontrollparameter $c \in \mathbb{R}^+$ einen Metropolis Algorithmus (IMA und NMA) modelliert
$P_{trans}(c) = \mathfrak{P}(c)$	Matrix P_{trans} der Übergangswahrscheinlichkeiten in einer Markov Kette, die für einen festen Kontrollparameter $c \in \mathbb{R}^+$ einen Metropolis Algorithmus (IMA und NMA) modelliert, $P_{trans}(c) := (P_{ij}(c))_{i, j \in S}$
$P_{ij}^n(c), i, j \in S, n \in \mathbb{N}, n > 1$	höhere Übergangswahrscheinlichkeit P_{ij}^n in einer Markov Kette, die für einen festen Kontrollparameter $c \in \mathbb{R}^+$ einen Metropolis Algorithmus (IMA und NMA) modelliert

$q_i(c), i \in S$
 $q(c)$

$Dom_x \subset R_D, x \in R_D$
 $Ind_x \subseteq R_D, x \in R_D$
 $NDom_x := R_D \setminus$
 $(Dom_x \cup Ind_x \cup \{x\}), x \in R_D$
 $\{Y_k\}_{k \in \mathbb{N}_0}$

$x_k, k \in \mathbb{N}_0$

$MA_k, k \in \mathbb{N}_0$

$\underline{L}_k \in \mathbb{N}_0, k \in \mathbb{N}_0$

$\bar{t}_k \in \mathbb{N}, k \in \mathbb{N}_0$
 $S_k \subseteq \{x_k\} \times (R_D \setminus \{x_k\}), k \in \mathbb{N}_0$
 $R_k \subseteq (R_D \setminus \{x_k\}), k \in \mathbb{N}_0$
 $Dom_k \subseteq R_k, k \in \mathbb{N}_0$
 $Ind_k \subseteq R_k, k \in \mathbb{N}_0$
 $NDom_k := R_k \setminus (Dom_k \cup Ind_k),$
 $k \in \mathbb{N}_0$
 $L_k \subseteq R_D, k \in \mathbb{N}_0$

$s_k \in \mathbb{N}, k \in \mathbb{N}_0$
 $x_k^{i_k} \in L_k, k \in \mathbb{N}_0, i_k \in \{1, \dots, s_k\}$

$MA_k^{i_k}, k \in \mathbb{N}_0, i_k \in \{1, \dots, s_k\}$

$\underline{L}_k^{i_k} \in \mathbb{N}_0, k \in \mathbb{N}_0, i_k \in \{1, \dots, s_k\}$

$\bar{t}_k^{i_k} \in \mathbb{N}, k \in \mathbb{N}_0, i_k \in \{1, \dots, s_k\}$
 $numcont \in \mathbb{N}$

$count_k^{i_k} \in \mathbb{N}, k \in \mathbb{N}_0,$
 $i_k \in \{1, \dots, s_k\}$
 $S_k^{i_k} \subseteq \{x_k^{i_k}\} \times (R_D \setminus \{x_k^{i_k}\}), k \in \mathbb{N}_0,$
 $i_k \in \{1, \dots, s_k\}$

$q_i(c) := \lim_{n \rightarrow \infty} P_{ji}^n(c) \quad \forall j \in S$ (falls existent)
stationäre Verteilung der Markov Kette, die für einen
festen Kontrollparameter $c \in \mathbb{R}^+$ einen Metropolis Al-
gorithmus (IMA und NMA) modelliert (falls existent),
 $q(c) := (q_i(c))_{i \in S}$

Menge aller Alternativen $y \in R_D$, die x dominieren
Menge aller Alternativen $y \in R_D, y \neq x$, mit $y \succ x$
Menge aller von x dominierten Alternativen $y \in R_D$

stochastischer (Super)prozeß, der in Abschnitt 3.2.8.2 in
mehreren Formen definiert wird

Referenzalternative $\in R_D$, für die eine Metropolis-
Prozedur (NMA) dominierende Alternativen sucht
Metropolis Algorithmus (NMA), der für x_k nach domi-
nierenden Alternativen sucht

erste Iteration von MA_k , deren Standpunkt das Ergebnis
von MA_k liefern könnte (benutzerdefiniert)

maximal erlaubte Iterationszahl für $MA_k, \bar{t}_k \geq \underline{L}_k$

Suchraum von MA_k

$S_k := \{x_k\} \times R_k$

Menge aller Alternativen $y \in R_k$, die x_k dominieren

Menge aller Alternativen $y \in R_k$ mit $y \succ x_k$

Menge aller von x_k dominierten Alternativen $y \in R_k$

Alternativenliste mit $x \succ y \quad \forall x, y \in L_k$, wobei L_k "bes-
ser" als L_{k-1} ist für alle $k \in \mathbb{N}$

$s_k := |L_k|$

Alternativen aus L_k , für die eine Metropolis-Prozedur
(NMA) dominierende Alternativen sucht, die in L_{k+1}
aufgenommen werden könnten

Metropolis Algorithmus (NMA) mit Referenzalternati-
ve $x_k^{i_k}$

erste Iteration von $MA_k^{i_k}$, deren Standpunkt das Ergebnis
von $MA_k^{i_k}$ liefern könnte (benutzerdefiniert)

maximal erlaubte Iterationszahl für $MA_k^{i_k}, \bar{t}_k^{i_k} \geq \underline{L}_k^{i_k}$

Anzahl der Einträge, die ein Metropolis Algorithmus
(NMA) der Stufe k an $L_{k+1}, k \in \mathbb{N}_0$, zu liefern versuchen
muß

Anzahl der Einträge der Liste L_{k+1} von $MA_k^{i_k}$

Suchraum von $MA_k^{i_k}$

$$R_k^{i_k} \subseteq (R_D \setminus \{x_k^{i_k}\}), k \in \mathbb{N}_0,$$

$$i_k \in \{1, \dots, s_k\}$$

$$Red_k^{i_k}, k \in \mathbb{N}_0, i_k \in \{1, \dots, s_k\}$$

$$Red_{y^j}, j \in \mathbb{N}_0$$

$$Dom_k^{i_k} \subseteq R_k^{i_k}, k \in \mathbb{N}_0,$$

$$i_k \in \{1, \dots, s_k\}$$

$$Ind_k^{i_k} \subseteq R_k^{i_k}, k \in \mathbb{N}_0,$$

$$i_k \in \{1, \dots, s_k\}$$

$$NDom_k^{i_k} := R_k^{i_k} \setminus (Dom_k^{i_k} \cup Ind_k^{i_k}),$$

$$k \in \mathbb{N}_0, i_k \in \{1, \dots, s_k\}$$

$$S_k^{i_k} := \{x_k^{i_k}\} \times R_k^{i_k}$$

sei in der Liste L_k der Stufe $k \geq 1$ $x_k^{i_k}$ ein Eintrag von $MA_{k-1}^{i_{k-1}}$, dann wird $R_{k-1}^{i_{k-1}}$ um die Menge $Red_k^{i_k}$ verkleinert auf $R_k^{i_k}$; in Stufe $k = 0$ reduziert $Red_0^{i_0}$ die Menge R_D auf $R_0^{i_0}$

y^j sei ein Kandidat für die Aufnahme in L_{k+1} , der von $MA_k^{i_k}$ (NMA), $k \in \mathbb{N}_0$, $i_k \in \{1, \dots, s_k\}$, stammt; dann ist die Menge Red_{y^j} die Vereinigung aus der Menge $E^{(y^j)D}$ und allen Mengen E^{zD} der $z \in R_D$, von denen bekannt ist, daß y^j sie dominiert

Menge aller Alternativen $y \in R_k^{i_k}$, die $x_k^{i_k}$ dominieren

Menge aller Alternativen $y \in R_k^{i_k}$ mit $y \succ x_k^{i_k}$

Menge aller von $x_k^{i_k}$ dominierten Punkte $y \in R_k^{i_k}$

Einleitung

In der Praxis bestehen viele **Entscheidungsprobleme** nicht nur aus **sehr vielen Handlungsalternativen**, sondern es gilt obendrein, mehrere **verschiedene Ziele** zu verfolgen. Dabei darf man kaum erwarten, daß es eine Handlungsalternative gibt, die für alle Ziele die optimale Alternative darstellt - die Zielsetzungen widersprechen sich häufig. Einige dieser Entscheidungsprobleme können mathematisch als **Mehrzieloptimierungsprobleme/Vektoroptimierungsprobleme** modelliert werden. Bei ihnen muß ein **Vektor aus skalaren Zielfunktionen, die sogenannte Mehrzielfunktion**, über einen bestimmten zulässigen Bereich $R \subset \mathbb{R}^m, m \in \mathbb{N}$, "optimiert" werden. Die Menge R heißt **Alternativenmenge**. Die mathematische Formulierung dieser Vektoroptimierungsprobleme lautet gewöhnlich:

$$\text{"opt"}_{x \in R} F(x), F(x) := \begin{pmatrix} f_1(x) \\ f_2(x) \\ \dots \\ f_n(x) \end{pmatrix}, \quad (0.1)$$
$$f_i : R \mapsto \mathbb{R}, x \mapsto f_i(x), i = 1, \dots, n.$$

Da die Mehrzielfunktion ein Vektor aus mehreren skalaren Zielfunktionen ist, setzt sich ihre Wertemenge, der Zielraum, aus Elementen der Menge $\mathbb{R}^n, n \in \mathbb{N}, n > 1$, zusammen. Weil \mathbb{R}^n nicht vollständig geordnet ist, ist nicht unmittelbar klar, wie man Optimalität definieren soll - in den meisten Mehrzielproblemen (hier Minimierungsprobleme) gibt es keine Alternative mit einem Zielwertvektor, der kleinergleich allen anderen zulässigen Zielwertvektoren ist. Es gibt verschiedene Definitionen der Optimalität, die in der Vektoroptimierung verbreitet sind. Die bekannteste unter ihnen ist die **Effizienz** oder **Paretooptimalität**. Ein zulässiger Punkt $x \in R$ heißt paretooptimal, falls es kein $y \in R$ gibt, das in allen Zielen mindestens genauso gut und in mindestens einem Ziel besser ist als x . $F(x)$ nennt man effizient. Kapitel 1 geht ausführlicher auf die Optimalität in der Vektoroptimierung ein.

Kapitel 2 gibt einen Überblick über bereits bekannte Methoden der Mehrzieloptimierung. Natürlich ist er nicht vollständig, aber der Leser wird die grundlegenden Motivationen und Ideen hinter den Vektoroptimierungsansätzen der Literatur kennenlernen. Man kann die Methoden grob klassifizieren, indem man darauf schaut, wann und wie die Präferenzen des Entscheidungsträgers, der bei der Bearbeitung eines Mehrzieloptimierungsproblems von den mathematischen Verfahren unterstützt wird, in die mathematischen Lösungsschritte eingehen. Die Literatur unterscheidet drei Arten von Methoden:

- Methoden mit a priori Präferenzäußerung,
- Methoden mit fortschreitender Präferenzäußerung,
- Generierungstechniken / Methoden mit a posteriori Präferenzäußerung.

Die kurz umrissenen Methoden werden kritisiert in Hinblick auf ihre Brauchbarkeit für die Lösung der speziellen Klasse von Vektoroptimierungsproblemen, die sich diese Arbeit zur Aufgabe gemacht hat. Aus den auf diese Weise herausgestellten Nachteilen der verschiedenen beschriebenen Mehrzieltechniken wird der Leser erkennen können, warum die Entwicklung einer neuen Generierungstechnik notwendig war, um diese Aufgabe zu erfüllen.

Diese Arbeit befaßt sich mit bestimmten sogenannten **diskret-kontinuierlichen Vektoroptimierungsproblemen**. Jedes $x \in R$ enthält Komponenten aus endlichen Teilmengen von \mathbb{R} und aus überabzählbaren Teilmengen von \mathbb{R} . Erstere wollen wir diskrete, letztere kontinuierliche Komponenten nennen:

$$\begin{aligned}
 R &:= C \times D, \\
 C &\subseteq \mathbb{R}^{m_1} \quad \text{mit } |C| = |\mathbb{R}|, \\
 D &\subset \mathbb{R}^{m_2} \quad \text{mit endlich vielen Elementen,} \\
 m_1, m_2 &\in \mathbb{N}, m_1 + m_2 = m.
 \end{aligned}
 \tag{0.2}$$

Die genaue Problemstellung erfolgt in Kapitel 3, Abschnitt 3.1.

Wie kann man in einem diskret-kontinuierlichen Mehrzieloptimierungsproblem zu einem Paretooptimum gelangen? Man versucht von einem beliebigen zulässigen Startpunkt aus so lange einem **Pfad dominierender Punkte** zu folgen, bis man ein Paretooptimum erreicht hat. Eine Alternative $y \in R$ **dominiert** eine Alternative $x \in R$, wenn sie in jedem Ziel mindestens so gute Funktionswerte wie x aufweist und in mindestens einem Zielfunktionswert x sogar überbietet. y ist "besser" als x . Das Mehrzieloptimierungsverfahren, welches in dieser Arbeit entwickelt wird, wählt sich stochastisch einen Startpunkt $x_0 \in R$ und sucht wiederum stochastisch nach einer Alternative $y_0 \in R$, die x_0 dominiert. Findet es eine solche Alternative y_0 , so wird diese neuer Referenzpunkt x_1 , für den das Verfahren erneut stochastisch nach einem dominierenden Punkt $y_1 \in R$ forscht. Wird so ein y_1 entdeckt, dient es als neuer Referenzpunkt x_2 , für den stochastisch nach einem dominierenden Punkt zu suchen ist... Über diese Folge immer besserer Alternativen x_0, x_1, x_2, \dots gelangt das Verfahren schließlich mit einer bestimmten Wahrscheinlichkeit zu einem Paretooptimum $x_k \in R, k \in \mathbb{N}_0$, zu dem natürlich kein dominierender Punkt existiert.

Selbverständlich ist das Ziel, nicht nur ein Paretooptimum, sondern möglichst viele zu erlangen. Die erzeugte Menge von Paretooptima soll die Menge aller Paretooptima in R gut repräsentieren, d.h. eine gute Vorstellung von ihr vermitteln. Dies ist auf zwei Arten erreichbar:

- Nachdem sich das Verfahren über einen Pfad $x_0, x_1, x_2, \dots, x_k$ zu einem Punkt x_k vorgearbeitet hat, den es als Paretooptimum einstuft, sucht es von diesem x_k aus stochastisch nach einer möglichst "guten" Alternative $y_k \in R$. Wir wollen darunter einen zulässigen Punkt verstehen, der nicht von x_k dominiert wird, also nicht "schlechter" ist als x_k . Dieses y_k dient dem Verfahren als neuer zulässiger Startpunkt \hat{x}_0 , von dem aus es dann stochastisch einen Pfad dominierender Punkte $\hat{x}_0, \hat{x}_1, \hat{x}_2, \dots, \hat{x}_p, p \in \mathbb{N}_0$, erzeugt, der es zu einem weiteren Punkt \hat{x}_p führt, den es für paretooptimal hält. Vom neuen Paretooptimum aus kann das Verfahren analog fortfahren. Auf diese Weise können hintereinander Pfade dominierender Punkte beschrritten und nach und nach Paretooptima eingesammelt werden.
- Das Verfahren kann jedoch auch stochastisch mehrere verschiedene Startpunkte $x_0 \in R$ wählen und parallel Pfade $x_0, x_1, x_2, \dots, x_k$ generieren, um zu mehreren Paretooptima x_k zu gelangen.

Für die Umsetzung der beschriebenen Suchidee muß man sich vor allem mit der Tatsache auseinandersetzen, daß der zulässige Bereich der zu lösenden Klasse von Vektoroptimierungsproblemen **diskret-kontinuierlich** ist, d.h. die Alternativen sowohl aus diskreten als auch kontinuierlichen Komponenten bestehen. Diese Arbeit darf davon ausgehen, daß C kompakt ist und für jede Festlegung $\bar{x}_D \in D$ der diskreten Variationsmöglichkeiten das zugehörige kontinuierliche Vektoroptimierungsproblem

$$\begin{aligned}
 & \text{"min"}_{x_C \in C} F(x_C, \bar{x}_D) = \text{"min"}_{x_C \in C} F^{\bar{x}_D}(x_C), \\
 & F(x_C, \bar{x}_D) := \begin{pmatrix} f_1(x_C, \bar{x}_D) \\ f_2(x_C, \bar{x}_D) \\ \dots \\ f_{n-1}(x_C, \bar{x}_D) \\ f_n(x_C, \bar{x}_D) \end{pmatrix}, F^{\bar{x}_D}(x_C) := \begin{pmatrix} f_1^{\bar{x}_D}(x_C) \\ f_2^{\bar{x}_D}(x_C) \\ \dots \\ f_{n-1}^{\bar{x}_D}(x_C) \\ f_n^{\bar{x}_D}(x_C) \end{pmatrix}, \quad (0.3) \\
 & f_i^{\bar{x}_D} : C \mapsto \mathbb{R}, x_C \mapsto f_i(x_C, \bar{x}_D), i = 1, \dots, n,
 \end{aligned}$$

zweimal stetig differenzierbare Zielfunktionen $f_i^{\bar{x}_D}, i = 1, \dots, n$, besitzt. Das entwickelte Verfahren macht sich diesen Umstand zunutze und zerlegt jede Suche von einem Punkt $x \in R$ aus nach einer ihn dominierenden Alternative in diskrete und kontinuierliche Suchschritte. Bei der stochastischen Generierung eines Punktes $y \in R$

- erzeugt das Verfahren zunächst stochastisch ein $y_D \in D$, also eine zulässige diskrete Parametervariation (diskreter Suchschritt),
- dann läßt es von einem stochastischen kontinuierlichen Mehrzieloptimierungsverfahren das kontinuierliche Vektoroptimierungsproblem (0.3) für $\bar{x}_D = y_D$ lösen, sofern es nicht bei einer früheren Generierung von y_D bereits gelöst wurde. Dieses erarbeitet eine endliche Lösungsmenge $\hat{L}^{y_D} \subseteq C$, die eine numerische Approximation einer endlichen Teilmenge der tatsächlichen Menge aller Paretooptima $L^{y_D} \subseteq C$ von (0.3) darstellt (kontinuierlicher Suchschritt).

- Es ergibt sich die diskrete Menge

$$\hat{E}^{y_D} := \{(y_C, y_D) | y_C \in \hat{L}^{y_D}\} \subseteq R, \quad (0.4)$$

eine numerische Approximation einer endlichen Teilmenge aller **kontinuierlich-effizienten Punkte** $E^{y_D} := \{(y_C, y_D) | y_C \in L^{y_D}\} \subseteq R$ zu festem y_D , die gespeichert wird, so daß bei Bedarf auch im weiteren Suchverlauf noch darauf zugegriffen werden kann. Aus dieser Menge wählt das Verfahren stochastisch einen Punkt y (diskreter Suchschritt).

Die anfallenden diskreten Suchschritte übernimmt ein speziell konstruierter **Metropolis Algorithmus**.

Der Kern des in dieser Arbeit entwickelten Verfahrens zur Lösung einer speziellen Klasse diskret-kontinuierlicher Vektoroptimierungsprobleme liegt in zwei Aspekten:

- der abwechselnden Durchführung diskreter und kontinuierlicher Suchschritte und
- der Hintereinanderausführung speziell konstruierter Metropolis Algorithmen. Je ein solcher Metropolis Algorithmus übernimmt die diskreten Schritte während der Suche nach einem dominierend Punkt für einen Referenzpunkt $x \in R$, erzeugt also einen Punkt zu Punkt Übergang auf dem Pfad dominierender Punkte von einem Startpunkt zu einem Paretooptimum.

Insgesamt handelt es sich also bei dem Verfahren um eine **Verallgemeinerung der skalaren diskreten Optimierungsmethode Metropolis Algorithmus auf die Anwendung für diskret-kontinuierliche Vektoroptimierungsprobleme**. Dieser verallgemeinerte Metropolis Algorithmus läßt anfallende kontinuierliche Sub-Vektoroptimierungsprobleme von bereits bekannten kontinuierlichen Mehrzielverfahren bearbeiten. Daraus ergeben sich zwei vorteilhafte Umstände, die das Verfahren natürlich ausnützt:

1. Bei der Lösung des diskret-kontinuierlichen Problems muß nicht auf wertvolle Gradienteninformation, die im stetigen Teil des diskret-kontinuierlichen Mehrzielproblems, d.h. in den kontinuierlichen Subproblemen steckt, verzichtet werden. Bei dem Versuch, das diskret-kontinuierliche Problem ohne kontinuierliche Suchschritte zu lösen, wäre vorhandene Gradienteninformation nicht nutzbar.
2. Aufgrund der speziellen Struktur der Menge $R_D := \cup_{x_D \in D} E^{x_D} \subseteq R$ ist es mit der Kenntnis von Punkten, die von $x \in R$ dominiert werden, möglich, den Suchraum für den Metropolis Algorithmus, der auf R_D nach einem x dominierenden Punkt sucht, im Vergleich zu $R_D \setminus \{x\}$ erheblich zu reduzieren.

In Kapitel 3 wird der Leser den Mehrziel-Metropolis Algorithmus genau kennenlernen.

Um dem Leser vor Augen zu führen, wie der in Kapitel 3 erarbeitete Algorithmus auf Probleme der Praxis angewandt werden kann, werden in Kapitel 4 einige Anwendungen beschrieben. Hier wird ihm auch eine konkrete, problemspezifische Konstruktion des Mehrziel-Metropolis Algorithmus demonstriert. Testergebnisse zeigen die Qualitäten des diskret-kontinuierlichen Mehrzielverfahrens.

Danksagungen

Ich möchte meinem Doktorvater, Herrn Prof. Dr. Dr. Stefan Schäffler, für sein Interesse an meiner Arbeit, seinen fachlichen Rat und seine große organisatorische Unterstützung danken, sowie Herrn Prof. Dr. Albert Gilg (Siemens AG München), daß er diese Arbeit in Form einer Doktorandenstelle finanzierte und mir den mit den zur Erstellung notwendigen technischen Hilfsmitteln ausgerüsteten Arbeitsplatz zur Verfügung stellte.

Dankbar bin Herrn Dr. Dr. Reinhart Schultz, daß er den Fortschritt der Arbeit stets mit wertvollen mathematischen und programmiertechnischen Hinweisen begleitet hat und Herrn Dr. Michael Greiner für seine Notdienstleistungen bei Computerproblemen. Sie alle haben eine gute Arbeitsatmosphäre erzeugt und mich stets vor Panik bewahrt.

Ein weiteres großes Dankeschön geht an meinen Mann für das sorgfältige Durchlesen der Arbeit und seine wertvollen Tips, desweiteren an meinen Mitstudenten, Herrn Dipl. Math. Andreas Fröhlich, und an meine Mutter für die Suche nach Rechtschreib- und Tippfehlern.

Bei Herrn Prof. Dr. Claus Hillermeier möchte ich mich ganz speziell dafür bedanken, daß er sich freundlicherweise bereit erklärt hat, meine Dissertation als externer Berichter zu prüfen.

Kapitel 1

Grundlagen der Vektoroptimierung

Gegeben sei ein Entscheidungsproblem. Eine Person, wir nennen sie **Entscheidungsträger**, sei mit sehr vielen Handlungsalternativen konfrontiert. Der Entscheidungsträger möchte verschiedene Ziele erreichen. Jede Entscheidung für eine bestimmte Handlungsalternative hat ein bestimmtes Ergebnis zur Folge, erfüllt die Wünsche des Entscheidungsträgers in Hinblick auf seine verschiedenen Ziele in unterschiedlichem Maße. Da er die Handlungsmöglichkeiten nicht nur bezüglich eines Zieles, sondern in Hinblick auf mehrere Ziele beurteilen muß, kann sich der Entscheidungsträger angesichts der resultierenden Effektivvielfalt außerstande sehen, die Alternativen nach dem Grad ihrer Zielerfüllung vollständig zu ordnen. Falls die beschriebene Situation sich mathematisch modellieren läßt, kann die Mathematik in Form von **Vektoroptimierung** oder **Mehrzieloptimierung** dem Entscheidungsträger einen Weg aus seinem Dilemma zeigen. Die folgenden Abschnitte dieses Kapitels halten sich an Definitionen und Sätze aus [22] und [24].

1.1 Die Alternativenmenge

Alle zulässigen Handlungsalternativen, unter denen der Entscheidungsträger wählen kann, werden in der Menge R zusammengefaßt. R wird gewöhnlich **Alternativenmenge** oder einfach **zulässiger Bereich** genannt.

Vektoroptimierung versucht, den Lösungsvorgang bei einem Entscheidungsproblem in mathematischer Formulierung zu unterstützen. Häufig ist dabei $R \subseteq \mathbb{R}^m$, $m \in \mathbb{N}$. Genaugenommen enthält R nicht die Handlungsalternativen, sondern ihre irgendwie codierten Repräsentanten, meist Vektoren $x := (x_1, x_2, \dots, x_m)$, $m \in \mathbb{N}$, reeller Zahlen $x_i \in \mathbb{R}$, $i = 1, \dots, m$. Die Werte der x_i , $i = 1, \dots, m$, zu variieren, heißt verschiedene Alternativen zu betrachten. Daher kann R auch **Menge der Entscheidungsvariablen** genannt werden.

R kann **kontinuierlich**, d.h. eine Menge mit Mächtigkeit $|R| = |\mathbb{R}|$, oder **diskret**, d.h. eine Menge mit abzählbar vielen Elementen sein sowie (gemischt) **diskret-kontinuierlich**, d.h.

$$\begin{aligned} R &:= C \times D, \\ C &\subseteq \mathbb{R}^{m_1} \quad \text{mit } |C| = |\mathbb{R}|, \\ D &\subseteq \mathbb{R}^{m_2} \quad \text{mit abzählbar vielen Elementen,} \\ m_1, m_2 &\in \mathbb{N}, m_1 + m_2 = m. \end{aligned} \tag{1.1}$$

Falls R in irgendeiner Weise durch Nebenbedingungen beschränkt ist, drücken in der mathematischen Formulierung des gegebenen Entscheidungsproblems (**Nebenbedingungs**)**funktionen** diese Nebenbedingungen aus:

$$\begin{aligned} g(x) &\leq 0, \\ h(x) &= 0, \\ g &: \mathbb{R}^m \mapsto \mathbb{R}^k, x \mapsto g(x), \\ h &: \mathbb{R}^m \mapsto \mathbb{R}^l, x \mapsto h(x), \\ &k, l \in \mathbb{N}. \end{aligned}$$

Eine Definition von R kann daher folgendermaßen aussehen:

$$R := \{(x_C, x_D) \in C \times D \mid g(x_C, x_D) \leq 0, h(x_C, x_D) = 0\}.$$

1.2 Die Kriterienmenge

Die unterschiedlichen Konsequenzen, die sich aus der Entscheidung für die einzelnen Handlungsalternativen ergeben, müssen irgendwie vergleichbar sein. Daher ist es nötig, mindestens ein Kriterium einzuführen, anhand dessen man in einem formalen Modell jede Alternative im Hinblick auf ihre Auswirkungen bewertet. Notwendigerweise müssen die Auswirkungen einer Entscheidung für eine bestimmte Alternative quantitativ meßbar sein. Synonym nennt man ein Bewertungskriterium auch Ziel. Quantitative Bewertung kann eindimensional (skalare Optimierung mit einem Kriterium/Ziel) oder mehrdimensional (Vektoroptimierung mit mehreren Kriterien/Zielen) vorgenommen werden. In mathematischen Modellen bewerten Funktionen, die sogenannten **Zielfunktionen**, die Alternativen:

$$F(x) := \begin{pmatrix} f_1(x) \\ f_2(x) \\ \dots \\ f_n(x) \end{pmatrix}, x \in R, n \in \mathbb{N}, f_i : R \mapsto \mathbb{R}, x \mapsto f_i(x), i = 1, \dots, n.$$

Der Wert $f_i(x)$, $i = 1, \dots, n$, drückt die Leistung der Alternative x in Bezug auf das i -te Kriterium/Ziel aus. Im Falle der Mehrzieloptimierung gilt $n \geq 2$.

Die Menge $F(R) := \{F(x), x \in R\}$ heißt **Kriterienmenge** oder **Zielraum**. Die Kriterienmenge repräsentiert die quantitative Leistung aller Handlungsalternativen in den verschiedenen Kriterien/Zielen.

Da $n \geq 2$, $F : R \mapsto \mathbb{R}^n, x \mapsto F(x)$, ein Vektor aus Zielfunktionen, genannt **Mehrzielfunktion**, ist, erhalten die Begriffe "Vektoroptimierung" und "Mehrzieloptimierung" ihre Berechtigung.

1.3 Das Vektoroptimierungsproblem

Die mathematische Formulierung von Problemen, die mit Hilfe der Vektoroptimierung gelöst werden sollen, lautet gewöhnlich

$${}''opt_{x \in R}'' F(x), F(x) := \begin{pmatrix} f_1(x) \\ f_2(x) \\ \dots \\ f_n(x) \end{pmatrix}. \quad (1.2)$$

Die Zielfunktionen f_1, \dots, f_n sind irgendwie simultan zu optimieren. Gesucht ist also eine Menge $R^* \subseteq R$, deren Elemente einem noch genau zu definierenden Optimalitätskriterium genügen, das in jedem Kriterium/Ziel möglichst gute Leistung fordern wird. Unter der Annahme, daß für alle n Zielfunktionen Extrema gefunden werden müssen und da man eine zu maximierende Zielfunktion durch Multiplikation mit -1 leicht in eine zu minimierende umwandeln kann, ist es möglich, das allgemeine Vektoroptimierungsproblem folgendermaßen hinzuschreiben:

$$\text{"min"}_{x \in R} F(x) = \text{"min"}_{z \in F(R)} z. \quad (1.3)$$

Die Menge der optimalen Lösungen aus R in Hinblick auf das i -te Ziel, $i = 1, \dots, n$, ohne Berücksichtigung aller anderen Kriterien, sei

$$Min_i := \{\hat{x} \in R \mid f_i(\hat{x}) \leq f_i(x) \quad \forall x \in R\}.$$

Falls $\bigcap_{i=1}^n Min_i$ eine oder mehrere Alternativen enthält, besteht zwischen den n Zielen kein Konflikt; es gibt mindestens eine **ideale Alternative**, die bezüglich aller Ziele die beste Leistung bringt. In der Realität existiert aber meistens keine ideale Alternative. Die Theorie nutzt jedoch häufig den Punkt $(f_1(x_1), \dots, f_n(x_n)), x_i \in Min_i \quad \forall i = 1, \dots, n$, als Referenzpunkt in $F(R)$, um Alternativen zu finden, die ein spezielles Optimalitätskriterium erfüllen.

1.4 Optimalität in der Vektoroptimierung

Ein kurzer Blick auf den Fall $n = 1$ zeigt, daß der Zielraum $F(R) := f_1(R) \subseteq \mathbb{R}$ vollständig geordnet ist, d.h. für alle $z_1, z_2 \in \mathbb{R}$ gilt immer

$$\text{entweder } z_1 \leq z_2 \quad \text{oder } z_2 \leq z_1.$$

Je zwei Alternativen $x_1, x_2 \in R$ sind hinsichtlich ihrer Ziel-Auswirkungen leicht vergleichbar. In einem Minimierungsproblem sucht man nach allen $\hat{x} \in R$ mit

$$F(\hat{x}) \leq F(x) \quad \forall x \in R.$$

In skalaren Optimierungsproblemen wird also die topologische Ordnung auf $F(R) \subseteq \mathbb{R}$ verwendet, um Optimalität zu definieren.

Im Gegensatz zur skalaren Optimierung ist der Begriff "Optimalität" in der Vektoroptimierung (Fall $n > 1$) nicht unmittelbar einsichtig. Dies ergibt sich aus der Tatsache, daß der Zielraum $F(R) \subseteq \mathbb{R}^n$ nicht vollständig geordnet ist - man kann nicht alle Alternativen anhand ihrer Zielfunktionswerte miteinander vergleichen. Angesichts dieses Problems muß man einen anderen Weg einschlagen, um Elemente $z_1, z_2 \in F(R)$ und daher Alternativen $x_1, x_2 \in R$ mit $F(x_1) = z_1$ und $F(x_2) = z_2$ gegeneinander abzuwägen.

Natürlich kann man zu diesem Zwecke die Halbordnung \leq auf \mathbb{R}^n als Ausgangspunkt nehmen. Die **Dominanz** ist eine Relation, die auf dieser Halbordnung basiert, jedoch weder Reflexivität noch Antisymmetrie aufweist:

Definition 1.4.1. Ein Punkt $u \in F(R)$ **dominiert** einen Punkt $v \in F(R)$, falls

$$u_i \leq v_i \quad \forall i = 1, \dots, n \quad \text{und} \quad \exists k \in \{1, \dots, n\} : \quad u_k < v_k. \quad (1.4)$$

Synonym kann man sagen, v sei **inferior zu** u . △

Dominiert ein Vektor u von Zielfunktionswerten einen Vektor v von Zielfunktionswerten, so wird der Entscheidungsträger die durch u ausgedrückte Zielausprägung der durch v ausgedrückten Zielausprägung vorziehen, da die Leistungen u_i in allen Zielen $i = 1, \dots, n$ mindestens so gut wie die Leistungen v_i sind und bezüglich des Kriteriums k die Ausprägung u_k sogar besser als die Ausprägung v_k ist. Wenn der Entscheidungsträger zwischen den Alternativen $x, y \in R$ mit $F(x) = u$ und $F(y) = v$ zu wählen hat, wird er sich natürlich für x entscheiden. Wie man sieht, kann also das Prinzip der Dominanz, das ja zunächst für $F(R)$ definiert ist, auch genutzt werden, um Alternativen paarweise in R zu vergleichen:

Definition 1.4.2. Eine Alternative $x \in R$ **dominiert** eine Alternative $y \in R$ (hinsichtlich der Zielfunktionswerte), falls

$$f_i(x) \leq f_i(y) \quad \forall i = 1, \dots, n \quad \text{und} \quad \exists k \in \{1, \dots, n\} : f_k(x) < f_k(y). \quad (1.5)$$

Synonym kann man sagen, y sei **inferior zu** Alternative x (hinsichtlich der Zielfunktionswerte). △

Bemerkung 1.4.3. Die "Dominanz" nach Definition 1.4.1 vergleicht zwei Elemente aus $F(R)$. Korrekterweise müßte also der Begriff "Dominanz" nach Definition 1.4.2 immer durch den Zusatz "hinsichtlich der Zielfunktionswerte" ergänzt werden, da ja zwei Elemente aus R anhand ihrer Zielfunktionswerte in $F(R)$ verglichen werden. Der Einfachheit halber sparen wir uns diesen Zusatz jedoch immer, wenn von Dominanz die Rede ist. △

Von der Dominanz leiten sich drei Relationen, nämlich R_{\prec}, R_{\succ} und $R_{\asymp} \subseteq R \times R$ her. x und y seien aus R :

$$xR_{\prec}y \quad \text{oder} \quad x \prec y, \quad \text{falls } x \text{ Alternative } y \text{ dominiert,} \quad (1.6)$$

$$xR_{\succ}y \quad \text{oder} \quad x \succ y, \quad \text{falls } y \text{ Alternative } x \text{ dominiert,} \quad (1.7)$$

$$xR_{\asymp}y \quad \text{oder} \quad x \asymp y, \quad \text{sonst.} \quad (1.8)$$

Bemerkung 1.4.4.

$$x \prec y \iff F(x) \leq F(y) \quad \text{und} \quad F(y) \not\leq F(x),$$

$$x \succ y \iff F(y) \leq F(x) \quad \text{und} \quad F(x) \not\leq F(y),$$

$$x \asymp y \iff F(x) \not\leq F(y) \quad \text{und} \quad F(y) \not\leq F(x)$$

$$\text{oder} \quad F(x) = F(y). \quad \text{△}$$

Der Begriff "dominieren" und das Symbol " \prec " stehen synonym für die Aussage "besser", " \succ " für "schlechter" und " \asymp " für "genauso gut", ohne daß dabei Präferenzen des Entscheidungsträgers, welche Ziele er für wichtiger und welche für unwichtiger hält, mit einbezogen werden. Da die Relationen \prec und \succ nicht reflexiv und \asymp nicht transitiv sind (z.B. sichtbar bei $F(x) = (1, 0, 2), F(y) = (2, -1, 1), F(z) = (1, 0, 1)$), ist keine dieser Relationen eine (Halb)ordnung, und wir haben somit strenggenommen immer noch keine vollständige Ordnung auf R gefunden. Dennoch besitzen wir jetzt ein Werkzeug, mit dessen Hilfe man verschiedene Arten der Optimalität definieren kann. Da die drei Relationen paarweise disjunkt sind und $R_{\prec} \cup R_{\succ} \cup R_{\asymp} = R \times R$ gilt, sind wir jetzt in der Lage, eine abwägende Aussage über jedes beliebige Alternativenpaar zu machen, auch wenn nicht je zwei Alternativen bezüglich einer einzigen Relation in R vergleichbar sind: Entweder gilt $x \prec y$ oder $x \succ y$ oder $x \asymp y$.

1.4.1 Effizienz

Es liegt nahe, eine Alternative als optimal zu erachten, falls es keine bessere, d.h. dominierende Alternative gibt. Diese Überlegung führt zum meistverbreiteten Optimalitätsbegriff in der Vektoroptimierung, der **Effizienz** oder **Paretooptimalität**; statt **effiziente Lösung** sagt man auch häufig **nichtdominierte Lösung** oder **nichtinferiore Lösung**.

Definition 1.4.5. Eine Alternative $x \in R$ heißt **paretooptimal** oder **(funktional)effizient**, falls es kein $y \in R$ gibt mit

$$f_i(y) \leq f_i(x) \quad \forall \quad i = 1, \dots, n \quad \text{und} \quad \exists k \in \{1, \dots, n\} : f_k(y) < f_k(x), \quad (1.9)$$

d.h. kein y gibt mit $y \prec x$. Oder anders ausgedrückt:

$$(F(x) + (R_-^n \cup \{0\})) \cap F(R) = \{F(x)\}.$$

△

Bemerkung 1.4.6. Wie in der Literatur üblich und auch der Einfachheit halber gebraucht man die Begriffe "effizient", "nichtinferior" und "nichtdominiert" nicht nur für die Charakterisierung von Punkten der Menge $F(R)$, sondern auch synonym für den Begriff "paretooptimal", also beim Umgang mit Punkten in R . Genaugenommen müßte man natürlich bei Punkten in R von "funktionaleffizient", "funktionalnichtinferior" und "funktionalnichtdominiert" sprechen - sehr unhandliche Wörter. In dieser Arbeit wird immer erkennbar sein, ob von Alternativen in R oder den zugehörigen Vektoren von Zielwerten in $F(R)$ die Rede ist.

△

Paretooptimalität ist der Optimalitätsbegriff der Vektoroptimierung, den diese Arbeit verwendet.

Bekanntlich sind in der skalaren Optimierung **lokale Optima** für Algorithmen, die immer der Richtung des steilsten Abstieges folgen, gewissermaßen "Fallgruben". In einer "Nachbarschaft" eines lokalen Minimierers existiert kein Punkt mit kleinerem Zielfunktionswert, "irgendwo weiter entfernt" existieren jedoch durchaus Punkte mit besseren Zielfunktionswerten. Einmal gefangen in einem lokalen Minimierer haben die Algorithmen des steilsten Abstieges keine Chance mehr, so einen Punkt zu verlassen, um danach eine global optimale Lösung zu erreichen. Leider existiert dieses Phänomen auch in der Vektoroptimierung mit gleicher negativer Auswirkung:

Definition 1.4.7. Eine Alternative $x \in R$ wird als **lokal paretooptimal** oder **lokal funktionaleffizient** bezeichnet, falls es für ein $\varepsilon > 0$ kein $y \in N_\varepsilon(x)$, $N_\varepsilon(x) := \{y \in R \mid \|x - y\|_2 < \varepsilon\}$, gibt mit

$$f_i(y) \leq f_i(x) \quad \forall \quad i = 1, \dots, n \quad \text{und} \quad \exists k \in \{1, \dots, n\} : f_k(y) < f_k(x). \quad (1.10)$$

Oder anders ausgedrückt:

$$(F(x) + (R_-^n \cup \{0\})) \cap F(N_\varepsilon(x)) = \{F(x)\}.$$

△

Abbildung 1.1: Ein deterministischer Algorithmus, der von jedem Punkt x aus der Suchrichtung $s := -\nabla F(x)$ folgt, kann nach Anlaufen eines lokalen Minimierers x_l diesen nicht mehr verlassen und daher keinen globalen Minimierer x_g erreichen.

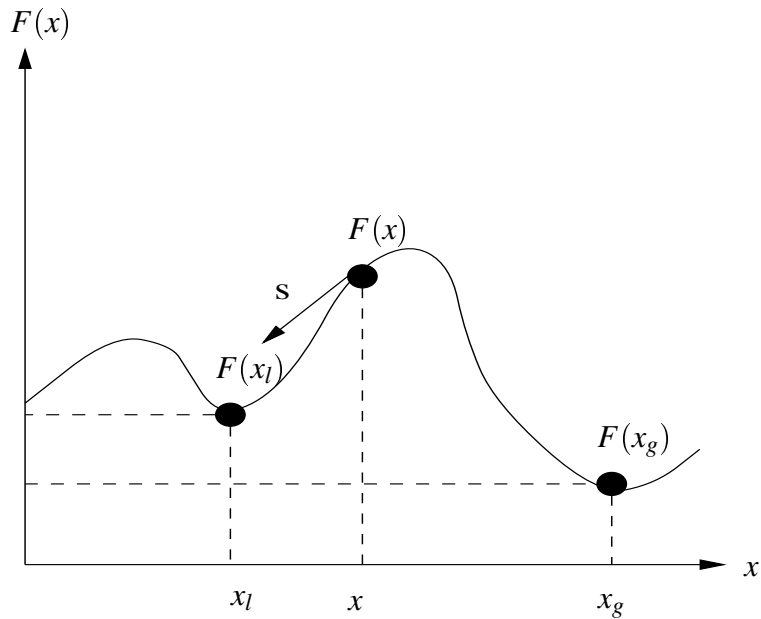
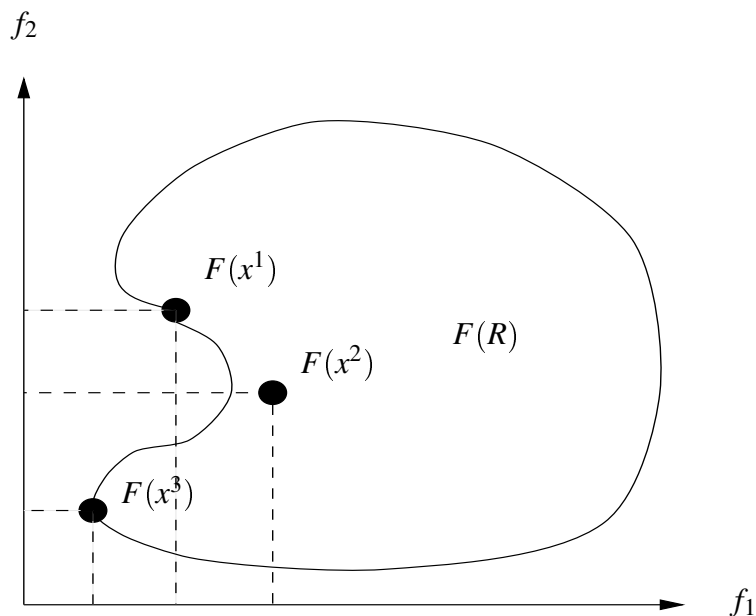


Abbildung 1.2: (lokale) Paretooptimalität in einem nichtkonvexen Zielraum $F(R)$ bei stetigen Zielfunktionen: x^1 ist lokal paretooptimal, ebenso x^2 , für das $F(N_\varepsilon(x^2) \setminus \{x^2\}) \cap (F(x^2) + (\mathbb{R}_-^2 \cup \{0\})) = \emptyset$ für ein $\varepsilon > 0$ gelte; x^3 ist global paretooptimal.



1.4.1.1 Eigenschaften der Effizienzmenge

Der Ausdruck "Effizienzmenge" steht in dieser Arbeit für die Menge der Zielvektoren aller Paretooptima des Vektoroptimierungsproblems

$$\text{"min"}_{x \in R} F(x), F : R \mapsto \mathbb{R}^n, x \mapsto F(x), R \neq \emptyset, R \subseteq \mathbb{R}^m, m, n \in \mathbb{N}, n \geq 2. \quad (1.11)$$

Notation 1.4.8. $E(F(R), \leq)$ bezeichne die Effizienzmenge in $F(R)$ in der Problemstellung (1.11). △

Satz 1.4.9. $E(F(R), \leq)$ liegt auf dem Rand $\delta(F(R))$ von $F(R)$.

Bemerkung 1.4.10. $\delta(F(R))$ besteht aus den Punkten aus \mathbb{R}^n , von denen jede Umgebung sowohl Punkte aus $F(R)$ als auch aus $\mathbb{R}^n \setminus F(R)$ enthält. △

Beweis: Es gelte $F(\bar{x}) \in E(F(R), \leq)$ und $F(\bar{x}) \notin \delta(F(R))$. Dann gibt es zu einem $\varepsilon > 0$ eine Menge $N_\varepsilon(0) := \{z \in \mathbb{R}^n \mid \|z\|_2 < \varepsilon\}$ mit $F(\bar{x}) + N_\varepsilon(0) \subseteq F(R)$. Für ein $0 \neq c \in (R_-^n \cup \{0\})$ wähle man ein $\lambda > 0$ mit $\lambda c \in N_\varepsilon(0)$. Dann gilt $F(\bar{x}) + \lambda c \in F(R)$. Wegen $\lambda c \in (R_-^n \cup \{0\})$ erhalten wir

$$F(\bar{x}) \neq (F(\bar{x}) + \lambda c) \in [F(R) \cap (F(\bar{x}) + (R_-^n \cup \{0\}))],$$

was der angenommenen Effizienz von $F(\bar{x})$ widerspricht. ■

Nach dieser Erkenntnis über die Lage der Effizienzmenge $E(F(R), \leq)$ in $F(R)$ kann man in speziellen Fällen eine interessante Aussage über die Gestalt der Effizienzmenge machen:

Satz 1.4.11. $F(R) \neq \emptyset$ sei konvex, abgeschlossen und beschränkt. Dann ist die Menge $E(F(R), \leq)$ zusammenhängend, d.h. es gibt keine zwei offenen Mengen $O_1, O_2 \subseteq \mathbb{R}^n$, für die gilt

$$\begin{aligned} E(F(R), \leq) &\subseteq O_1 \cup O_2, & E(F(R), \leq) \cap O_1 &\neq \emptyset, \\ E(F(R), \leq) \cap O_2 &\neq \emptyset, & E(F(R), \leq) \cap O_1 \cap O_2 &= \emptyset. \end{aligned}$$

Aussagen über die Gestalt der Effizienzmenge machen natürlich nur Sinn, wenn die Existenz mindestens eines Paretooptimums garantiert ist.

Satz 1.4.12. Es sei $\emptyset \neq F(R) \subset \mathbb{R}^n$. Falls $F(R)$ einen kompakten Ausschnitt enthält, d.h. für ein $\hat{z} \in F(R)$ ist die Menge

$$\{z \in F(R) : z \leq \hat{z}\} = (\hat{z} + (R_-^n \cup \{0\})) \cap F(R)$$

kompakt, dann gilt $E(F(R), \leq) \neq \emptyset$.

Leser, die sich für die Beweise der letzten beiden Sätze interessieren, finden diese für eine allgemeiner gehaltene Formulierung dieser Sätze in [24].

Die noch folgenden Definitionen von Optimalität sollen einen Eindruck von der Vielfalt der Möglichkeiten, Optimalität in der Vektoroptimierung zu interpretieren, vermitteln.

1.4.2 Schwache Effizienz

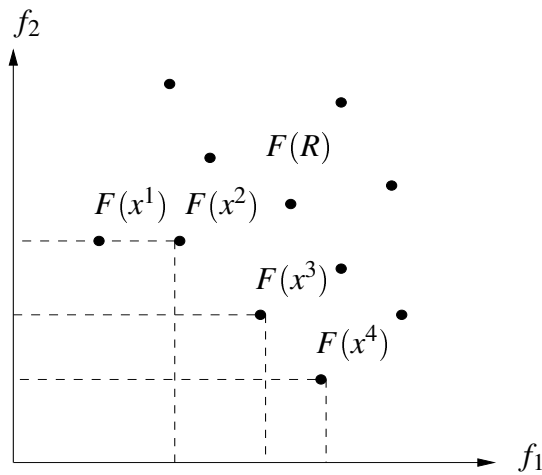
Definition 1.4.13. Eine Alternative $x \in R$ heißt **schwach (funktional)effizient**, falls kein $y \in R$ existiert mit

$$f_i(y) < f_i(x) \quad \forall \quad i = 1, \dots, n. \quad (1.12)$$

△

In der Regel ist nicht jeder schwach effiziente Punkt auch effizient.

Abbildung 1.3: schwache Effizienz und Effizienz: x^1, x^2, x^3, x^4 sind schwach (funktional)effizient, nur x^1, x^3, x^4 sind (funktional)effizient.

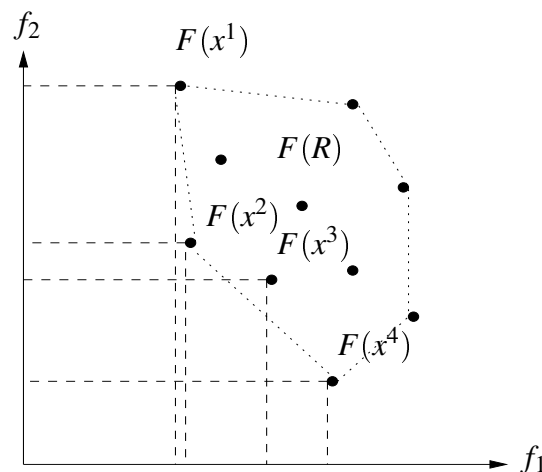


1.4.3 Wesentliche Effizienz

Definition 1.4.14. $\overline{F(R)}$ bezeichne die kleinste konvexe Menge, die $F(R)$ enthält. Diese wird **konvexe Hülle von $F(R)$** genannt. Eine Alternative $x \in R$ heißt **wesentlich (funktional)effizient**, falls $F(x)$ in $\overline{F(R)}$ effizient ist.

△

Abbildung 1.4: wesentliche Effizienz und Effizienz: x^1, x^2, x^3, x^4 sind (funktional)effizient, nur x^1, x^2, x^4 wesentlich (funktional)effizient.



Im Gegensatz zur Menge der schwach effizienten Punkte ist die Menge der wesentlich effizienten Punkte eine Teilmenge der Effizienzmenge. Letztere beiden Mengen sind offensichtlich im Falle der Konvexität von $F(R)$ identisch.

1.4.4 Eigentliche Effizienz

Hier wird lediglich Geoffrions Definition der eigentlichen Effizienz vorgestellt. Für die eigentliche Effizienz gibt es mehrere verschiedene Festlegungen, wie man z.B. in [6], [8], [29] oder [42] nachlesen kann. Die verschiedenen Auslegungen der eigentlichen Effizienz stimmen nur unter bestimmten Konvexitätsbedingungen für R und $F(R)$ überein.

Definition 1.4.15. Eine (funktional)effiziente Alternative $x \in R$ heißt **eigentlich (funktional)effizient nach Geoffrion**, falls es für jedes $k \in \{1, \dots, n\}$ und jede Alternative $y \in R$ mit $f_k(y) < f_k(x)$ möglich ist, einen Skalar $\alpha(x) > 0$ und mindestens ein $j \in \{1, \dots, n\}$ mit $f_j(x) < f_j(y)$ zu finden, die

$$\frac{f_k(x) - f_k(y)}{f_j(y) - f_j(x)} \leq \alpha(x) \quad (1.13)$$

erfüllen. △

Wann auch immer x in einem Kriterium $k \in \{1, \dots, n\}$ ein schlechteres Ergebnis als irgendeine Alternative y erzielt, muß dies in einem anderen Kriterium $j \in \{1, \dots, n\}$ durch eine ausreichend bessere Leistung als sie y dort bringt ausgeglichen werden.

Geber zeigt in [22], daß in einem Problem mit endlicher Alternativenmenge alle effizienten Punkte eigentlich effizient sind:

Mit $\alpha(x) := \frac{\alpha_1}{\alpha_2}$ zusammengesetzt aus

$$\alpha_1 := \max_{k \in \{1, \dots, n\}} \{ \max_{x \in R} f_k(x) - \min_{y \in R} f_k(y) \},$$

$$\alpha_2 := \min_{k \in \{1, \dots, n\}} \{ \min_{(x,y) \in R \times R, x \neq y} \{ |f_k(y) - f_k(x)| \}, f_k(y) \neq f_k(x) \}$$

ist es jedem paretooptimalen x möglich, die Ungleichung (1.13) für beliebiges $y \in R$ und beliebiges $k \in \{1, \dots, n\}$ mit $f_k(y) < f_k(x)$ zu erfüllen.

Um mit dem Filter der eigentlichen Effizienz die Effizienzmenge wirklich sinnvoll aus-sieben zu können, sollte der Entscheidungsträger je Paretooptimum $x \in R$ einen Skalar $\bar{\alpha}(x)$ festlegen, für den (1.13) einzuhalten ist. Diese Modifikation von Definition 1.4.15 führt zu dem Begriff **streng eigentlich effizient**:

Definition 1.4.16. Eine (funktional)effiziente Alternative $x \in R$ heißt **streng eigentlich effizient** bezüglich eines vorgegebenen Skalars $\bar{\alpha}(x) > 0$, falls es für jedes $k \in \{1, \dots, n\}$ und jede Alternative $y \in R$ mit $f_k(y) < f_k(x)$ möglich ist, mindestens ein $j \in \{1, \dots, n\}$ mit $f_j(x) < f_j(y)$ zu finden, so daß gilt:

$$\frac{f_k(x) - f_k(y)}{f_j(y) - f_j(x)} \leq \bar{\alpha}(x). \quad (1.14)$$

△

Kein Ziel wird im Vergleich zu anderen völlig vernachlässigt. Jeder Verlust in einem Ziel muß ausreichend durch einen Gewinn in mindestens einem anderen Ziel ausgeglichen werden. Der Kompensationsgrad wird durch $\bar{\alpha}(x)$ kontrolliert: je kleiner $\bar{\alpha}(x)$ ist, desto höher ist der geforderte Ausgleich.

Kapitel 2

Methoden der Vektoroptimierung

Um dem Leser einen knappen und zweckmäßigen Literaturüberblick zu verschaffen, gliedert dieses Kapitel die bekannten Methoden der Vektoroptimierung in Anlehnung an eine in der Literatur häufige Klassifikation, charakterisiert kurz die Grundideen der Klassen und stellt einige Vertreter jeder Klasse vor. Natürlich ist dieser Literaturüberblick nicht vollständig. Vielmehr soll er nahebringen, auf welche Problemgattungen der Mehrzieloptimierung sich bisher die Lösungsansätze konzentrieren, welcher Motivation sie entspringen, wo Lücken im Lösungsangebot liegen und warum für die Klasse von Mehrzieloptimierungsproblemen, die diese Arbeit betrachtet, noch eine neue Lösungstechnik zu entwickeln ist. Vor allem auf dem Gebiet der kontinuierlichen Mehrzieloptimierungstechniken könnten noch beliebig viele interessante Verfahren vorgestellt werden, wie z.B. Homotopiemethoden (Beispiel [30]), was jedoch für diese Arbeit nicht relevant ist. Nach jeder Skizzierung einer Vektoroptimierungsmethode werden deren Charakteristika hinsichtlich der Anwendbarkeit auf ein diskret-kontinuierliches Mehrzieloptimierungsproblem, wie es in Kapitel 3 formuliert wird, zusammengefaßt.

In [35],[11] und [44] richtet sich die Klassifikation der Mehrzieltechniken nach der Rolle des Entscheidungsträgers vor, während und nach der Anwendung des Verfahrens. Die kategorisierenden Aspekte sind laut [11] die Informationsflüsse während des Lösungsprozesses und die Hintergründe des Entscheidungsproblems. Die Informationsflüsse bestimmen die Rollen des Optimierers (Analysten) und des Entscheidungsträgers im Lösungsprozeß, die Hintergründe definieren die Zielsetzungen des Entscheidungsproblems. Cohon unterscheidet zwei Arten von Informationsflüssen:

- den Informationsfluß von Entscheidungsträger zu Analyst ("top-down"), der auftritt, wenn der Entscheidungsträger ausdrücklich seine Präferenzen in Bezug auf die Ziele äußert, um die von ihm unter allen Alternativen bevorzugte effiziente Alternative oder **Kompromißlösung** zu erhalten und
- den Informationsfluß von Analyst zu Entscheidungsträger ("bottom-up"), der Information über effiziente Alternativen, ihre Zielausprägungen und über Trade-Offs zwischen den Zielen übermittelt.

Methoden, die Präferenzen berücksichtigen, arbeiten mit ersteren Informationsflüssen, die interaktiven unter ihnen auch mit letzteren. Es kommt auch darauf an, wann und wie diese Präferenzen in den Lösungsvorgang mit eingebaut werden: a priori, interaktiv, implizit, explizit Die sogenannten Generierungstechniken enthalten die zweite Sorte von Informationsflüssen.

Bevor der Literaturüberblick beginnt, benötigen wir die Definition der Menge, die den in jede Richtung aus $\mathbb{R}_+^n \cup \{0\}$ unendlich geöffneten Zielraum $F(R)$ darstellt:

Definition 2.0.17.

$$F(R) + (\mathbb{R}_+^n \cup \{0\}) := \{(F(x) + v) \in \mathbb{R}^n \mid x \in R, v \in \mathbb{R}_+^n \cup \{0\}\}. \quad (2.1)$$

△

2.1 Methoden mit a priori Präferenzäußerung

Diese Methoden, auch **nicht-iterative präferenzorientierte Techniken** genannt, benötigen im Voraus Information über die Präferenzen des Entscheidungsträgers. Diese Information kann

- von kardinaler Form sein. Dazu zählen z.B. Höchstgrenzen für die Zielniveaus oder die Trade-Off Rate zwischen je zwei Zielen, d.h. wieviel Verlust in einem Kriterium für einen bestimmten Gewinn in einem anderen hingenommen wird.
- von ordinaler Form sein, z.B. eine Rangordnung der Ziele nach Priorität.

In jedem Fall ist die Absicht, zur Kompromißlösung zu kommen, ohne die gesamte Effizienzmenge $\subseteq R$ oder repräsentative Teile davon erarbeiten zu müssen. Die Präferenzinformation leitet die Algorithmen in bestimmte Gebiete der Alternativenmenge; Handlungsalternativen außerhalb dieser Gebiete werden nicht beachtet. Nimmt man an, daß a priori Präferenzaussagen bei mangelndem Verständnis des Entscheidungsproblems geäußert werden, was ohne Kenntnis aller sinnvollen Handlungsalternativen (nämlich der Paretooptima) nicht weiter verwunderlich ist, könnten sie den Lösungsprozeß irreführen.

Zu der Gefahr der Voreiligkeit kommt außerdem, daß der Entscheidungsträger in manchen Fällen zögern wird, seine Trade-Off-Wünsche zum Ausdruck zu bringen, speziell, wenn einige Ziele ethischer oder moralischer Natur sind.

2.1.1 Nutzenfunktionsmethoden

(Multiattributive) Nutzenfunktionen $U : \mathbb{R}^n \mapsto \mathbb{R}, z \mapsto U(z)$, sind Funktionen, die jedem $z \in \mathbb{R}^n$ einen Nutzenwert zuordnen, insbesondere jeder zulässigen Kombination von Zielausprägungen $z \in F(R)$ und somit auch jeder Alternative $x \in R, F(x) = z$. Anhand des Nutzenwertes lassen sich die Alternativen vollständig ordnen. Nutzenfunktionen vereinfachen n Ziele zu einem Ziel, nämlich maximalen Nutzen zu erreichen:

$$\max_{x \in R} U(f_1(x), \dots, f_n(x)) = \max_{x \in R} U(F(x)). \quad (2.2)$$

$U(F(\bar{x}))$ quantifiziert den Nutzen, den der Entscheidungsträger aus der Wahl der Alternative $\bar{x} \in R$ gewinnt. Der Wert drückt den Nutzen der Kombination von Zielausprägungen/Zielniveaus der Alternative \bar{x} in Form einer reellen Zahl aus.

Wenn die Zielfunktionen zu minimieren sind, setzt man U als streng monoton fallend in jedem Ziel voraus. Das Maximieren von $U \circ F$ über R führt dann zu effizienten Punkten.

Meist wird obendrein die Konkavität der Funktion $U \circ F$ auf entsprechendem R gefordert, damit deterministische kontinuierliche Optimierungsverfahren auch wirklich die globalen Maximierer von $U \circ F$ in R erreichen und nicht in lokalen Maxima hängenbleiben.

Alle Zielwertkombinationen mit dem gleichen Nutzenniveau $U(F(\bar{x}))$ formen eine **Indifferenzkurve** - sie und mit ihnen die sie hervorbringenden Alternativen sind für den Entscheidungsträger gleichwichtig.

Alle Alternativen mit Zielwertvektoren auf der Indifferenzkurve des höchsten Nutzenniveaus, die $F(R)$ tangiert, sind Kompromißlösungen - effiziente Alternativen mit dem höchstmöglichen Nutzenwert. Alle Maxima von $U(F(x))$ sind garantiert paretooptimal, falls U in jedem einzelnen Ziel streng monoton fallend ist.

Das Hauptmanko der Nutzenfunktionen als Ausdrucksform der Präferenzen des Entscheidungsträgers liegt in den großen Schwierigkeiten, sie für ein gegebenes Mehrzielproblem passend zu konstruieren. Der Entscheidungsträger muß sich über seine Präferenzen im klaren sein, kann dies jedoch bei seinem Informationsstand vor der Problemlösung kaum. Ein Individuum, das sich all seiner Handlungsmöglichkeiten bewußt ist und obendrein seinen Grad der Zufriedenheit mit ihnen bewerten kann, ist selten. Darüber hinaus ist nicht gewährleistet, ob die Nutzenfunktion, wenn man das Glück hatte, eine zu finden, die bereits erwähnten mathematischen Eigenschaften besitzt. Dies spielt auf die häufig geforderte Konkavität von $U \circ F$ auf passendem R an.

Eine sehr beliebte Nutzenfunktion ist linear; es ist die negative **Summierung der gewichteten Zielfunktionen** $S_\alpha : \mathbb{R}^n \mapsto \mathbb{R}, z \mapsto \sum_{i=1}^n \alpha_i z_i$:

$$U := -S_\alpha,$$

$$\max_{x \in R} U(F(x)) = \min_{x \in R} \sum_{i=1}^n \alpha_i f_i(x), \quad (2.3)$$

$$\sum_{i=1}^n \alpha_i = 1, \alpha_i > 0, i = 1, \dots, n.$$

Die Gewichte $\alpha_i, i = 1, \dots, n$, verleihen jedem Ziel seinen Wichtigkeitsgrad. Sie stellen auch die marginale Substitutionsrate $-\frac{\alpha_i}{\alpha_r} = \frac{\delta f_r}{\delta f_i}, i = 1, \dots, n, r \in \{1, \dots, n\}$, dar, die angibt, wieviel Erhöhung $\delta f_i > 0$ des Zielwertes von f_i der Entscheidungsträger für eine Absenkung $\delta f_r < 0$ des Zielwertes einer Referenzielfunktion f_r akzeptiert. Aufgrund der Linearität der Nutzenfunktion bleibt der marginale Nutzen $-|\delta f_r|$, der aus Opferung von $+|\delta f_i|$ entsteht, auf jedem Niveau der Ziele i konstant. Das Lösen von (2.3) ermittelt wesentlich effiziente Punkte. Falls bei einigen $i \in \{1, \dots, n\}$ der Wert 0 für α_i zugelassen ist, können unter den Ergebnissen von (2.3) auch schwach effiziente Alternativen sein.

Zusammenfassung:

Vorteile:

- Unter der Annahme, daß es ein U gibt, das den Nutzen aller $x \in R$ sinnvoll einschätzt, indem es jedem $z \in F(R)$ einen Nutzenwert zuweist, ist die Idee, nach bevorzugten Alternativen (diejenigen mit höchstem Nutzenwert) zu suchen, plausibel.

Nachteile:

- Brauchbare Gewichte α oder brauchbare Nutzenfunktionen im Allgemeinen sind sehr schwer zu bestimmen.
- Ist $F(R) + (\mathbb{R}_+^n \cup \{0\})$ nicht konvex, so sind einige Paretooptima mit keiner der möglichen Zielgewichtungen in $U := -S_\alpha$ erreichbar, nämlich jene, die im nichtkonvexen Teil von $F(R) + (\mathbb{R}_+^n \cup \{0\})$ liegen, d.h. die nicht wesentlich effizienten Alternativen. Das bedeutet, daß zu der Gefahr einer ungeschickten Wahl von U und somit der ungeschickten Präferenzberücksichtigung noch hinzukommt, daß dem Entscheidungsträger lediglich aus mathematischen Gründen die Kenntnis interessanter Handlungsalternativen vorenthalten wird.

Die Literatur über Nutzenfunktionsmethoden und die Probleme bei der Bestimmung von U werden in [17], [20], [33],[38], [39] und [67] zusammengefaßt und diskutiert.

2.1.2 Methode der beschränkten Ziele

Wieder wird das Mehrzielproblem durch ein skalares Ersatzproblem ersetzt. Nur eine Zielfunktion $f_r, r \in \{1, \dots, n\}$, wird minimiert, die übrigen $n - 1$ werden, indem man sie beschränkt, in Nebenbedingungsfunktionen umgewandelt, wie z.B. in der **Methode der ε -Schranken** (dies ist auch der Name für eine weitere Vektoroptimierungsmethode, die in Abschnitt 2.3.1.2 beschrieben wird):

$$\min_{x \in R} f_r(x), f_i(x) \leq \varepsilon_i, \varepsilon_i \in \mathbb{R}, i = 1, \dots, n, i \neq r. \quad (2.4)$$

Falls die Lösung von (2.4) eindeutig ist, dürfen wir annehmen, daß sie effizient ist, ansonsten kann nur schwache Effizienz für die Lösungen garantiert werden. Eine Alternative, die (2.4) für eine Schrankenkombination $(\varepsilon_1, \dots, \varepsilon_n) \in \mathbb{R}^n$ für alle $r \in \{1, \dots, n\}$ löst, ist paretooptimal. Zu jeder effizienten Alternative existiert ein Schrankenvektor $(\varepsilon_1, \dots, \varepsilon_n) \in \mathbb{R}^n$, so daß die Alternative die zugehörigen Probleme (2.4) für alle $r \in \{1, \dots, n\}$ minimiert. Folglich sind alle Paretooptima durch Minimieren einer Referenzzielfunktion mit irgendeinem $\varepsilon \in \mathbb{R}^{n-1}$, das die restlichen Zielfunktionen beschränkt, erreichbar.

f_r wird vom Entscheidungsträger am meisten favorisiert und seine die anderen Ziele betreffenden Wünsche kommen in den Oberschranken $\varepsilon_i \in \mathbb{R}, r \neq i = 1, \dots, n$, als eine Art Mindestanforderung an deren Leistungsniveau zum Ausdruck.

Zusammenfassung:

Vorteile:

- Der Gedanke, sich auf die Minimierung einer Zielfunktion bei Vorgabe von Mindestanforderungen an die restlichen Zielfunktionen zu konzentrieren, ist für einige Probleme sicherlich sinnvoll.
- Wie auch immer R und $F(R)$ beschaffen sind - sofern eine skalare Optimierungsmethode zur Hand ist, die jedes globale Minimum von Problem (2.4) erreichen kann, ist jedes Paretooptimum bei geeigneter Wahl von $\varepsilon \in \mathbb{R}^{n-1}$ auffindbar.

Nachteile:

- Ist es wirklich sinnvoll, ausgerechnet Ziel r gegenüber allen anderen Zielen zu priorisieren?
- Setzt man die ε -Grenzen zu niedrig an, könnte der zulässige Bereich des skalaren Ersatzproblems plötzlich leer sein; setzt man sie zu hoch an, werden die Ziele $i = 1, \dots, n, i \neq r$, mehr oder weniger vernachlässigt und dem Ziel r kommt übersteigerte Bedeutung zu.
- Wie kann man ohne viel Arbeitsaufwand unter den Lösungsalternativen von (2.4) die effizienten Alternativen identifizieren?

2.1.3 Lexikographische Methode

Hier ordnet man die Ziele zunächst entsprechend ihrer Wichtigkeit für den Entscheidungsträger. Mit abnehmender Priorität steige der Index. Für $i = 1$ bis $i = n$ sind die Probleme P_i zu lösen, deren Lösungen die Mengen S_i definieren:

$$\begin{aligned} (P_1) \quad & \min_{x \in R} f_1(x), \\ (P_i) \quad & \min_{x \in S_{i-1}} f_i(x), i = 2, \dots, n. \end{aligned} \tag{2.5}$$

Ein S_i mit $|S_i| > 1, i = 1, \dots, n - 1$, enthält schwach effiziente Lösungen und mindestens ein Paretooptimum. Im Falle $i < n$ und $|S_i| = 1$ ist das Ergebnis sicher effizient, berücksichtigt jedoch nicht die Ziele $j = i + 1, \dots, n$; der Lösungsprozeß ist dann vorzeitig abgeschlossen. Ist die Bearbeitung von P_n noch möglich und nötig, so enthält S_n nichtdominierte Alternativen, die jedes Ziel gemäß seinem Rang berücksichtigen.

Zusammenfassung:

Vorteile:

- Unter den Zielen eine Rangordnung einzuführen, ist ein plausibler Versuch, die Präferenzen des Entscheidungsträgers zum Ausdruck zu bringen.

Nachteile:

- Leider hat man keine Chance, Paretooptima, die sich in keinem der Ziele $k \in \{1, \dots, n\}$ besonders auszeichnen, bei einem der Rangsznarien zu finden.
- Es ist schwierig, auf eine Rangordnung zu kommen, die wirklich den Präferenzvorstellungen des Entscheidungsträgers entspricht.
- Im Fall $|S_i| = 1, 1 \leq i < n$, bekommen die weniger wichtigen Ziele $j = i + 1, \dots, n$ nicht die Möglichkeit, das Optimierungsergebnis zu beeinflussen.

Lexikographisches Ordnen wird z.B. in [15] angewendet.

2.1.4 Optimierung mit Zielvorgaben

Die Mehrzielmethode, die sich hinter dem Namen Optimierung mit Zielvorgaben verbirgt, wurde von Charnes und Cooper im Rahmen eines linearen Optimierungsmodells vorgestellt. Andere Abhandlungen über Optimierung mit Zielvorgaben wurden von Ijiri ([37]), Lee ([43]) und Ignizio ([36]) veröffentlicht. Eine Übersicht findet sich in [10].

Der Entscheidungsträger gibt für jedes Ziel Zielvorgaben, die er zu erreichen wünscht. Eine Kompromißlösung ist in diesem Fall eine Alternative, deren Zielwerte minimal von den Zielvorgaben abweichen. Eine einfache Formulierung des Optimierungsproblems mit Zielvorgaben ist für festes $p \geq 1$ gegeben durch

$$\min_{x \in R} \sqrt[p]{\sum_{i=1}^n (d_i^- + d_i^+)^p}, \quad (2.6)$$

wobei $d_i^+ := (f_i(x) - g_i)^+$, $d_i^- := (g_i - f_i(x))^+$, $i = 1, \dots, n$.

g_i sind die angestrebten Leistungsniveaus (Zielvorgaben) für Ziel i , $i = 1, \dots, n$, d_i^+ gibt an, inwieweit man schlechter als die Zielvorgabe liegt und d_i^- , wie weit man über die Zielvorgabe hinausgeschossen ist (aus Sicht eines Minimierungsproblems).

$$\min_{x \in R} \sqrt[p]{\sum_{i=1}^n pr_i (w_i^- d_i^- + w_i^+ d_i^+)^p}, \quad (2.7)$$

$$\sum_{i=1}^n pr_i = 1, pr_i, w_i^-, w_i^+ \geq 0, i = 1, \dots, n.$$

Wie in (2.7) demonstriert, kann der Entscheidungsträger Gewichte $w_i^- \geq 0$ und $w_i^+ \geq 0$ einführen und somit anzeigen, ob er auch die Zielvorgabe übertreffende Leistungsniveaus in i ($w_i^- = 0, i \in \{1, \dots, n\}$) akzeptiert oder nur zufrieden ist, wenn die Zielvorgabe exakt eingehalten wird ($w_i^- > 0, w_i^+ > 0$), ob er gegebenenfalls Überbietung der Zielvorgabe der Unterbietung vorzieht ($w_i^+ > w_i^-$). Außerdem können die Ziele durch Beifügung von Prioritätsfaktoren $pr_i \geq 0, i = 1, \dots, n$, mit $\sum_{i=1}^n pr_i = 1$ in eine Rangordnung gebracht werden.

Eine Lösung des Minimierungsproblems

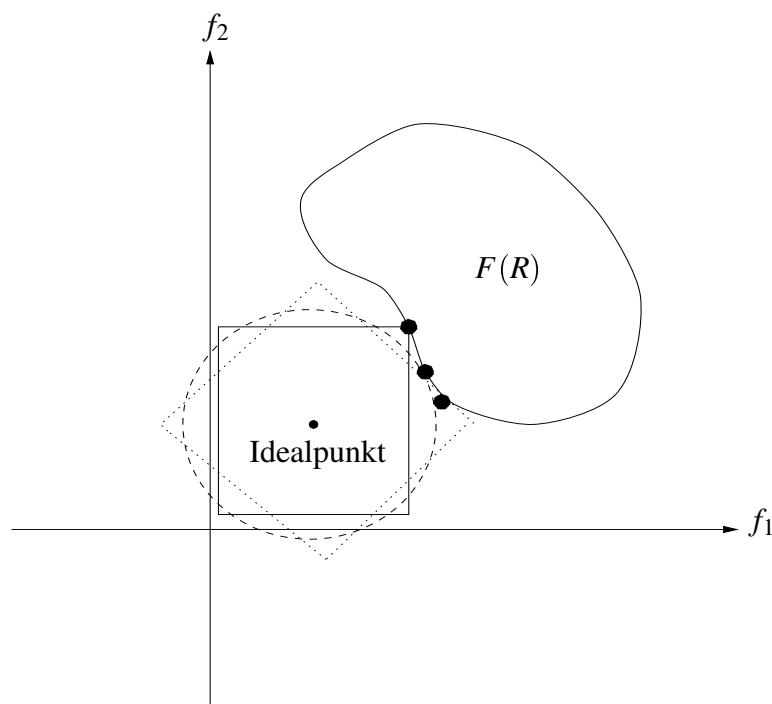
$$f_i^* := \min_{x \in R} f_i(x), i \in \{1, \dots, n\}, \quad (2.8)$$

sei $x_i^* \in R$. Für $g_i = f_i^* := f_i(x_i^*), i = 1, \dots, n$, ist die Strategie, Punkte des Suchraumes zu finden, die dem sogenannten **Idealpunkt** (f_1^*, \dots, f_n^*) "am nächsten" liegen, natürlich vorausgesetzt, dieser Idealpunkt existiert. Diese Vorgehensweise wird als **Methode des globalen Kriteriums** bezeichnet. Nachzulesen ist sie z.B. in [68], [70] und [71].

Verschiedene p -Werte führen zu verschiedenen Lösungen, wie Abbildung 2.1 illustriert, sie spiegeln aber in keiner Weise eine spezielle Problemsituation wider. Großes p hebt große Abweichungen $|f_i(x) - g_i|, i = 1, \dots, n$, von der Zielvorgabe hervor. Bei $p = \infty$ können Lösungen herauskommen, die nur schwach effizient sind, bei $1 \leq p < \infty$ sind alle Ergebnisse paretooptimal, vorausgesetzt $(g_1, \dots, g_n) \leq F(x) \quad \forall x \in R$ und $pr_i > 0, w_i^+ > 0 \quad \forall i = 1, \dots, n$.

Zur Präferenzeinbindung in der Methode der Optimierung mit Zielvorgaben stellen wir allgemein fest: Bei einem Minimierungsproblem ist ein Ziel umso bedeutender, je niedriger der

Abbildung 2.1: Methode des globalen Kriteriums: Verschiedene Metriken, z.B. $L_1(p = 1)$, $L_2(p = 2)$ und $L_\infty(p = \infty)$, führen zu unterschiedlichen effizienten Punkten.



Wert der Zielvorgabe angesetzt ist, wohingegen hohe Prioritätsfaktoren betonen, wie wichtig es ist, die Zielvorgabe zu erreichen (oder gar zu übertreffen).

Zusammenfassung:

Vorteile:

- Kann der Entscheidungsträger seine Wünsche in Form von Zielvorgaben ausdrücken, so macht es durchaus Sinn, nach Alternativen zu suchen, deren Leistung den Zielvorgaben möglichst nahe kommt oder diese sogar soviel wie möglich übertrifft.
- Mit einer skalaren Optimierungsmethode als Werkzeug, die alle globalen Minima eines Problems (2.7) erreichen kann, sind bei geschickter Wahl von Zielvorgaben, Gewichten und Prioritätsfaktoren alle effizienten Alternativen greifbar, egal wie R und $F(R)$ aussehen.

Nachteile:

- Sind die Zielvorgaben sinnvoll? $(g_1, \dots, g_n) \in \text{int}(F(R)) := F(R) \setminus \delta(F(R))$ z.B. wird uns keine effizienten Alternativen verschaffen. Welche Zielvorgaben führen dann zu (schwach) effizienten Punkten?
- Ist die Rangordnung unter den Zielen, die durch die Wahl der Zielvorgaben indirekt eingeführt wird, wirklich gerechtfertigt?
- Wie wählt man überhaupt gute Gewichte und/oder Prioritätsfaktoren?

2.2 Methoden mit fortschreitender Präferenzäußerung

In diesem Zusammenhang ist der Begriff **Trade-Off** wichtig:

Definition 2.2.1. Es seien $x, y \in R$ zwei Alternativen. Die Austauschrate zwischen zwei Zielen $f_i, f_j, i, j \in \{1, \dots, n\}$, beim Übergang von x zu y ist

$$\hat{\lambda}_{ij}(x, y) := \frac{f_i(x) - f_i(y)}{f_j(x) - f_j(y)}, f_j(x) - f_j(y) \neq 0. \quad (2.9)$$

$\hat{\lambda}_{ij}(x, y)$ heißt **partieller Trade-Off** zwischen f_i und f_j beim Tausch von Alternative x gegen y , falls $f_k(x) = f_k(y) \quad \forall k \neq i, j$. Ansonsten heißt $\hat{\lambda}_{ij}$ **vollständiger Trade-Off**. △

Methoden mit fortschreitender Präferenzäußerung, häufig einfach **interaktive Methoden** oder auch **iterative präferenzorientierte Techniken** genannt, bauen auf die mit der Erforschung des Suchraumes fortschreitende Äußerung von Präferenzen. In jeder Iteration betrachtet der Entscheidungsträger die momentane Lösung oder Lösungsmenge und beantwortet daraufhin Fragen hinsichtlich gewünschter Trade-Offs. Lokal gültige Präferenzaussagen werden also mit Blick auf lokale Information über die Effizienzmenge gemacht. Auf diese Weise bewegt sich die Suchprozedur geleitet von den Antworten des Entscheidungsträgers von einem Paretooptimum zum nächsten. Einige Verfahren benötigen explizite Trade-Off Information, andere Verfahren arbeiten mit impliziter Information von der Art "akzeptabel" oder "nicht akzeptabel". Der Informationsfluß vom Entscheidungsträger soll die Verfahren auf neue und bessere Lösungen (Lösungsmengen) hinsteuern, die wiederum Reaktionen von ihm auslösen. Dieses Wechselspiel währt, bis der Entscheidungsträger mit einem Ergebnis zufrieden ist oder ein maximaler Arbeitsaufwand überschritten wird.

Hinter diesen Methoden steckt die Idee, daß die Komplexität des gegebenen Entscheidungsproblems es dem Entscheidungsträger unmöglich machen kann, im Vorhinein seine Präferenzen festzulegen und zum Ausdruck zu bringen. Im Verlauf des Lösungsprozesses lernt der Entscheidungsträger mehr und mehr über das Problem dazu und man traut ihm daher zumindest lokal gültige seriöse Wunschäußerungen zu. Das klingt sehr vernünftig, man darf sich jedoch nicht über gewisse Nachteile dieser Sichtweise und ihrer Umsetzung hinwegtäuschen lassen:

- Welche Lösungen gefunden werden, hängt nach wie vor stark von der Verlässlichkeit der Aussagen des Entscheidungsträgers ab. Widerspricht etwa die Information in einem Dialog der Information in einem anderen Dialog, so führt das im Suchvorgang zu Inkonsistenz.
- Obwohl er im Laufe des Lösungsvorganges dazulernt, könnte dem Entscheidungsträger dennoch das ausreichende Verständnis für die Problemstellung fehlen, und es könnten interessante Lösungen unentdeckt bleiben, weil er das Verfahren irreleitet.
- Viele Methoden können nicht garantieren, daß die bevorzugte Alternative in endlich vielen Schritten erreicht werden kann.
- Eventuell ist dem Entscheidungsträger die Arbeit im Dialog mit dem Mehrzielverfahren zu zeitaufwendig.

Einige Methoden erwarten in jeder interaktiven Iteration explizite Trade-Off Information; sie heißen deshalb **Methoden mit expliziter Trade-Off Information**. Andere wiederum liefern eine Menge von Lösungsvorschlägen, aus denen der Entscheidungsträger diejenigen, die ihm am besten erscheinen, auswählen kann. Indem er sie durchdenkt, belegt er die Suche steuernde Parameter mit neuen Werten oder bestimmt, in welchen Zielen er seine Zielvorstellungen abändert. Die gängige Literatur nennt zwei Vorteile dieser **Methoden mit impliziter Trade-Off Information** gegenüber den Methoden mit expliziter Trade-Off Information:

- Dem Entscheidungsträger ist eher zuzutrauen, daß er vernünftige Vorgaben für ein gewünschtes Zielniveau machen oder seinen Favoriten aus einer Menge von Lösungsvorschlägen ermitteln kann, als daß er sinnvolle Trade-Off Wünsche formulieren kann.
- Die Trade-Off Information, mit der der Entscheidungsträger in Methoden mit expliziter Trade-Off Information konfrontiert wird, ist nur lokal gültig. Ist sich der Entscheidungsträger dessen nicht bewußt, könnten seine Trade-Off Wünsche unpassend sein und den Algorithmus in die falsche Richtung führen. Die Angabe impliziter Trade-Off Äußerungen vermeidet diese Gefahr.

Zu den Gefahren, die in der Philosophie der interaktiven Methoden liegen, kommt noch der weitere Haken, daß unter ihnen die meisten Verfahren nur Probleme mit sehr einschränkenden Eigenschaften lösen können, hinzu. Außerdem sind in jeder Rechenphase zwischen zwei Frage-Antwort Austauschschritten skalare Ersatzprobleme des gegebenen Vektoroptimierungsproblems zu lösen, deren Skalierungskonstrukte bereits aus Abschnitt 2.1 bekannt sind. Daher beschränken wir uns lediglich auf eine Aufzählung der bekanntesten Methoden, ohne diese im Detail zu erklären.

2.2.1 Methoden mit expliziter Trade-Off Information

Die **Methode von Geoffrion**, die Geoffrion, Dyer und Feinberg in [18] entwickeln, geht von der Existenz einer Nutzenfunktion $U : F(R) \mapsto \mathbb{R}, z \mapsto U(z)$, aus. Sie soll maximiert werden, ohne daß jedoch die explizite Form von U bekannt wäre. Von Iteration zu Iteration leitet man die Suchrichtungen aus lokalen Trade-Off Aussagen des Entscheidungsträgers ab.

Auch die **Ersatzwert Trade-Off Methode**, ein von Haimes, Hall und Freedman in [27] und [26] vorgeschlagenes Verfahren, basiert auf dem Gedanken, daß der Entscheidungsträger eine Nutzenfunktion implizit im Kopf hat. In jeder Iteration wird ein ε -Schrankenproblem der Form (2.4) gelöst, was lokale Trade-Off Information zutage bringt. Geleitet von seiner Vorstellung einer Nutzenfunktion verarbeitet der Entscheidungsträger diese Information zu einem neuen ε -Schrankenproblem, das in der nächsten Iteration zu bearbeiten ist. Diese Handlungsfolge endet, wenn der Entscheidungsträger mit einer gefundenen Lösung zufrieden ist.

Die **Methode der zufriedenstellenden Zielniveaus**, die Benson in [7] vorstellt, wendet in jeder Iteration die Methode der ε -Schranken für ein $\varepsilon \in \mathbb{R}^n$ an, um eine Lösungsalternative hervorzubringen. Der Entscheidungsträger betrachtet die Lösung und bestimmt das Kriterium, das für seinen Geschmack in ihr am schlechtesten ausgeprägt ist, zum Referenzziel der nächsten Iteration; außerdem legt er (neue) Maximalniveaus für alle restlichen Zielwerte fest

(meist weniger strenge als in der Voriteration). Auf diese Weise entsteht das neue Problem beschränkter Ziele für die nächste Iteration. Ist eine erarbeitete Lösung dem Entscheidungsträger gut genug, wird auf weiteres Suchen verzichtet.

Die Motivation hinter der **Methode von Zionts-Wallenius**, die auf Zionts und Wallenius (siehe [65] und [72]) zurückgeht, ist erneut die Minimierung einer implizit bekannten Nutzenfunktion. Man nimmt an, daß diese Nutzenfunktion eine negative Summe gewichteter Zielfunktionen sei. Dem Entscheidungsträger wird die Lösung der jeweiligen Iteration, entstanden aus der Minimierung einer gewichteten Zielfunktionssumme, präsentiert, und er wird daraufhin um die Angabe lokaler Trade-Off Wünsche gebeten. Aus diesen entstehen die Gewichte der gewichteten Zielfunktionssumme der nächsten Iteration. Die Suche endet, sobald der Entscheidungsträger in einer gewichteten Zielfunktionssumme die richtige Nutzenfunktion erkennt.

2.2.2 Methoden mit impliziter Trade-Off Information

Der Name **STEP-Methode** oder **STEM** steht für mehrere miteinander verwandte Verfahren, die Benayoun, Larichev, de Montgolfier, Tergny und Keuneman in [4], [3] und [5] erklären. Hier wird nicht angenommen, daß eine Nutzenfunktion existiert. In jeder Iteration sorgt eine modifizierte Methode des globalen Kriteriums mit einer Art L_∞ -Metrik und Zielabweichungsgewichtungen für die Bereitstellung einer Lösungsalternative. Der Entscheidungsträger wählt das Kriterium mit dem befriedigendsten Leistungsniveau. Die dazugehörige Differenz aus Zielwert der Lösung und Wert der Komponente des Idealpunktes erhält den Gewichtungsfaktor 0 und wird in der skalarisierten Zielfunktion der folgenden Iteration nicht mehr berücksichtigt. Die Zielfunktion des Kriteriums verbleibt als Nebenbedingung im Problem der nächsten Iteration - der Entscheidungsträger legt ein maximales Wertenniveau fest, das evtl. etwas über dem in der Voriteration erreichten Niveau liegt. Mit Auffinden einer den Entscheidungsträger befriedigenden Alternative endet das Verfahren.

Die **Interaktive Gewichtete Tchebycheff Prozedur** ist eine Idee von Steuer und Choo ([58]) und Steuer ([56], [57]). Nicht nur eines, sondern mehrere Probleme des globalen Kriteriums werden je Iteration bearbeitet. Sie unterscheiden sich in der Gewichtung der Zielabweichungen. Diese Gewichtungen stellen verschiedene Präferenzszenarien dar. Die Lösungen, die die verschiedenen Prozeduren des globalen Kriteriums hervorbringen, werden dem Entscheidungsträger gezeigt, der daraus seinen Favoriten erwählt. Demnach ist der Gewichtsvektor, der zum skalaren Ersatzproblem gehört, das diesen Favoriten hervorbrachte, als bisher bester Ausdruck der Präferenzen des Entscheidungsträgers zu erachten. Als Konsequenz daraus wählt man die Gewichtsvektoren für die Probleme des globalen Kriteriums der Folgeiteration aus der näheren Umgebung dieses Gewichtsvektors in \mathbb{R}^n . Diese Vorgänge werden wiederholt, bis dem Entscheidungsträger eine Lösung gut genug ist.

In der **Methode von Steuer** ([59] und [60]) ersetzt die Methode der Summierung der gewichteten Zielfunktionen die Methode des Globalen Kriteriums in der eben beschriebenen Vorgehensweise. Die ersten Gewichtsvektoren werden unter Gleichverteilung aus dem Inneren des n -dimensionalen Einheits-Hyperwürfels gewählt.

Die **Interaktive ε -Gitter Methode** von V.V. Merkurev und M.A. Moldavskii (siehe [44]) stellt eine Verallgemeinerung der Strategie von Steuer dar. Sie verwenden verschiedene Arten der Skalarisierung der Mehrzielfunktion, die jeweils von bestimmten Parametervektoren abhängen. In der ersten Iteration wird jede skalarisierte Zielfunktion für bestimmte Wertebelegungen der Parametervektoren über R minimiert. Diese Wertebelegungen sind so geschickt zu wählen, daß effiziente Punkte aus R ziemlich gut verteilt über die Effizienzmenge erzeugt werden. Konfrontiert mit den Lösungsalternativen dieser Iteration bestimmt der Entscheidungsträger eine oder mehrere Wertebelegungen der Parametervektoren, aus deren näheren Umgebung weitere Wertebelegungen für die Parametervektoren gewählt werden, so daß für die nächste Iteration skalare Ersatzprobleme definiert werden, deren Lösungen Zielwertvektoren in den vom Entscheidungsträger bevorzugten Gebieten von $F(R)$ liegen. Mit der Iteration, in der der Entscheidungsträger mit den hervorgebrachten Paretooptima zufrieden ist, endet die Suche.

2.3 Generierungstechniken / Methoden mit a posteriori Präferenzäußerung

Generierungstechniken konzentrieren sich darauf, einen möglichst guten Eindruck vom gegebenen Mehrzieloptimierungsproblem zu bieten, indem sie möglichst viel Information über das Mehrzieloptimierungsproblem erarbeiten. Dadurch, daß sie offenlegen, was möglich ist und was für welchen Gewinn geopfert werden muß, zielen sie auf gutinformierte Entscheidungsfindung ab und entlasten damit den Entscheidungsträger.

Alle Methoden dieser Klasse erarbeiten eine Approximation einer Teilmenge aller Paretooptima eines Vektoroptimierungsproblems. Ziel ist, alle Paretooptima zu erlangen bzw. eine repräsentative Teilmenge der Effizienzmenge in R . Der Entscheidungsträger kann aus den von den Verfahren produzierten Ergebnismengen seine favorisierte effiziente Alternative auswählen. Der Einblick in die Approximation der gesamten Effizienzmenge bzw. einer repräsentativen Teilmenge der Effizienzmenge informiert den Entscheidungsträger bestmöglich über das zu lösende Mehrzielproblem, da die Spannweite der Zielausprägungen und die Trade-Offs zwischen den Zielen sichtbar werden.

Darüber hinaus ist der Entscheidungsträger nicht gezwungen, vor Ende der Suchprozedur Präferenzinformationen zu geben.

Einige Autoren der Mehrzieloptimierungsliteratur befürchten, daß die Präsentation so vieler Paretooptima wie möglich den Entscheidungsträger überfordern könnte und bevorzugen daher Methoden mit a priori Präferenzäußerung oder interaktive Methoden, in der Annahme, diese könnten dem Entscheidungsträger seine favorisierte Lösung liefern. Das ist Meinungssache und in der vorliegenden Arbeit wird diese Ansicht nicht geteilt.

2.3.1 Deterministisches Suchen

2.3.1.1 Wichtungsmethode (parametrische Methode)

Zadeh ([69]) empfahl als erster die Minimierung verschiedener gewichteter Zielfunktionssummen, um zu einer Approximation der Effizienzmenge zu kommen.

Die notwendigen Kuhn-Tucker Bedingungen (siehe [40]) für Paretooptima \hat{x} in einem Problem mit stetig differenzierbaren Ziel- und Nebenbedingungsfunktionen weisen bereits auf die

Möglichkeit, derart effiziente Punkte zu finden, hin:

$$\begin{aligned}
 -\sum_{i=1}^n w_i \nabla f_i(\hat{x}) &= \sum_{i=1}^p \mu_i \nabla g_i(\hat{x}), \\
 w_i &\geq 0, i = 1, \dots, n, \\
 \mu_k &\geq 0, k = 1, \dots, p, p \in \mathbb{N}_0, \\
 \mu_k g_k(\hat{x}) &= 0, k = 1, \dots, p, \\
 g_k : \mathbb{R}^m &\mapsto \mathbb{R}, x \mapsto g_k(x), k = 1, \dots, p, \quad \text{Nebenbedingungsfunktionen,} \\
 R &:= \{x \in \mathbb{R}^n \mid g_k(x) \leq 0, k = 1, \dots, p\}.
 \end{aligned} \tag{2.10}$$

Eine Teilmenge der Effizienzmenge wird erzeugt durch die Lösung mehrerer skalarer Ersatzprobleme der Form

$$\begin{aligned}
 \min_{x \in R} \sum_{i=1}^n w_i f_i(x), \\
 \sum_{i=1}^n w_i = 1, w_i > 0 \quad \forall \quad i = 1, \dots, n.
 \end{aligned} \tag{2.11}$$

Ist $w_i = 0$ für einige $i \in \{1, \dots, n\}$ zugelassen, können bei der Lösung von (2.11) auch schwach effiziente Punkte herauskommen, aber unter ihnen ist zumindest eine paretooptimale Alternative. Demnach ist eine eindeutige Lösung von (2.11) mit zugelassenen Nullgewichten paretooptimal.

Die notwendigen Kuhn-Tucker Bedingungen für Minima von (2.11) sind identisch mit denen für effiziente Punkte von (2.10). Interpretiert man die negative Summierung gewichteter Zielfunktionen mit bestimmten Gewichten als spezielle Nutzenfunktion U , kann man die Motivation für die Wichtungsmethode auch im Durchspielen verschiedener Nutzenszenarien bzw. Präferenzszenarien sehen (siehe Abschnitt 2.1.1).

Die Schwierigkeit besteht nun darin, die Gewichte so geschickt zu wählen, daß die aufgefundenen Paretooptima möglichst gut über die Effizienzmenge verteilt liegen. Das und Dennis beschäftigen sich in [14] mit diesem Thema.

Falls $F(R) + (\mathbb{R}_+^n \cup \{0\})$ nicht konvex ist, kann man mit der Wichtungsmethode nicht jede effiziente Alternative erreichen, was die Illusion, eine repräsentative Teilmenge der Effizienzmenge erarbeiten zu können, zerstört.

Zusammenfassung:

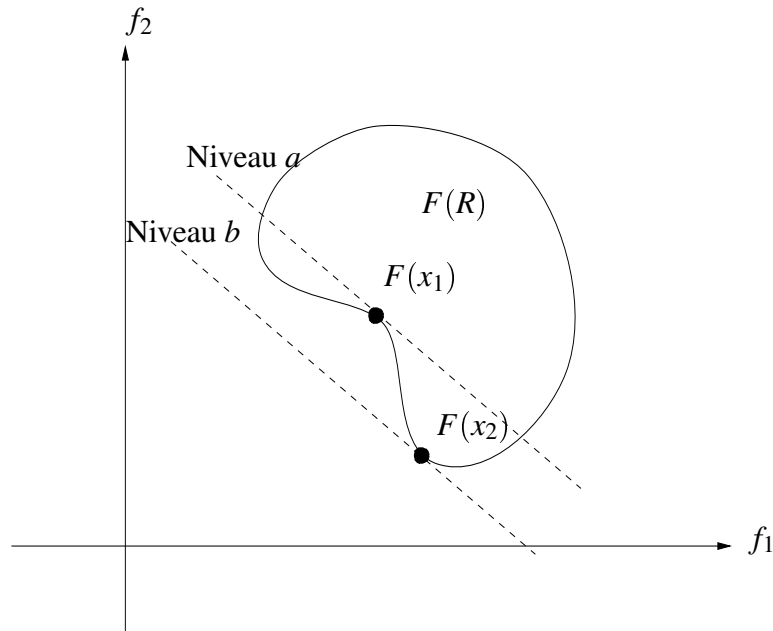
Vorteile:

- Die Motivation hinter der Wichtungsmethode läßt sich sowohl von mathematischen Optimalitätsbedingungen als auch vom Versuch, verschiedene Nutzenszenarios zu simulieren, herleiten.

Nachteile:

- Die geschickte Variation von Gewichten ist schwierig.
- Der letzte Punkt zusammen mit der Tatsache, daß bei nichtkonvexem $F(R) + (\mathbb{R}_+^n \cup \{0\})$ nicht alle Paretooptima Minima eines Problems (2.11) sind, welche Gewichtung auch immer man probieren mag, macht die Repräsentativität der erzeugten Lösungsmenge fraglich.

Abbildung 2.2: Die beiden gestrichelten Geraden stellen Indifferenzkurven einer Nutzenfunktion (negative gewichtete Zielfunktionssumme) dar. Da Niveau a kleiner als Niveau b ist, kann das Paretooptimum x_1 keine Lösung des dargestellten gewichteten Zielfunktionssummenproblems sein. x_1 ist für keine zugelassene Gewichtung Lösung eines gewichteten Zielfunktionssummenproblems.



2.3.1.2 Methode der ε -Schranken

Wieder soll die Bearbeitung mehrerer skalarer Ersatzprobleme eine Teilmenge der Menge aller Paretooptima hervorbringen. Diese skalaren Ersatzprobleme sind von der Form

$$\begin{aligned} \min_{x \in R} f_r(x), r \in \{1, \dots, n\}, \\ f_j(x) \leq \varepsilon_j, r \neq j = 1, \dots, n. \end{aligned} \quad (2.12)$$

Man beachte, daß Lösungen von (2.12) auch lediglich schwach effizient sein können, aber mindestens ein Paretooptimum unter ihnen ist. Für eine eindeutige Lösung ist die Effizienz natürlich gesichert. Eine Alternative \bar{x} ist genau dann paretooptimal, falls sie Lösung des Problems (2.12) für jedes Referenzziel $l \in \{1, \dots, n\}$ mit $\varepsilon_l = f_l(\bar{x}), l \neq i = 1, \dots, n$, ist.

Anstatt der Gewichte kann man jetzt die maximal zugelassenen Zielwerte $(\varepsilon_1, \dots, \varepsilon_{r-1}, \varepsilon_{r+1}, \dots, \varepsilon_n)$ und das Referenzziel $r \in \{1, \dots, n\}$ variieren - wieder werden verschiedene Präferenzszenarien durchgespielt (siehe Abschnitt 2.1.2).

Zusammenfassung:

Vorteile:

- Unabhängig von der Form von $F(R)$ können alle Paretooptima erreicht werden, indem man skalare Ersatzprobleme der Form (2.12) löst. Dabei sei vorausgesetzt, daß eine skalare Optimierungstechnik zur Verfügung steht, die alle globalen Minima eines Minimierungsproblems (2.12) finden kann.

Nachteile:

- Schwierigkeiten können aus dem Problem entstehen, die Zielschranken $\varepsilon \in \mathbb{R}^{n-1}$ und die Referenzziele sinnvoll zu wählen und zu variieren sowie dabei,

- schwach effiziente Lösungen unter möglichst geringem Aufwand auszusieben.

Haimes beschreibt die Methode der ε -Schranken wie eingeführt in [25], Marglin schlägt in [48] einen ähnlichen Ansatz vor.

2.3.1.3 Methode der Gleichungsnebenbedingungen

Dieser Ansatz ist der Methode der ε -Schranken verwandt und wurde von J.G. Lin entwickelt ([45], [46], [47]).

Je Szenario wird nur eine Referenzziefunktion $f_r, r \in \{1, \dots, n\}$, minimiert. Die restlichen Ziele übernehmen die Rolle von Gleichungsnebenbedingungen:

$$\begin{aligned} \min_{x \in R} f_r(x), \\ f_j(x) = \varepsilon_j, r \neq j = 1, \dots, n. \end{aligned} \quad (2.13)$$

Lin formuliert folgende notwendige und hinreichende Bedingung für die Effizienz einer Alternative: Für jedes $\varepsilon^* \in \mathbb{R}^{n-1}$, so gewählt, daß $\cap_{r \neq i=1, \dots, n} \{x \in R | f_i(x) \leq \varepsilon_i^*\} \neq \emptyset$, ist die Lösung x^* von (2.13) effizient genau dann, wenn für alle ε , für die

$$\Phi_r(\varepsilon) := \min_{x \in R} \{f_r(x) | f_i(x) \leq \varepsilon_i, r \neq i = 1, \dots, n\} \quad (2.14)$$

eine endliche Lösung hat, gilt:

$$\Phi_r(\varepsilon) > f_r(x^*) \quad \forall \quad \varepsilon \leq \varepsilon^*, \varepsilon \neq \varepsilon^*. \quad (2.15)$$

Dies ist mehr oder weniger eine Umformulierung der Definition von Effizienz.

Zusammenfassung:

Vorteile:

- Wie bei der Methode der ε -Beschränkungen können von der Methode der Gleichungsnebenbedingungen mit einer entsprechend guten skalaren Optimierungsmethode als Werkzeug alle Paretooptima gefunden werden, gleichgültig, ob $F(R) + (\mathbb{R}_+^n \cup \{0\})$ konvex ist oder nicht.

Nachteile:

- Es bleibt die Frage der passenden Variation von ε , vor allem bei endlichen Mengen $F(R)$ und der sinnvollen Variation der jeweiligen Referenzziele.
- Außerdem dürfte Lins formulierte notwendige und hinreichende Bedingung für Paretooptimalität, die im Verfahren als Identifikationsmittel eingesetzt wird, speziell bei nicht differenzierbaren Zielfunktionen schwer nachweisbar sein.

2.3.1.4 Hybrider Wichtungs- und Nebenbedingungsansatz

In [9] weisen Chankong und Haimes auf eine Kombination aus Wichtungsmethode und Methode der ε -Schranken hin:

$$\begin{aligned}
 \min_{x \in R} \sum_{i=1}^n w_i f_i(x), \\
 F(x) \leq \hat{\varepsilon}, \\
 \sum_{i=1}^n w_i = 1, w_i > 0 \quad \forall i = 1, \dots, n, \\
 \hat{\varepsilon} \in \{\varepsilon \in \mathbb{R}^n \mid \{x \in R : F(x) \leq \varepsilon\} \neq \emptyset\}.
 \end{aligned} \tag{2.16}$$

Wendel und Lee ([66]) wie auch Corley ([12]) haben bewiesen, daß ein $\bar{x} \in R$ genau dann effizient ist, wenn es (2.16) für eine bestimmte Wertebelegung von $w, \varepsilon \in \mathbb{R}^n$, die die Bedingungen aus (2.16) erfüllt, minimiert.

Zusammenfassung:

Vorteile:

- Die Idee hier ist offensichtlich, die Wichtungsmethode und die Methode der ε -Schranken zu kombinieren und dabei die meisten ihrer Nachteile auszuschalten.
- Mit einer guten skalaren Optimierungsmethode als Hilfsmittel kann selbst bei nichtkonvexem $F(R) + (\mathbb{R}_+^n \cup \{0\})$ jedes Paretooptimum erreicht werden.
- Jedes Ergebnis eines Suchdurchganges ist effizient.

Nachteile:

- Leider bleibt die Schwierigkeit, ε und w geschickt zu belegen.

2.3.1.5 Methode der gewichteten L_p -Metrik

Auch mit Hilfe der Methode des globalen Kriteriums, die bereits in Abschnitt 2.1.4 vorgestellt wurde, kann man laut [49] eine Generierungstechnik basteln. Dazu führt man in die jeweiligen Metriken Gewichtskoeffizienten ein. Für $1 \leq p < \infty$ betrachte man das Problem

$$\begin{aligned}
 \min_{x \in R} \left(\sum_{i=1}^n w_i (f_i(x) - f_i^*)^p \right)^{1/p}, \\
 w_i \geq 0 \quad \forall i = 1, \dots, n, \\
 \sum_{i=1}^n w_i = 1.
 \end{aligned} \tag{2.17}$$

Im Falle $p = \infty$ erhält (2.17) die Form

$$\begin{aligned}
 \min_{x \in R} \{ \max_{1 \leq i \leq n} [w_i (f_i(x) - f_i^*)] \}, \\
 w_i \geq 0 \quad \forall i = 1, \dots, n, \\
 \sum_{i=1}^n w_i = 1.
 \end{aligned} \tag{2.18}$$

Die Rolle von (f_1^*, \dots, f_n^*) bekommt meist der Idealpunkt, falls dieser existiert, kann aber auch von jedem $y \in \mathbb{R}^n$ mit $y_i \leq f_i(x) \quad \forall x \in R$ in jeder Komponente $i \in \{1, \dots, n\}$ übernommen werden.

Eine Lösung von (2.17) ist effizient, falls sie eindeutig ist oder alle Gewichte $w_i, i = 1, \dots, n$, größer als 0 sind. Jede Lösung von Problem (2.18) ist mindestens schwach effizient und mindestens eine unter ihnen effizient.

Analog zur Wichtungsmethode werden die Gewichte variiert, um eine Teilmenge der Menge aller (schwach) effizienten Punkte zu erhalten. Mit $1 \leq p < \infty$ sind lediglich für konvexes $F(R) + (\mathbb{R}_+^n \cup \{0\})$ alle Paretooptima erlangbar, wenn man die Gewichte $w \geq 0$ geschickt genug wählt. Für $p = \infty$ existiert zu jedem effizienten Punkt $x \in R$ ein Gewichtsvektor $w \in \mathbb{R}^n$ mit $w_i \geq 0 \quad \forall i = 1, \dots, n$, so daß x das entsprechende Problem (2.18) minimiert.

Zusammenfassung:

Vorteile:

- Der Grundgedanke dieses Ansatzes ergibt sich aus dem Grundgedanken hinter der Methode des globalen Kriteriums.
- Mit der L_∞ -Metrik kann man, unabhängig von der Form der Menge $F(R)$, an alle effizienten Alternativen herankommen.
- Jeder Punkt, den man unter Verwendung der L_p -Metrik mit $1 \leq p < \infty$ erhält, ist sicher paretooptimal, wenn alle Gewichte positiv sind.

Nachteile:

- Bei nichtkonvexem $F(R) + (\mathbb{R}_+^n \cup \{0\})$ kann man mit der L_p -Metrik mit $1 \leq p < \infty$ eventuell nicht alle Elemente der Effizienzmenge erzeugen.
- Die Verwendung der L_∞ -Metrik kann auch lediglich schwach effiziente Ergebnisse liefern.
- Wie bereits bei den vorherigen Methoden ist auch hier die passende Variation der Gewichte keine leichte Aufgabe.

2.3.1.6 Hyperbel Methode

Nach [16] ist eine weitere Möglichkeit, die Mehrzielfunktion zu skalarisieren,

$$\begin{aligned} & \max_{(x,t) \in R \times \mathbb{R}^t}, \\ & \prod_{i=1}^n (z_i - w_i t - f_i(x)) = c, \\ & x \in R, \\ & z_i - w_i t - f_i(x) \geq 0, i = 1, \dots, n. \end{aligned} \tag{2.19}$$

$z \in \mathbb{R}^n, w \in \mathbb{R}_+^n$ und $c > 0$ sind im Vorhinein festzulegende Parameter. In [24] wird bewiesen, daß das \bar{x} aus jedem Maximierer (\bar{x}, \bar{t}) von Problem (2.19) paretooptimal ist. Außerdem können

bei nach unten beschränktem und abgeschlossenem $F(R)$ zu jeder paretooptimalen Alternative $\hat{x} \in R$ und zu jedem $\varepsilon > 0$ Parameter $z \in \mathbb{R}^n, w \in \mathbb{R}_+^n$ und $c > 0$ gefunden werden, so daß die Lösung (\bar{x}, \bar{t}) des entsprechenden Problems (2.19) $\|F(\bar{x}) - F(\hat{x})\| < \varepsilon$ erfüllt.

Zusammenfassung:

Vorteile:

- Gute Variation der Parameter des skalaren Ersatzproblems kann zur Approximation einer repräsentativen Teilmenge der Menge aller Paretooptima führen. Alle effizienten Punkte können nämlich im Zielraum beliebig nahe approximiert werden.

Nachteile:

- Natürlich ist wieder die passende Parametervariation schwer zu finden.

2.3.1.7 Normalen-Rand Intersektion

Betrachtet wird ein Mehrzielproblem mit zweimal stetig differenzierbaren Zielfunktionen und Nebenbedingungsfunktionen $h : \mathbb{R}^m \mapsto \mathbb{R}^{n_1}, x \mapsto h(x), g : \mathbb{R}^m \mapsto \mathbb{R}^{n_2}, x \mapsto g(x), n_1, n_2 \in \mathbb{N}$, die $R := \{x \in \mathbb{R}^m | h(x) = 0, g(x) \leq 0, a \leq x \leq b\}$ bestimmen. Für die vorkommenden Box-Constraints gilt $a \in (\mathbb{R} \cup \{-\infty\})^n, b \in (\mathbb{R} \cup \{\infty\})^n$. Außerdem findet der Idealpunkt (f_1^*, \dots, f_n^*) , der ja aus den minimalen Zielwerten der einzelnen Zielfunktionen besteht (sofern er existiert), Anwendung. Der Koordinatenursprung im \mathbb{R}^n werde mittels Achsentransformation auf den Idealpunkt verlegt.

Das und Dennis ([13]) bilden die $n \times n$ -Matrix

$$\Phi = \begin{pmatrix} f_1(x_1^*) - f_1^* & f_2(x_1^*) - f_2^* & \dots & f_n(x_1^*) - f_n^* \\ f_1(x_2^*) - f_1^* & f_2(x_2^*) - f_2^* & \dots & f_n(x_2^*) - f_n^* \\ \dots & \dots & \dots & \dots \\ f_1(x_n^*) - f_1^* & f_2(x_n^*) - f_2^* & \dots & f_n(x_n^*) - f_n^* \end{pmatrix}, \tag{2.20}$$

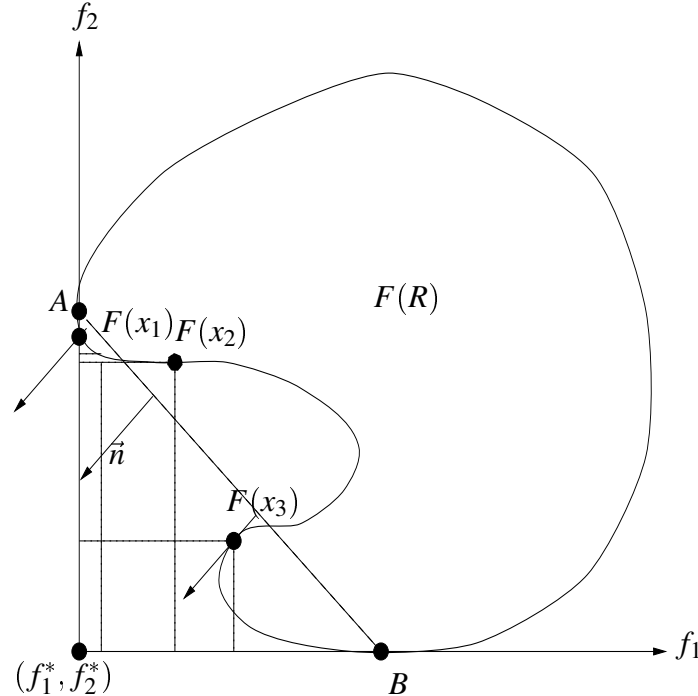
wobei x_i^* das Ergebnis der individuellen Minimierung von $f_i, i = 1, \dots, n$, über R ist. Sie nennen das Gebilde

$$\Phi w, \sum_{i=1}^n w_i = 1, w_i \geq 0, i = 1, \dots, n, \tag{2.21}$$

KHIM, eine Abkürzung für **konvexe Hülle der individuellen Minima**. Sie betrachten weiterhin den niedrigstdimensionalen affinen Raum, der KHIM enthält, bezeichnen ihn mit $KHIM_\infty$ und vergeben an die kleinste einfach verbundene Menge, die jeden Punkt aus dem Schnitt des Randes $\delta F(R)$ von $F(R)$ mit $KHIM_\infty$ enthält, den Namen **KHIM+**. Bei $n = 2$ sind **KHIM+** und **KHIM** identisch.

$\vec{n} \in \mathbb{R}^n$ bezeichne die Einheitsnormale zum KHIM-Simplex, die zum Koordinatenursprung zeigt, der ja im Idealpunkt liegt. Ist $F(R) + (\mathbb{R}_+^n \cup \{0\})$ konvex, dann ist die Alternative $x \in R$ mit einem Zielwertvektor $F(x)$, der der dem Ursprung am nächsten liegende Schnittpunkt von

Abbildung 2.3: $\overline{AB} = \text{KHIM}$. Bei nichtkonvexer Effizienzmenge sind alle paretooptimalen Alternativen x_1 mit $F(x_1)$ im konvexen Teil von $F(R) + (\mathbb{R}_+^n \cup \{0\})$ durch Normalen-Rand-Intersektion zu erlangen. x_2 mit $F(x_2)$ im nichtkonvexen Teil kann z.B. nicht gefunden werden. Außerdem können auch dominierte Alternativen wie x_3 Ergebnisse dieser Methode sein.



$\Phi w + t\vec{n}, \sum_{i=1}^n w_i = 1, t \geq 0, w \geq 0$, mit $F(R)$ ist, global paretooptimal. Im Falle $n = 2$ kann zu jedem nichtdominierten Punkt $z \in \delta(F(R))$ ein Gewichtsvektor $\bar{w} \in \mathbb{R}^n$ gefunden werden, so daß sich $\Phi\bar{w} + t\vec{n}$ und $\delta(F(R))$ in z schneiden. Dies ist für $n > 3$ nicht immer möglich, da hier $\text{KHIM} \neq \text{KHIM}^+$ auftreten kann und nicht alle effizienten Punkte $\in F(R)$ (in \vec{n} -Richtung) unter KHIM liegen. Bei nichtkonvexem $F(R) + (\mathbb{R}_+^n \cup \{0\})$ könnte der Normalen-Rand-Schnittpunkt, der dem Ursprung am nächsten liegt, dominiert sein und einige global effiziente Punkte könnten in $(-\vec{n})$ -Richtung angesiedelt, also für die beschriebene Technik unerreichbar sein.

Alternativen mit einem Zielwertvektor, der der dem Ursprung am nächsten liegende Schnittpunkt von $\Phi w + t\vec{n}, \Phi w \in \text{KHIM}, \sum_{i=1}^n w_i = 1, w \geq 0$ mit $F(R)$ ist, lösen auch folgendes skalares Ersatzproblem:

$$\begin{aligned} \min_{(x,t) \in R \times \mathbb{R}_0^+} & -t, \\ \Phi w + t\vec{n} &= F(x), \\ R &:= \{x \in \mathbb{R}^m \mid h(x) = 0, g(x) \leq 0, a \leq x \leq b\}. \end{aligned} \quad (2.22)$$

Die Idee ist, (2.22) für verschiedene $w \geq 0, \sum_{i=1}^n w_i = 1$, zu lösen und somit mehrere Punkte $x \in R$ mit $F(x) \in \delta(F(R))$ zu finden, die insgesamt eine gute punktweise Approximation der Effizienzmenge bilden. Wenn die verschiedenen w gleichmäßig verteilt über $\{w \in \mathbb{R}^n \mid \sum_{i=1}^n w_i = 1, w \geq 0\}$ gewählt werden, liefert die Bearbeitung der entsprechenden Probleme (2.22) eine gute Teildarstellung der Effizienzmenge in $F(R)$, zumindest bei konvexem $F(R) + (\mathbb{R}_+^n \cup \{0\})$ und $n = 2$.

Der Einfachheit halber kann man die Normale \vec{n} ersetzen durch eine Quasi-Normalenrichtung

$$\vec{n} := -\Phi e, e = (1, \dots, 1). \quad (2.23)$$

Für zwei Gewichtsvektoren w und \bar{w} mit kleinem euklidischen Abstand $\|w - \bar{w}\|_2$ ist es nicht abwegig zu erwarten, daß Lösungen (x^*, t^*) und (\bar{x}^*, \bar{t}^*) der entsprechenden skalaren Ersatzprobleme (2.22) auch nahe beieinander liegen. (x^*, t^*) stellt demnach einen guten Startpunkt für die Lösung von Problem (2.22) mit $w = \bar{w}$ dar.

Das und Dennis zeigen, daß zwischen der Normalen-Rand Intersektion und der Skalarisierungstechnik der Summierung gewichteter Zielfunktionen ein enger Zusammenhang besteht. Aus den notwendigen Kuhn-Tucker Bedingungen für Optimalität in der skalaren Optimierung kann man bei jeder Lösung (\hat{x}, \hat{t}) von (2.22) Gewichte $\hat{\alpha} \in \mathbb{R}^n$ herleiten, für die \hat{x} auch

$$\min_{x \in R} \sum_{i=1}^n \hat{\alpha}_i f_i(x) \quad (2.24)$$

löst. Der hergeleitete Gewichtsvektor kann allerdings auch negative Komponenten besitzen. In solch einem Fall ist entweder die Effizienz von \bar{x} anzuzweifeln oder/und $F(\hat{x})$ liegt in einem nichtkonvexen Teilbereich der Effizienzmenge in $F(R)$. Das hergeleitete $\hat{\alpha} \in \mathbb{R}^n$ enthüllt den Grad der Bedeutung, der jedem einzelnen Ziel bei der Erarbeitung der Lösung von (2.22) zukam.

Zusammenfassung:

Vorteile:

- Die Methode der Normalen-Rand Intersektion basiert auf einer guten Idee.
- Bei $n = 2$ und konvexem $F(R) + (\mathbb{R}_+^n \cup \{0\})$ kann mit Hilfe einer guten skalaren Optimierungstechnik in der Tat jeder effiziente Punkt ausgemacht werden.
- Es ist im obigen Fall relativ einfach, die Gewichte so zu variieren, daß das Verfahren eine repräsentative Teilmenge der Effizienzmenge hervorbringt.

Nachteile:

- Die Normalen-Rand Intersektion ist sicher nicht auf diskrete Probleme, d.h. Probleme mit endlichem R , anwendbar. Ziel- und Nebenbedingungsfunktionen sollten mindestens stetig differenzierbar sein, wenn nicht sogar zweimal stetig differenzierbar (wie es bei kontinuierlichen Methoden allgemein üblich ist).
- Nur bei Problemen mit zwei Zielen und konvexer Menge $F(R) + (\mathbb{R}_+^n \cup \{0\})$ kann das Verfahren alle Paretooptima ausfindig machen. Bei $n > 3$ und/oder nichtkonvexem $F(R) + (\mathbb{R}_+^n \cup \{0\})$ sind eventuell einige effiziente Punkte nicht erreichbar.
- Globale Effizienz kann für Lösungen nur bei konvexem $F(R) + (\mathbb{R}_+^n \cup \{0\})$ garantiert werden, ansonsten könnte die Normalen-Rand Intersektion auch dominierte Punkte liefern.

2.3.1.8 Methode der Effizienzmengenschätzung

Diese Methode, kurz **NISE** (Noninferior Set Estimation), wurde von Cohon für Probleme mit zwei Zielen entwickelt (siehe [11]). Theoretisch ist sie auch auf Probleme mit mehr als zwei Zielen anwendbar, dabei steigt aber der Rechenaufwand stark an. Außerdem kümmert sich Cohon nur um Probleme mit linearen Zielfunktionen, da hier für alle $F(c)$ auf der Verbindungsgeraden $\overline{F(a), F(b)}$ von $F(a)$ nach $F(b)$ gilt, daß c auf der Verbindungsgeraden $\overline{a, b}$ von a nach b liegt, wobei a, b, c aus R seien.

Eine Forderung schränkt den Kreis der von NISE bearbeitbaren Probleme stark ein - die Konvexität von R und $F(R)$.

NISE versucht einige (schwach) effiziente Alternativen $x \in R$ zu finden, indem es mehrere Probleme gewichteter Zielfunktionssummen löst und dann die Effizienzmenge in $F(R)$ durch Verbindungsgeraden zwischen den Zielwertvektoren $F(x)$ approximiert. Bei linearen Zielfunktionen stellen die Verbindungsgeraden zwischen den erzeugten (schwach) effizienten $x \in R$ eine Approximation der Menge aller Paretooptima in R dar. Im Falle von nichtlinearen Zielfunktionen kann die Ermittlung einer Alternative $x \in R$, deren Zielwertvektor $F(x)$ Element einer Verbindungsgeraden zwischen zwei Punkten in $F(R)$ ist, schwierig sein. Dennoch sind in dieser Vorstellung der Methode auch Probleme mit nichtlinearen Zielfunktionen zugelassen. Eine Approximation wird als gut genug erachtet, wenn der maximal mögliche Fehler, eine Größe, die die Qualität der Approximation bewertet, unter einem vom Entscheidungsträger festgelegten Wert liegt. Die wichtigsten Schritte von NISE lassen sich folgendermaßen gliedern:

Schritt 0: separate Minimierung jeder Zielfunktion:

Man berechne $x^i \in R$ mit

$$x^i = \min_{x \in R} f_{3-i}(x), i = 1, 2. \quad (2.25)$$

$s_1 = F(x^1)$ und $s_2 = F(x^2)$ seien die ersten beiden erarbeiteten (schwach) effizienten Punkte $\in F(R)$, x^1, x^2 die ersten beiden (schwach) effizienten Alternativen $\in R$. Die Verbindungsgerade zwischen s_1 und s_2 stellt die erste Approximation der Effizienzmenge in $F(R)$ dar; die Konvexität von $F(R)$ garantiert $\overline{s_1, s_2} \subseteq F(R)$.

Die Indifferenzkurven $0f_1(x) + 1f_2(x) = f_2(x^1)$ durch s_1 und $1f_1(x) + 0f_2(x) = f_1(x^2)$ durch s_2 schneiden sich im Punkt $M_1 \in \mathbb{R}^2$. Der Abstand $d_{1,2}$ zwischen M_1 und der Geraden $\overline{s_1, s_2}$ definiert den maximal möglichen Fehler der momentanen Approximation. Diese Definition ist sinnvoll, weil jeder nichtdominierte Punkt $y \in F(R)$ mit $f_1(x^2) \leq y_1 \leq f_1(x^1)$, der einen Punkt auf $\overline{s_1, s_2}$ dominiert, im Dreieck $\overline{s_1, s_2, M_1}$ liegt.

Jede Iteration $k \geq 1$ besteht aus

Schritt 1: Prüfung, ob NISE gestoppt werden kann:

Falls alle $d_{i,i+1} < \max E \quad \forall \quad i = 1, \dots, k$, wobei $\max E$ den maximal erlaubten Fehler bezeichnet, so wird die Menge $\overline{s_1, s_2} \cup \overline{s_2, s_3} \cup \dots \cup \overline{s_k, s_{k+1}}$ als annehmbare Approximation für die Effizienzmenge in $F(R)$ erachtet. Ansonsten fahre man mit **Schritt 2** fort.

Schritt 2: Erarbeitung von s_{k+2} :

Man bestimme $l := \max_{i=1, \dots, k} d_{i,i+1}$. Die Gerade $\overline{s_l, s_{l+1}}$ ($s_l = F(x^l), s_{l+1} = F(x^{l+1})$) scheint der am wenigsten akzeptable Teil der aktuellen Approximation zu sein. Daraufhin löse man

$$\min_{x \in R} (f_2(x^{l+1}) - f_2(x^l))f_1(x) + (f_1(x^l) - f_1(x^{l+1}))f_2(x), \quad (2.26)$$

wobei es sich um die Minimierung einer gewichteten Zielfunktionssumme handelt, deren Indifferenzkurve dieselbe Steigung wie $\overline{s_l, s_{l+1}}$ hat. Mit dem Zielwertvektor der Lösung \hat{x} wird $\overline{s_l, s_{l+1}}$ in der Approximation durch $s_l, F(\hat{x}) \cup F(\hat{x}), s_{l+1}$ ersetzt, was die momentane Approximation weiter verbessert.

Die Punkte $\{s_1, s_2, \dots, s_{k+1}, F(\hat{x})\}$ werden mit $s_i, i = 1, \dots, k+2$ neu benannt, wobei der Index mit sinkendem Wert von f_1 wächst. $F(\hat{x})$ habe nun den Namen $s_j, j \in \{1, \dots, k+2\}$ erhalten.

Gilt $s_j \in \overline{s_{j-1}, s_{j+1}}$, dann ist $d_{j-1,j} = d_{j,j+1} = 0$. Ansonsten erinnere man sich der Indifferenzkurven durch s_{j-1} und s_{j+1} , die Nutzenniveaus der jeweiligen gewichteten Zielfunktionssummen darstellen. Ihr Schnittpunkt M_{j-1} ist nicht mehr von Bedeutung, dafür sind es jedoch ihre Schnittpunkte \hat{M} und \bar{M} mit der Indifferenzkurve durch s_j , die aus Problem (2.26) herrührt. Die Elemente der Menge $\{M_1, \dots, M_k, \hat{M}, \bar{M}\} \setminus \{M_j\}$ werden neu durchnummeriert, mit wachsendem Index, je weiter links auf der Ebene der Punkt liegt, ebenso wie die maximal möglichen Fehler $d_{i,i+1}, i = 1, \dots, k+1$ aller Geradenabschnitte der Approximation. Man kehre zu Schritt 1 zurück mit $k = k+1$.

Falls die Menge $F(R) + (\mathbb{R}_+^n \cup \{0\})$ nichtkonvex ist, können die effizienten Punkte in den nichtkonvexen Teilen von $F(R) + (\mathbb{R}_+^n \cup \{0\})$ nicht angenähert werden, weil keine Summierung gewichteter Zielfunktionen zu einer skalaren Ersatzfunktion führt, zu deren Minimierern die Paretooptima mit diesen Zielwertvektoren gehören.

Weitere Schwierigkeiten ergeben sich bei $n > 2$. In einem Dreizielproblem gehen die Gewichte der gewichteten Zielfunktionssummen der aktuellen Iteration aus den Gewichten der gewichteten Zielfunktionssummen der vorherigen Iterationen hervor, da sie ja derart gewählt werden, daß die Indifferenzkurven oder besser Indifferenzebenen der entsprechenden linearen Nutzenfunktionen zu Ebenen $\overline{s_{j-1}, s_j, s_{j+1}}, j > 1$ parallel sind, wobei $s_{j-1}, s_j, s_{j+1} \in F(R)$ Elemente der letzten Approximation der Effizienzmenge sind. Bei einigen Formen der Menge $F(R) + (\mathbb{R}_+^n \cup \{0\})$, selbst wenn sie konvex ist, können eventuell nicht alle effizienten Punkte Minima eines der derart erzeugten Probleme der gewichteten Zielfunktionssumme sein. Die zugehörigen Paretooptima kann man nur erreichen, indem man für die Zweizielszenarien f_1, f_2 und f_1, f_3 , und f_2, f_3 zweidimensionale NISE-Suchprozeduren startet.

Cohon empfiehlt für $n = 4$ ein ähnliches Vorgehen, nämlich neben dem vierdimensionalen NISE-Suchlauf vier dreidimensionale NISE-Suchläufe durchzuführen, welche sich mit den vier Dreizielproblemen beschäftigen, die sich aus dem Vierzielproblem durch völliges Ignorieren eines der Ziele ergeben.

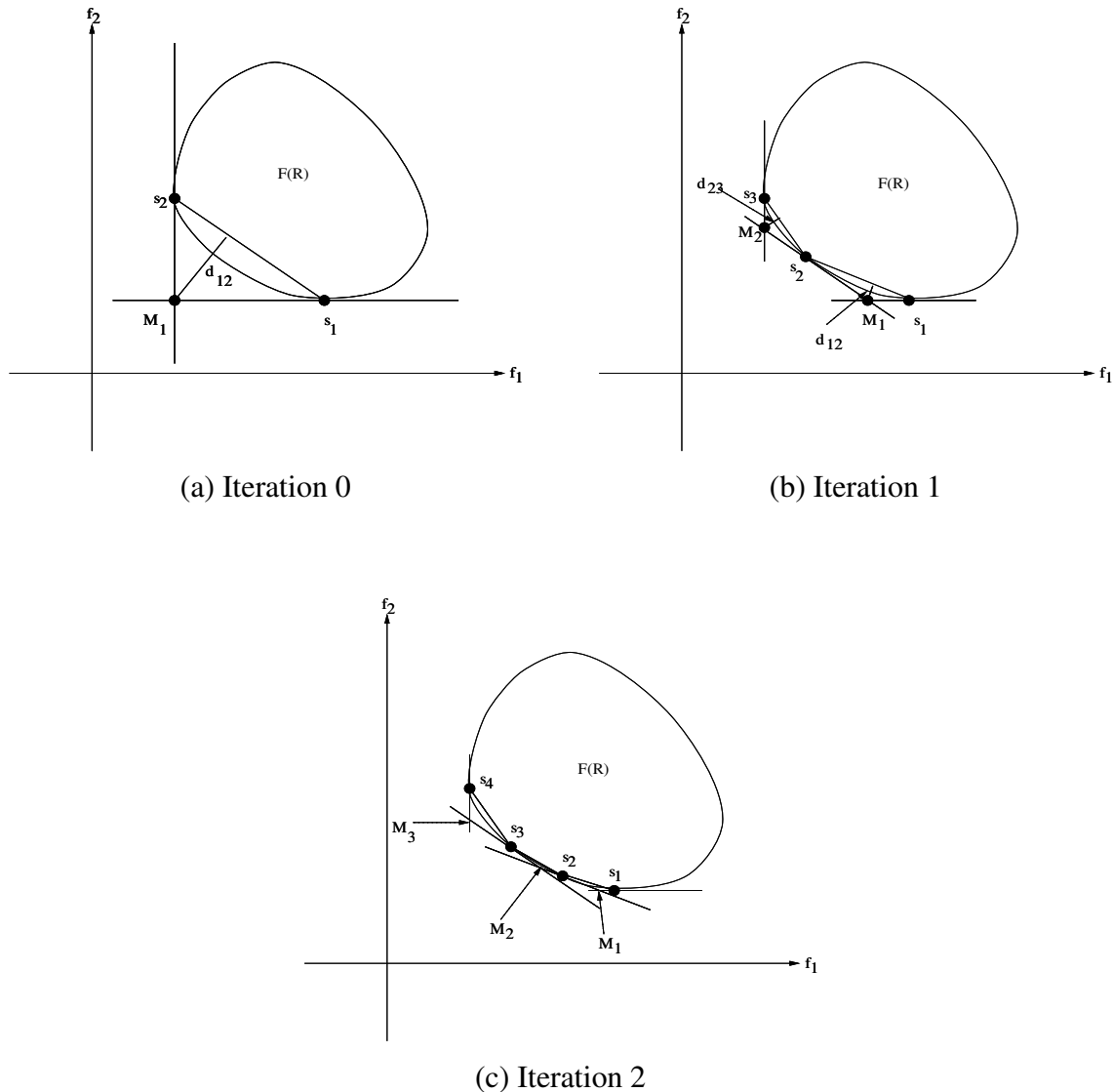
Man kann sich leicht vorstellen, daß Vektoroptimierungsprobleme mit höherer Zieldimension $n \geq 3$ der NISE-Taktik sehr hohen Rechenaufwand verursachen, da ja jeweils einmal ein n -dimensionaler, n mal $(n-1)$ -dimensionale, $n(n-1)$ mal $(n-2)$ -dimensionale und $n(n-1)(n-2) \dots 3$ mal zweidimensionale NISE-Prozeduren laufen müssen.

Zusammenfassung:

Vorteile:

- Durch Approximation der Effizienzmengen in R und $F(R)$ versucht das Verfahren, dem Entscheidungsträger einen bestmöglichen Eindruck von der Menge aller paretooptimalen Alternativen zu vermitteln.
- Selbst bei nichtlinearen Zielfunktionen erarbeitet NISE eine repräsentative Teilmenge der Paretooptima in R (keine Approximation der gesamten Effizienzmenge!), da die Schritte zur Verbesserung der Approximation der Effizienzmenge in $F(R)$ zu einer guten Wahl

Abbildung 2.4: Drei Iterationen von NISE verbessern die Approximation der Effizienzmenge in $F(R)$ von $\overline{s_1, s_2}$ in (a) zu $\overline{s_1, s_2} \cup \overline{s_2, s_3} \cup \overline{s_3, s_4}$ in (c).



der Gewichte für die gewichteten Zielfunktionssummen führen - hier fungiert NISE eher als Wichtungsmethode, bei der die schwierige Aufgabe der Gewichtsvariation geschickt angegangen wird.

Nachteile:

- Der Einsatz von NISE ist nur bei konvexer Menge $F(R)$ sinnvoll.
- NISE kann bei vernünftigem Rechenaufwand nur mit zwei Zielen zurechtkommen.
- Im Falle von nichtlinearen Zielfunktionen kann es sich als zu schwierig oder aufwendig erweisen, aus der Approximation der Effizienzmenge in $F(R)$ eine Approximation der Menge aller Paretooptima in R zu errechnen.

2.3.1.9 Stückweise Lineare Approximation

Dieser Name würde auch auf das eben beschriebene **NISE**-Verfahren passen und tatsächlich ähnelt die von Polishchuk in [50] entwickelte Mehrzielmethode NISE sehr stark.

Damit ein Vektoroptimierungsproblem mit **Stückweiser Linearer Approximation** bearbeitbar ist, muß $F(R)$ konvex sein. Außerdem ist dieses Verfahren wie NISE nur für Zweizielprobleme konstruiert.

Wieder ist hier die Taktik, einige paretooptimale Alternativen zu erarbeiten, diesmal durch Minimierung einer der Zielfunktionen bei Beschränkung der restlichen Zielfunktionen durch maximal erlaubte Zielwerte, und dann durch Verbindungsgeraden zwischen ihren Zielwertvektoren die Effizienzmenge in $F(R)$ zu approximieren. Die Verbindungsgeraden und somit alle Teile der Näherungsmenge liegen in $F(R)$, da der Zielraum ja konvex ist. Die Approximation wird von Iteration zu Iteration verbessert, bis ihr maximaler Fehler unter dem maximal zugelassenen Fehler, der vom Entscheidungsträger im Vorhinein festgelegt wird, liegt.

Die gesamte Effizienzmenge in $F(R)$ läßt sich in der Form

$$\xi(t) := \min_{x \in R} \{f_2(x) \mid f_1(x) \leq t\}, t \in [t_{min}, t_{sup}], \quad (2.27)$$

niederschreiben, wobei $t_{min} < t_{sup}$ bestimmt werden durch

$$t_{min} := \min_{x \in R} f_1(x), \quad (2.28)$$

$$t_{sup} := \max_{x \in R} f_1(x), \quad (2.29)$$

$$t_{max} := \min \{t \in [t_{min}, t_{sup}] \mid \xi(t) = \xi(t_{sup})\}. \quad (2.30)$$

$t^0 := t_{min}, t^1 := t_{max}$. Es benenne $-\lambda_j \leq 0, j \in \mathbb{N}_0$, die Steigung der Kurve, die vom effizienten Rand von $F(R)$ gebildet wird, im Punkte t^j ; der maximale Fehler $\delta_{[t^j, t^{j+1}]}(\xi)$ der Näherungsgeraden $(t^j, \xi(t^j)), (t^{j+1}, \xi(t^{j+1}))$ aus der momentanen Approximation für die Effizienzmenge errechnet sich für $j \in \mathbb{N}_0$ auf folgende Weise: Die Geraden $\xi(t^j) - \lambda_{t^j}(t - t^j)$ und $\xi(t^{j+1}) - \lambda_{t^{j+1}}(t - t^{j+1})$ schneiden sich in \bar{t}^j . Mit den Definitionen

$$\zeta_{[t^j, t^{j+1}]}(t) := \begin{cases} \xi(t^j) - \lambda_{t^j}(t - t^j), & t^j \leq t \leq \bar{t}^j, \\ \xi(t^{j+1}) - \lambda_{t^{j+1}}(t - t^{j+1}), & t^{j+1} \geq t \geq \bar{t}^j, \end{cases} \quad (2.31)$$

und

$$A_{[t^j, t^{j+1}]}(t) := \xi(t^j) + \frac{\xi(t^{j+1}) - \xi(t^j)}{t^{j+1} - t^j}(t - t^j) \quad (2.32)$$

wird festgelegt:

$$\delta_{[t^j, t^{j+1}]}(\xi) := \|A_{[t^j, t^{j+1}]} - \zeta_{[t^j, t^{j+1}]}\|_{\infty} := \max_{t \in [t^j, t^{j+1}]} |A_{[t^j, t^{j+1}]}(t) - \zeta_{[t^j, t^{j+1}]}(t)|. \quad (2.33)$$

Jede Iteration $k > 1$ besteht aus den Schritten

Schritt 1: Prüfung, ob der Algorithmus gestoppt werden kann:

Falls $\forall j = 0, \dots, n-1$ gilt $\delta_{[t^j, t^{j+1}]}(\xi) \leq \varepsilon$ ($n+1$ effiziente Alternativen seien bereits erzeugt worden) und $\varepsilon > 0$ der maximal erlaubte Approximationsfehler ist, dann wird die aktuelle Approximation $(t^0, \xi(t^0)), (t^1, \xi(t^1)) \cup \dots \cup (t^{n-1}, \xi(t^{n-1})), (t^n, \xi(t^n))$ der Effizienzmenge in $F(R)$ als gut genug eingestuft. Ansonsten führe man für jedes (t^l, t^{l+1}) mit

$\delta_{[t^l, t^{l+1}]}(\xi) > \varepsilon, l \in \{0, \dots, n-1\}$, **Schritt 2** aus.

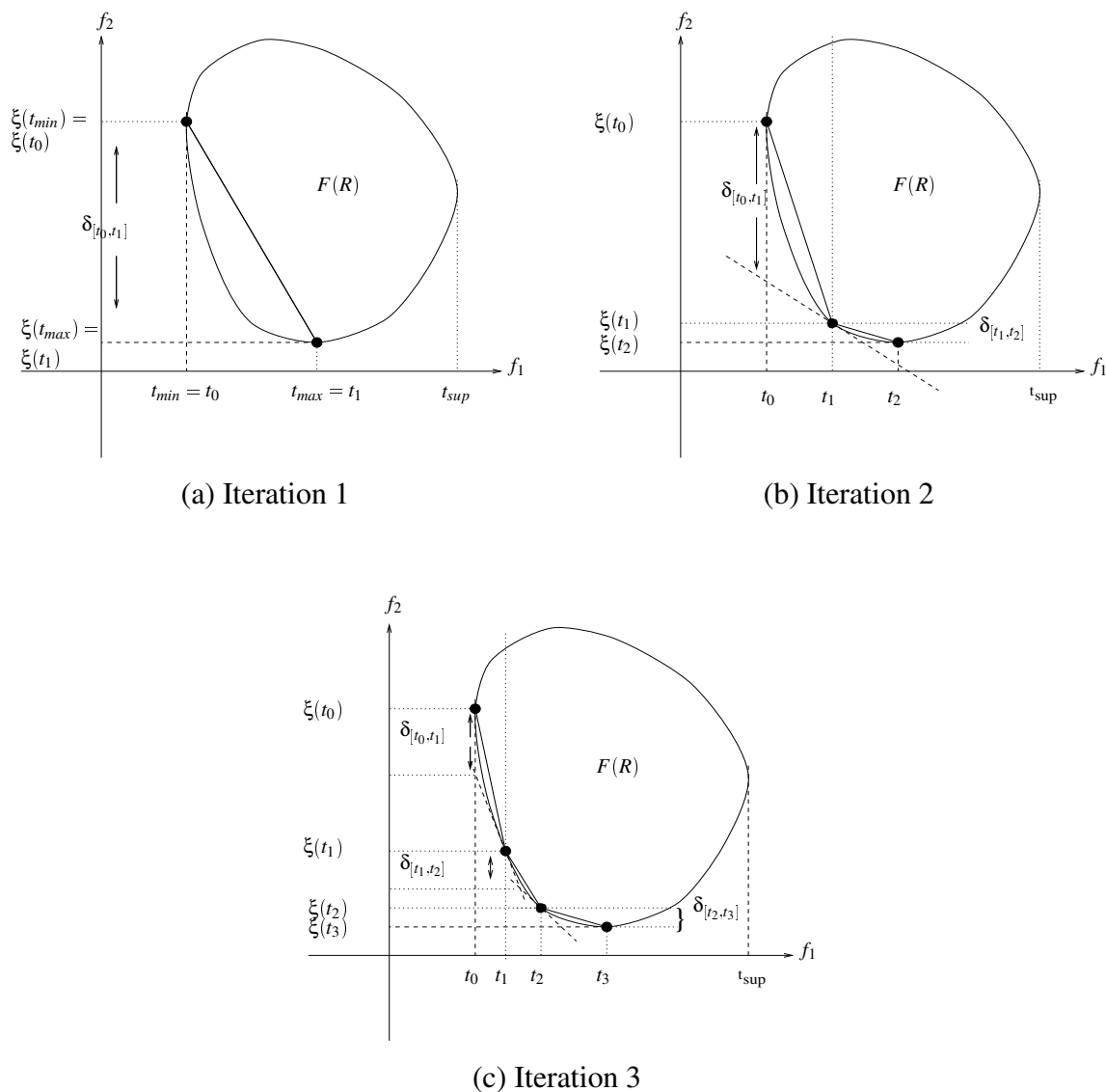
Schritt 2: Verbesserung der Approximation:

Man berechne $\xi(t^{l, l+1})$ für $t^{l, l+1} := \frac{t^l + t^{l+1}}{2}$ sowie $\delta_{[t^l, t^{l+1}]}(\xi)$ und $\delta_{[t^{l+1}, t^{l+1}]}(\xi)$.

Schritt 3: Neubenennung:

Man benenne alle $t \in [t_{min}, t_{max}]$, für die $\xi(t)$ bisher berechnet wurde, neu mit $t_j, j \in \mathbb{N}_0$, wobei der Index j zusammen mit dem Wert von t wachse. Analog benenne man die maximal möglichen Fehler neu. Man gehe zu **Schritt 1** zurück; es beginnt Iteration $k = k + 1$.

Abbildung 2.5: Drei Iterationen der Stückweisen Linearen Approximation verbessern die Approximation der Effizienzmenge in $F(R)$ von $(t^0, \xi(t^0)), (t^1, \xi(t^1))$ in (a) zu $(t^0, \xi(t^0)), (t^1, \xi(t^1)) \cup (t^1, \xi(t^1)), (t^2, \xi(t^2)) \cup (t^2, \xi(t^2)), (t^3, \xi(t^3))$ in (c).



Für Probleme mit stetigen Zielfunktionen zeigt Polishchuk, daß die maximal nötige Anzahl von $\xi(t)$ -Auswertungen, bis der maximal mögliche Fehler jedes Streckenabschnittes der

Approximation der Effizienzmenge unter einer Vorgabe $\varepsilon > 0$ liegt, von der Ordnung $o(\varepsilon^{-1/2})$ ist.

Bei nichtkonvexer Menge $F(R) + (\mathbb{R}_+^n \cup \{0\})$ funktioniert die definierte Schätzung des maximal möglichen Fehlers nicht mehr, und die Näherungsmenge für die Effizienzmenge liegt eventuell nicht vollständig in $F(R)$. Dennoch ist es jedoch möglich, jeden effizienten Punkt aus $F(R)$ zu finden, in dem man t geschickt genug variiert.

Denkt man an eine Anwendungsausdehnung der Stückweisen Linearen Approximation auf Mehrzielprobleme mit $n \geq 3$, so liegt die Art der Realisierung nicht auf der Hand und man muß gewaltig ansteigenden Rechenaufwand erwarten.

Zusammenfassung:

Vorteile:

- Das Verfahren versucht die gesamte Effizienzmenge in $F(R)$ anzunähern.
- Auch wenn in R nicht die Menge aller Paretooptima approximiert wird, erarbeitet das Verfahren zumindest eine repräsentative Teilmenge dieser Menge, da die Schritte zur Verbesserung der Approximation der Effizienzmenge in $F(R)$ zu einer geschickten Variation des Parameters $t \in \mathbb{R}$ führen, der als Oberschranke für Ziel 1 agiert. Die Stückweise Lineare Approximation arbeitet auf R eher als Verfahren der ε -Schranken, bei dem die Schwierigkeit der ε -Variation gut gelöst ist.

Nachteile:

- Die Technik in der eben beschriebenen Ausführung kann nur mit Problemen mit konvexem $F(R)$ umgehen.
- In dieser Version ist nur die Anwendung auf Zweizielprobleme möglich. Eine Ausdehnung auf mehr als zwei Ziele ist vom Rechenaufwand her wahrscheinlich zu komplex.

2.3.1.10 Methode der Vergleichszahlen zum Auffinden von hochoptimalen und hocheffizienten Lösungen diskreter Vektoroptimierungsprobleme

”Hochoptimal” und ”hocheffizient” sind ein Versuch, die englischen Begriffe ”major optimal” und ”major efficient” ins Deutsche zu übertragen.

Die Methode der Vergleichszahlen von Hu und Sun ([32]) benötigt eine endliche, nicht zu große Alternativenmenge R in Vektoroptimierungsproblemen, damit sie verwendet werden kann. Obwohl viele Mehrzielprobleme von dieser starken Einschränkung ausgeschlossen sind, ist die Motivation hinter dieser Methode dennoch interessant: Es werden außer den Zielfunktionswerten keinerlei Informationen über F verlangt, Skalarisierungen des Mehrzielproblems sind in keinem Verfahrensschritt notwendig, und die Alternativen werden nicht absolut, sondern paarweise miteinander verglichen.

Hu und Sun führen zwei neue Optimalitätsbegriffe in der Vektoroptimierung ein.

$$H := \left\{ v \in \mathbb{R}^n \mid \frac{n}{2} - \frac{1}{2} \sum_{i=1}^n \operatorname{sgn}(v_i) \leq \frac{n}{2} \right\} \quad (2.34)$$

ist eine Menge, die zur Definition der **Hochoptimalität** und der **Hocheffizienz** verwendet wird:

Definition 2.3.1. $\bar{x} \in R$ heißt **hochoptimale** Alternative, falls

$$F(x) - F(\bar{x}) \in H \quad \forall x \in R \quad (2.35)$$

gilt, und **hocheffizient**, falls es kein $x \in R$ gibt mit

$$F(\bar{x}) - F(x) \in H \setminus \{0\}. \quad (2.36)$$

Es ist leicht zu beweisen, daß die Menge der hocheffizienten Punkte eine Teilmenge der Menge der hochoptimalen Punkte und diese wiederum eine Teilmenge der Menge aller Paretooptima ist. Außerdem kann gezeigt werden, daß

Satz 2.3.2.

$$\bar{x} \in R \text{ hochoptimal} \iff \inf_{x \in R} \sum_{i=1}^n \text{sign}[f_i(x) - f_i(\bar{x})] \geq 0, \quad (2.37)$$

$$\bar{x} \in R \text{ hocheffizient} \iff \inf_{x \in R: F(x) \neq F(\bar{x})} \sum_{i=1}^n \text{sign}[f_i(x) - f_i(\bar{x})] > 0. \quad (2.38)$$

Der Name **Vergleichszahl** bezeichnet das Konstrukt

$$C(x, y) := \sum_{i=1}^n \text{sign}[f_i(y) - f_i(x)], x, y \in R. \quad (2.39)$$

Indem man für alle Alternativen $x \in R$

$$C(x) := \min_{y \in R: F(y) \neq F(x)} C(x, y) \quad (2.40)$$

berechnet, erhält man eine Aussage über die Hochoptimalität oder Hocheffizienz jeder Alternative:

Satz 2.3.3.

$$C(\hat{x}) > 0 \implies \bar{x} \text{ ist hocheffizient}, \quad (2.41)$$

$$C(\hat{x}) \geq 0 \implies \bar{x} \text{ ist hochoptimal}, \quad (2.42)$$

$$C(\hat{x}) < 0 \implies \bar{x} \text{ ist nicht hochoptimal}. \quad (2.43)$$

Zusammenfassung:

Vorteile:

- Die Methode ist auf diskrete Probleme anwendbar.
- Die Minimierung von $C(x, y)$ über $\{y \in R | F(y) \neq F(x)\}$ liefert eine Aussage darüber, ob $x \in R$ zu einer Teilmenge der Menge aller Paretooptima gehört oder nicht.

Nachteile:

- Nur Probleme mit einer endlichen und genügend kleinen Alternativenmenge können von der Methode der Vergleichszahlen bearbeitet werden.
- Es gibt keinen Vorschlag, wie man die Information "x ∈ R ist nicht hochoptimal" weiterverarbeitet, d.h. wie man von x aus weitersuchen könnte, um an hochoptimale oder hocheffiziente Lösungen heranzukommen.

2.3.2 Kontinuierliches stochastisches Suchen

2.3.2.1 Methode der sich anpassenden Suche

Das Konzept hinter dieser Methode von Beeson und Meisel ([2]) ist, eine repräsentative Menge von Paretooptima zu erzeugen, deren Zielfunktionswerte in $F(R)$ so weit auseinanderliegen, daß sie wirkliche Handlungsalternativen darstellen.

Alle Zielfunktionen müssen differenzierbar sein, sonst kann das Verfahren nicht angewandt werden. Zunächst wird jede Zielfunktion f_i separat über R minimiert, was zu den Ergebnissen $x^i, i = 1, \dots, n$, führt. Diese x^i sind mindestens schwach effizient und, sollten sie ein eindeutiger Minimierer von f_i sein, paretooptimal. Denkt man an Prioritäten unter den Zielen, so stammen die $x^i, i = 1, \dots, n$, aus "extremalen" Gebieten in der Menge aller (schwach) effizienten Punkte, da in ihnen jeweils nur ein Ziel unter Vernachlässigung aller anderen berücksichtigt ist.

Diese verschiedenen "extremalen" Gebiete werden weiter erforscht, indem man von jedem Punkt $x^i, i = 1, \dots, n$, aus mehrere Suchprozeduren startet, die jeweils nach Paretooptima Ausschau halten. Diese Suchprozeduren gehorchen folgender Iterationsanweisung:

$$x(k+1) = x(k) - \beta(k)[J^T(x(k))w(k)] + \gamma(k) \quad \forall k \in \mathbb{N}_0. \quad (2.44)$$

$J(x(k))$ ist die Jakobi Matrix in $x(k)$, ihre Zeilen die Gradienten der Zielfunktionen im Punkt $x(k)$ und $w(k)$ stellt einen n -dimensionalen Gewichtsvektor dar. Somit ist $J^T(x(k))w(k)$ mit $\sum_{i=1}^n w_i(k) = 1, w_i(k) \geq 0 \quad \forall i = 1, \dots, n$, eine Konvexkombination von Gradienten, die als negative Suchrichtung verwendet wird. Der Skalar $\beta(k) > 0$ steht für die Schrittweite und $\gamma(k) \in \mathbb{R}^m$ fungiert als Korrektur der Suchrichtung, hervorgerufen durch einen Strafterm, der die Verletzung von Nebenbedingungen zu verhindern sucht.

Für die Festlegung von $\beta(k+1)$ schlagen die Erfinder verschiedene Heuristiken vor, die auf die Leistung des Algorithmus in der Voriteration reagieren und wählen $w(k+1)$ stochastisch.

In jedem neuen Punkt $x(k+1)$ werden Zielfunktionswerte und Gradienten berechnet. Haben in einer nicht auf dem Rand von R liegenden Alternative "zwei Gradienten von Zielfunktionen entgegengesetzte Vorzeichen" (das könnte heißen $\nabla f_i(x(k+1)) = -\nabla f_j(x(k+1)), i, j \in \{1, \dots, n\}$) oder die beiden Gradienten könnten z.B. auch lauten: $(1, 1, \dots, 1)$ und $(-3, -2, \dots, -5)$) oder liegt $x(k+1)$ auf dem Rand von R , versteht man dies als einen Hinweis, daß $x(k+1)$ ein lokales Paretooptimum sein könnte. Dieser Verdacht erhärtet sich, wenn man im Vergleich feststellt, daß $x(k+1)$ von keiner bisher entdeckten und gespeicherten Alternative dominiert wird. Sollten diese Kriterien eintreten und $x(k+1)$ bzw. $F(k+1)$ liegt genügend weit entfernt von jedem bisher gefundenen und gespeicherten $x \in R$ bzw. $F(x) \in F(R)$ (verwendete Metrik noch zu bestimmen), so wird $x(k+1)$ ebenfalls gespeichert.

Den Suchvorgang zu beenden, wenn nach einer bestimmten Anzahl von Iterationen kein neuer Punkt mehr der Speicherung für wert befunden werden konnte, ist eine Möglichkeit für ein Stoppkriterium.

Ein Schwachpunkt der Methode liegt in der unklaren Formulierung des Kriteriums, das anzeigen soll, ob eine lokal paretooptimale Alternative gefunden wurde. Sollte gemeint sein, daß zwei Gradienten genau in entgegengesetzte Richtung zeigen, so ist dies zwar ein hinreichendes, jedoch kein notwendiges Kriterium für (lokale) Paretooptima im Inneren von R und viele wirkliche (lokale) Paretooptima werden vom Verfahren eventuell übersehen. Sollte jedoch gemeint sein, daß für zwei Gradienten jeder positiven Komponente des einen Gradienten eine negative im anderen gegenübersteht (egal welchen Wertes) und umgekehrt, so

ist das Kriterium für (lokale) Effizienz im Inneren von R weder hinreichend noch notwendig. Um die Gefahr, eine dominierte Alternative einzusammeln, zu reduzieren, wird immerhin der Dominanztest mit bereits gespeicherten Alternativen durchgeführt.

Zusammenfassung:

Vorteile:

- Obwohl nicht anwendbar auf diskret-kontinuierliche Vektoroptimierungsprobleme, also solche mit diskret-kontinuierlichem R , ist die Methode allein deshalb der Erwähnung wert, weil sie Gradienteninformation einsetzt, um immer bessere und schließlich pareto-optimale Punkte zu erreichen.
- Ein weiterer geschickter Schachzug ist, den Suchraum möglichst in jede Richtung hin auszuloten, indem man mehrere Suchen von den verschiedenen "extremalen" Regionen aus startet.
- Das Verfahren bleibt je Iterationsfolge nicht in lokalen Paretooptima hängen.

Nachteile:

- Das zu lösende Mehrzielproblem muß aus Zielfunktionen bestehen, die (mindestens) differenzierbar sind.
- Das Kriterium, das einen gefundenen Punkt als eventuell lokal paretooptimal einstuft, ist sehr zweifelhaft, nämlich je nach Interpretation nicht notwendig oder weder hinreichend noch notwendig für (lokale) Paretooptima im Inneren von R .
- Die Wahl eines passenden Maßes für "genug Entfernung" zwischen zwei entdeckten Lösungsalternativen könnte schwierig sein.

2.3.2.2 Stochastisches Suchen nach Timmel

In [63] betont Timmel, daß die meisten Methoden der Mehrzieloptimierung nur Probleme mit bestimmten, sehr einschränkenden Eigenschaften wie z.B.

- Konvexität von R oder/und $F(R) + (\mathbb{R}_+^n \cup \{0\})$ oder/und $F(R)$,
- Konvexität und Linearität der einzelnen Zielfunktionen

lösen können. Selbst wenn sie fähig sind, jede effiziente Alternative eines Problems ohne diese Eigenschaften grundsätzlich zu erreichen, laufen deterministische Methoden immer Gefahr, in einem lokal paretooptimalen Punkt hängen zu bleiben, ohne jemals zu einem globalen Paretooptimum gelangen zu können. Deshalb schlägt Timmel eine Prozedur vor, die stochastisch Alternativenfolgen $\{x^k\}_{k \in \mathbb{N}_0} \subseteq R$ mit $F(x^k) \prec F(x^{k-1})$ oder zumindest $F(x^k) \asymp F(x^{k-1})$, $k \in \mathbb{N}$, erzeugt. Jede Iteration produziert **relativ effiziente** Alternativen; das sind Alternativen, die von keiner der bisher gefundenen Alternativen dominiert werden. Daher sind die Ergebnispunkte jeder Iteration mindestens so "gut" wie diejenigen der Voriteration. Falls die Zielfunktionen stetig differenzierbar sind und $m \geq n$ ist, kann Timmel beweisen, daß die Zielwertvektoren der

unter Gleichverteilung aus R ausgewählten Startpunkte sich um die Zielwertvektoren der Punkte \bar{x} , die die notwendigen (nicht hinreichenden) Bedingungen für Stationarität im Inneren von R erfüllen (alle Paretooptima sind auch stationäre Punkte), konzentrieren:

$$D_{i_1, \dots, i_n}(\bar{x}) = \det \left(\frac{\partial (f_1, \dots, f_n)}{\partial (x_{i_1}, \dots, x_{i_n})} \Big|_{x=\bar{x}} \right) = 0 \quad \forall \quad \{i_1, \dots, i_n\} \subseteq \{1, \dots, m\}, m \geq n. \quad (2.45)$$

Timmels Verfahren versucht die Effizienzmenge in $F(R)$ oder wenigstens eine Teilmenge davon zu approximieren. Zu jedem Paretooptimum $\bar{x} \in R$ soll ein $\hat{x} \in R$ gefunden werden, das $\|F(\bar{x}) - F(\hat{x})\|_2 < \varepsilon$ für ein beliebiges $\varepsilon > 0$ erfüllt.

Timmel verlangt für seine Methode eine kompakte Alternativenmenge und stetige Zielfunktionen, womit schon einmal die Existenz von Paretooptima garantiert ist. Damit das theoretische Fundament des Verfahrens nicht ins Wanken gerät, sollten die Zielfunktionen auch stetig differenzierbar sein. Laut Timmel kann die Methode auch für die Anwendung auf diskrete Vektoroptimierungsprobleme angepaßt werden, soweit dort die Zielwertvektoren nahe beisammenliegender Alternativen ebenfalls nahe beieinanderliegen (die entsprechenden Metriken in \mathbb{R}^m und \mathbb{R}^n sind noch festzulegen), jedoch lassen sich dann seine Aussagen über Konvergenz und Approximationsgüte nicht mehr beweisen.

Die Generierung von Startpunkten:

Man betrachte einen Vektor $Z = (Z_1, \dots, Z_m)$ von Zufallsvariablen mit den Realisierungen $z = (z_1, \dots, z_m)$, die alle möglichen Alternativen aus R umfassen. Sei $m(R)$ ein Maß auf R . Z sei gleichverteilt auf R mit der Verteilungsdichte

$$h(z) = \begin{cases} \frac{1}{m(R)}, & \text{falls } z \in R, \\ 0, & \text{falls } z \notin R. \end{cases} \quad (2.46)$$

Ist f_i für alle $i = 1, \dots, n$ stetig differenzierbar und $m \geq n$, so erfüllt die Verteilungsdichte $\bar{h}(F(Z))$ des Zufallsvektors $F(Z)$ die Ungleichung

$$\bar{h}(F(z)) \geq \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} \frac{h(z)}{|D_{i_1, \dots, i_n}(z)|} dz_{i_{n+1}} \dots dz_{i_m} \quad (2.47)$$

$$\forall \{i_1, \dots, i_n\} \subseteq \{1, \dots, m\}.$$

Denkt man an die notwendigen Bedingungen für stationäre Punkte im Inneren von R , so erkennt man, daß für effiziente Alternativen $\bar{z} \in R \setminus \delta(R)$ ($R \setminus \delta(R)$ bezeichnet das Innere von R) $\bar{h}(F(\bar{z})) = \infty$ ist. Werden also die Startpunkte unter Gleichverteilung aus R ausgesucht, dann häufen sich ihre Zielwertvektoren um Punkte herum an, welche Bilder von Alternativen sind, die (2.45) erfüllen. Nachdem man aus der erzeugten Startpunktemenge dominierte Alternativen aussortiert hat, schwächt sich auch die ungewollte Konzentration in $F(R)$ um die Zielwertvektoren von stationären Punkten, die nicht effizient sind, ab. Das Ergebnis ist eine erste Approximation $App_0 := \{x_1^0, \dots, x_{r_0}^0\} \subseteq R, r_0 \in \mathbb{N}$, der Effizienzmenge in R .

Erzeugung einer neuen Approximation in Iteration $k \geq 1$:

Zu jedem $x_j^{k-1}, j = 1, \dots, r_{k-1}$, wird in Zufallsexperimenten ein $y \in R$ gesucht, für das entweder $F(y) \prec F(x_j^{k-1})$ oder $F(y) \succ F(x_j^{k-1})$ gilt.

- Die Suchrichtung $s_j^k \in \mathbb{R}^m$ wird für alle $j = 1, \dots, r_{k-1}$ zufällig erzeugt, wobei es einen Unterschied macht, ob man $\nabla f_i(x_j^{k-1}), i = 1, \dots, n$, berechnen oder annähern kann. Ist die Berechnung möglich, dann definiert sich die Suchrichtung aus

$$s_j^k := - \frac{\sum_{i=1}^n b_i \frac{\nabla f_i(x_j^{k-1})}{\|\nabla f_i(x_j^{k-1})\|_2}}{\left\| \sum_{i=1}^n b_i \frac{\nabla f_i(x_j^{k-1})}{\|\nabla f_i(x_j^{k-1})\|_2} \right\|_2}, \quad (2.48)$$

wobei die b_i Realisierungen voneinander stochastisch unabhängiger Zufallsvariablen mit Gleichverteilung in $[0; 1]$ sind.

Bei allen Punkten $x_j^{k-1}, j = 1, \dots, r_{k-1}$, im Inneren von R liegt in Richtung s_k^j sicherlich mindestens ein Punkt, der nicht von x_j^{k-1} dominiert wird.

Sind die Gradienten der Zielfunktion nicht greifbar, so wird s_k^j einfach unter Gleichverteilung aus $\{s \in \mathbb{R}^m \mid \|s\|_2 = 1\}$ gewählt.

- Für die Schrittweite t_j^k empfiehlt Timmel eine vorsichtige Reduzierung von Iteration zu Iteration, die **Dvoretzkys Konvergenzbedingungen** folgt:

$$\lim_{k \rightarrow \infty} t_j^k = 0, \quad \sum_{k=1}^{\infty} t_j^k = \infty, \quad \sum_{k=1}^{\infty} (t_j^k)^2 < \infty. \quad (2.49)$$

- Sollte es kein $t_j^k > 0$ mit $x_j^{k-1} + t_j^k s_j^k \in R$ geben, schlägt Timmel drei Vorgehensweisen vor:

- keine weitere Suche von x_j^{k-1} aus, in welche Richtung auch immer,
- Spiegelung von $x_j^{k-1} + t_j^k s_j^k$ auf den Rand von R , falls möglich,
- Wiederholung der Zufallswahl von s_j^k , bis es ein $t_j^k > 0$ gibt mit $x_j^{k-1} + t_j^k s_j^k \in R$.

Die letzten beiden Möglichkeiten sind vorzuziehen, da durch sie von Iteration zu Iteration auch mehr und mehr Alternativen auf dem Rand von R in die jeweilige Approximation der Effizienzmenge in R gelangen, da doch der Rand von R bei der Startpunktwahl eventuell vernachlässigt wurde.

- Falls $x_j^k = x_j^{k-1} + t_j^k s_j^k$ die Alternative x_j^{k-1} dominiert, ist der Punkt ein Kandidat für die neue Approximation der Effizienzmenge in R , falls $F(x_j^k) \preceq F(x_j^{k-1})$, sind beide Punkte x_j^{k-1} und x_j^k solche Kandidaten. Auch der Fall $F(x_j^k) \succ F(x_j^{k-1})$ ist möglich, falls die Gradienten der Zielfunktionen nicht zur Verfügung stehen - dann wird entweder x_j^k einfach nicht weiter beachtet oder die Zufallsgenerierung von s_j^k wird so lange wiederholt, bis ein passendes x_j^k aufgefunden ist. Wenn von jedem Punkt in App_{k-1} aus ein Suchschritt ausgeführt wurde, werden aus der Vereinigungsmenge der in dieser Iteration neu hinzugefundenen Alternativen und der Menge App_{k-1} alle dominierten Punkte aussortiert. Was übrig bleibt bildet die neue Approximation $App_k := \{x_1^k, \dots, x_{r_k}^k\}, r_k \in \mathbb{N}$, eine Menge, die "mindestens so gut wie" App_{k-1} ist.

Stopp-Kriterium:

Dies festzulegen bleibt dem Anwender. Man kann die Suchprozedur z.B. enden lassen, wenn für ein $k \in \mathbb{N}_0$ die Mächtigkeit $|App_k|$ eine bestimmte Größe überschreitet oder eine maximal erlaubte Zahl von Iterationen durchgeführt wurde. In [64] wird eine noch etwas kompliziertere Variante vorgeführt.

Timmel beweist, daß bei stetig differenzierbaren Zielfunktionen seine Methode stochastisch konvergiert, in dem Sinne, daß für alle $\varepsilon > 0, \varepsilon \in \mathbb{R}^n$, und für jeden nichtdominierten Punkt $F(\bar{x}) \in F(R)$ gilt

$$\lim_{k \rightarrow \infty} P(\exists x \in App_k : \|F(x) - F(\bar{x})\|_2 < \varepsilon) = 1. \quad (2.50)$$

Zusammenfassung:

Vorteile:

- Timmels stochastisches Suchen benötigt keine Konvexität von R oder/und $F(R) + (\mathbb{R}_+^n \cup \{0\})$ oder/und $F(R)$ oder Linearität oder Konvexität der Zielfunktionen.
- Es nutzt Gradienteninformation der Zielfunktionen geschickt aus.
- Die Stochastik in der Suche verhindert, daß der Algorithmus in lokalen Paretooptima hängen bleibt.

Nachteile:

- Das Verfahren konzentriert sich auf Probleme mit stetigen Zielfunktionen. Gute Leistung ist nur beweisbar, wenn die Zielfunktionen sogar stetig differenzierbar sind. Bei Ausdehnung der Anwendung dieser Methode auf diskrete Mehrzielprobleme R dürfte es bei der Bestimmung von Suchrichtungen und Schrittweiten Schwierigkeiten geben.

Im Detail kann man die Methode in [61], [62], [63] und [64] nachlesen.

2.3.2.3 Ein stochastisches Verfahren zur Lösung von Vektoroptimierungsproblemen ohne Nebenbedingungen

Dieses Verfahren versucht, alle oder eine große Zahl von Paretooptima zu berechnen. Die Suchstrategie ist eine geschickte Mischung aus Ausnutzung von Gradienteninformation aller Zielfunktionen und stochastischer Auslotung von R . Detaillierte Information zu dieser mathematisch anspruchsvollen Methode findet der interessierte Leser in [52].

Die vorzustellende Suchstrategie wurde konstruiert, um Mehrzieloptimierungsprobleme mit $R := \mathbb{R}^m$ (unbeschränkte Alternativenmengen) und zweimal stetig differenzierbaren Zielfunktionen $f_i : R \mapsto \mathbb{R}, x \mapsto f_i(x), i = 1, \dots, n$, zu bearbeiten:

$$''\min''_{x \in \mathbb{R}^m} F(x). \quad (2.51)$$

Motivation:

In der skalaren Optimierung kann das Suchverhalten einer großen Klasse von numerischen Methoden für Minimierungsprobleme ohne Nebenbedingungen mit zweimal stetig differenzierbarer Zielfunktion $f : \mathbb{R}^m \mapsto \mathbb{R}, x \mapsto f(x)$, als Prozedur beschrieben werden, die numerisch folgende Anfangswertprobleme löst:

$$\dot{x}(t) = -\nabla f(x(t)), x(0) = 0, \quad (2.52)$$

wobei $\nabla f(x)$ den Gradienten von f in x bezeichnet. Die Lösung $x : [0, \infty[\mapsto \mathbb{R}^m$ von (2.52) besitzt die schöne Eigenschaft

$$f(x(s)) > f(x(t)) \quad \forall 0 \leq s < t < \infty, \quad (2.53)$$

d.h. $x(t)$ liegt umso näher am (lokalen) Minimierer, je größer t ist.

Die Mehrzieloptimierungstechnik, die im Folgenden skizziert wird, verallgemeinert diesen Ansatz auf Vektoroptimierungsprobleme ohne Nebenbedingungen mit der Mehrzielfunktion $F : \mathbb{R}^m \mapsto \mathbb{R}^n, x \mapsto F(x)$.

Deterministische Seite des Algorithmus

Für die Konstruktion eines Anfangswertproblems, dessen eindeutige Lösung eine Kurve $x : [0, \infty[\mapsto \mathbb{R}^m$ dominierender Punkte ist, für die also gilt

$$F(x(s)) \succ F(x(t)) \quad \forall 0 \leq s < t < \infty, \quad (2.54)$$

betrachte man für jedes $x \in \mathbb{R}^m$ folgendes quadratisches Optimierungsproblem:

$$\begin{aligned} \min_{\alpha \in \mathbb{R}^n} & \left\| \sum_{i=1}^n \alpha_i \nabla f_i(x) \right\|_2^2, \\ & \alpha_i \geq 0, i = 1, \dots, n, \sum_{i=1}^n \alpha_i = 1, \end{aligned} \quad (2.55)$$

wobei $\nabla f_i(x)$ den Gradienten der i -ten Zielfunktion bezeichne. Für alle $x \in R$ definiert jeder globale Minimierer $\hat{\alpha}$ von Problem (2.55) die Funktion

$$q : \mathbb{R}^m \mapsto \mathbb{R}^n, x \mapsto \sum_{i=1}^n \hat{\alpha}_i \nabla f_i(x). \quad (2.56)$$

Es kann gezeigt werden, daß entweder $q(x) = 0$ oder $-q(x)$ in x für alle Zielfunktionen eine Abstiegsrichtung ist und daß q außerdem in jedem $x \in \mathbb{R}^m$ einer lokalen Lipschitzbedingung mit Konstante $L_x \in \mathbb{R}_0^+$ genügt.

Diese Eigenschaften von q ermöglichen es, das Anfangswertproblem (2.52) für skalare Optimierungsprobleme zu folgendem Anfangswertproblem zu verallgemeinern:

$$\dot{x}(t) = -q(x(t)), x(0) = x_0. \quad (2.57)$$

Theorem 2.3.4. *Man betrachte Problem (2.51) und das dazugehörige Anfangswertproblem (2.57) mit $q(x_0) \neq 0$. Sollte die Menge Dom_{x_0} aller Alternativen $\in \mathbb{R}^m$, die x_0 dominieren, beschränkt sein, gibt es eine eindeutige Lösung $x : [0, \infty[\mapsto \mathbb{R}^m$ für (2.57) mit*

$$x(s) \succ x(t) \quad \forall 0 \leq s < t < \infty. \quad (2.58)$$

Stochastische Seite des Algorithmus

Zu der in (2.56) definierten Funktion q existiere ein $\varepsilon > 0$, so daß gilt

$$\begin{aligned} x^T q(x) &\geq \frac{1+n\varepsilon^2}{2} \max(1, \|q(x)\|_2) \\ \forall x \in \mathbb{R}^m \setminus \{x \in \mathbb{R}^m \mid \|x\|_2 \leq r\}, r > 0. \end{aligned} \quad (2.59)$$

Diese Ungleichung beschreibt das Verhalten von q außerhalb einer Kugel mit Radius r , deren Existenz einfach vorausgesetzt wird. Erfüllt q (2.59) für ein $\varepsilon > 0$ und ein $r > 0$, so stammt es von einem Mehrzieloptimierungsproblem, das paretooptimale Lösungen besitzt.

Alle Zufallsvariablen, die in den weiteren Darstellungen vorkommen, leben auf einem bestimmten Wahrscheinlichkeitsraum $(\Omega, \mathfrak{B}(\Omega), \mathfrak{W})$, der in [52] genauer definiert ist.

Als nächstes wird eine Klasse von stochastischen Itô-Differentialgleichungen definiert. $\{B_t\}_{t \geq 0}$ bezeichne eine n -dimensionale Brownsche Bewegung auf Ω , ω sei ein Element von Ω , und q sei definiert durch (2.56) und erfülle (2.59) für ein $\varepsilon > 0$ und ein $r > 0$:

$$dX_t = -q(X_t)dt + \varepsilon dB_t, X_0 = x_0, \quad (2.60)$$

wobei $x_0 \in \mathbb{R}^m$ sei und $\{X_t\}_{t \geq 0}$ ein stochastischer Prozeß auf Ω sei.

Theorem 2.3.5. Für alle $x_0 \in \mathbb{R}^m$ und alle $\varepsilon > 0$, für die q für ein $r > 0$ Ungleichung (2.59) erfüllt, erhält man, daß

- ein eindeutig bestimmter stochastischer Prozeß $\{X_t\}_{t \geq 0}$ existiert, der (2.60) löst,
- alle Pfade von $\{X_t\}_{t \geq 0}$ stetig sind,
- $X_0 \equiv x_0$ gilt.

Stochastische Stoppzeiten $s_{\bar{x}, p}$ mit $p > 0$ werden die Fähigkeit des stochastischen Prozesses $\{X_t\}_{t \geq 0}$, der die Lösung von (2.60) ist, ein globales Paretooptimum $\bar{x} \in \mathbb{R}^m$ von Problem (2.51) zu erreichen, demonstrieren: $s_{\bar{x}, p} : \Omega \mapsto \mathbb{R}$,

$$\omega \mapsto \begin{cases} \inf\{t \geq 0 \mid \|X_t(\omega) - \bar{x}\|_2 \leq p\}, & \text{falls } \{t \geq 0 \mid \|X_t(\omega) - \bar{x}\|_2 \leq p\} \neq \emptyset, \\ \infty, & \text{sonst.} \end{cases} \quad (2.61)$$

Theorem 2.3.6. Bei beliebigem $x_0 \in \mathbb{R}^m$ erhält man für die Lösung $\{X_t\}_{t \geq 0}$ von (2.60) mit einem $\varepsilon > 0$, so daß q für ein $r > 0$ Ungleichung (2.59) erfüllt, und für jede paretooptimale Lösung $\bar{x} \in \mathbb{R}^m$ von (2.51) und jedes $p > 0$:

- $\mathfrak{W}(\{\omega \mid s_{\bar{x}, p}(\omega) < \infty\}) = 1$ (\mathfrak{W} Wiener Maß).
- $E(s_{\bar{x}, p}) < \infty$ ($E(\cdot)$ bezeichnet den Erwartungswert).
- Der in Theorem 2.3.5 definierte stochastische Prozeß $\{X_t\}_{t \geq 0}$ konvergiert in Verteilung gegen eine Zufallsvariable $X : \Omega \mapsto \mathbb{R}^m$ mit $E(q(X)) = 0$.

In Worten ausgedrückt bedeuten die Aussagen von Theorem 2.3.6, daß

- \mathcal{W} -fast alle Pfade von $\{X_t\}_{t \geq 0}$ jede Kugel in \mathbb{R}^m mit Radius $p > 0$ um jeden paretooptimalen Punkt \bar{x} in endlicher Zeit unabhängig vom Anfangswert/Startpunkt $x_0 \in \mathbb{R}^m$ treffen,
- der Erwartungswert von $s_{\bar{x}, p}$ endlich ist.

\mathcal{W} -fast jede stochastische Suche nach Paretooptima, die je Iteration $t \in \mathbb{N}_0$ eine Realisierung $x_t = X_t(\omega) \in \mathbb{R}^m$ des m -dimensionalen Zufallsvektors X_t aufsucht, wird in endlicher Zeit an jedes Paretooptimum \bar{x} in \mathbb{R}^m bis auf den Abstand $p > 0$ (ein beliebiger euklidischer Abstand) oder näher herankommen.

Die Bestimmung des Wertes von $\varepsilon > 0$ in (2.60) ist eine schwierige und wichtige Aufgabe. ε stimmt den Einfluß der Kurve dominierender Punkte

$$X_t^{x_0} = x_0 - \int_0^t q(X_\tau^{x_0}) d\tau \quad (2.62)$$

und den der Zufallssuche, die für $\omega \in \Omega$ dem Pfad der Folge aus normalverteilten Zufallsvektoren

$$X_t^{x_0}(\omega) = x_0 + B_t(\omega) - B_0(\omega) \quad (2.63)$$

mit wachsender Varianz folgt, im Gesamtsuchverhalten aufeinander ab. Bei zu kleinem ε dominiert für festes ω die Kurve (2.62) das Suchverhalten zu stark, was zur Folge hat, daß die Pfade von $\{X_t^{x_0}\}_{t \geq 0}$ zu lange Zeit nahe irgendeiner (lokal) paretooptimalen Alternative von Problem (2.51) verlaufen. Im Falle eines zu großen ε beraubt man die Pfade von $\{X_t^{x_0}\}_{t \geq 0}$ ihrer Tendenz zu Paretooptima, wie sie in Theorem 2.3.6 definiert ist. Die Wahl von ε hängt sehr stark von F und der verwendeten Skalierung ab.

Die numerische Berechnung der Pfade von $\{X_t^{x_0}\}_{t \geq 0}$ leitet sich von der Euler Methode her, einem Standardansatz der numerischen Analysis zur Lösung von gewöhnlichen Differentialgleichungen.

Die eben vorgestellte stochastische Methode zur Lösung von Vektoroptimierungsproblemen ohne Nebenbedingungen wurde bereits unter leichten Abwandlungen auch erfolgreich auf Probleme mit Nebenbedingungen angewendet.

Zusammenfassung:

Vorteile:

- \mathcal{W} -fast alle stochastisch erzeugten Suchwege der beschriebenen Technik in $R := \mathbb{R}^m$ kommen in endlicher Zeit beliebig nahe an jedes Paretooptimum in R heran.
- Es besteht (bei vernünftiger Wahl von ε) keine Gefahr, daß das Verfahren in lokalen Paretooptima hängen bleibt.
- $F(R) + (\mathbb{R}_+^n \cup \{0\})$ muß nicht konvex sein.

Nachteile:

- Die Methode ist nicht für diskret-kontinuierliche Mehrzielprobleme verwendbar, kann aber durchaus als Werkzeug des Verfahrens, das in dieser Arbeit entwickelt wird, verwendet werden.

2.3.3 Diskretes stochastisches Suchen

Alle ab jetzt noch beschriebenen stochastischen Generierungstechniken basieren auf **Genetischen Algorithmen** (GAs). Um nicht zu sehr vom eigentlichen Thema abzuschweifen, verzichtet diese Arbeit auf eine Einführung in die Grundlagen der Evolutionären Algorithmen. Der interessierte Leser kann Antworten auf seine drängendsten Fragen in [23] erwarten. Allerdings soll dargelegt werden, warum die in der Praxis oft erfolgreichen GA-Instrumente nicht zur Lösung der Klasse von diskret-kontinuierlichen Vektoroptimierungsproblemen, die diese Arbeit anstrebt, verwendet wurden.

Wenn Genetische Algorithmen R bearbeiten, müssen die Alternativen irgendwie als Vektoren $(s_1, \dots, s_p) \in \mathbb{R}^p, p \in \mathbb{N}$, codiert werden, die man **Chromosomen** nennt. Auch wenn Chromosomen lediglich Repräsentanten von Alternativen sind, dürfen Chromosom und Alternative ruhig synonym verwendet werden. Weitere Begriffe für Alternativen sind hier Punkt und Individuum - Letzteres in Anlehnung an die Biologie.

Gene sind Teile bzw. Abschnitte von Chromosomen, die normalerweise einen einzelnen Parameter codieren, in unserem Fall z.B. den Wert der i -ten Komponente einer Alternative, $i \in \{1, \dots, m\}$. **Schemata** sind bestimmte genetische Muster, d.h. Kombinationen von Genwerten, in den Chromosomen, die neben reellen Zahlen auch Symbole $*$ für Unentschiedenheit beinhalten können. Man betrachte als Beispiel das Schema

$$(*, *, \dots, *, \bar{s}_{i_1}, *, \bar{s}_{i_2}, \bar{s}_{i_3}, *, \dots, *, \bar{s}_{i_4}, *, \dots, *), \quad (2.64)$$

$$i_1 < i_2 < i_3 < i_4 \in \{1, \dots, p\}. \quad (2.65)$$

$(\hat{s}_1, \dots, \hat{s}_{i_1-1}, \bar{s}_{i_1}, \hat{s}_{i_1+1}, \bar{s}_{i_2}, \bar{s}_{i_3}, \hat{s}_{i_3+1}, \dots, \hat{s}_{i_4-1}, \bar{s}_{i_4}, \hat{s}_{i_4+1}, \dots, \hat{s}_p)$ paßt in dieses Schema. Es bezeichne $o(H)$ die Anzahl der ein Schema H definierenden Komponenten und $\rho(H)$ den Abstand (die Anzahl von Positionen) zwischen erster und letzter definierender Komponente von H . In unserem Beispiel ist $o(H) = 4$ und $\rho(H) = i_4 - i_1$. Je Iteration, hier Generation, erzeugt/findet ein evolutionärer Suchprozeß eine Menge von Individuen, die Population heißt. Einige Individuen der Population werden aufgrund besonderer Qualitäten stochastisch zu sogenannten Eltern bestimmt. Aus diesen Eltern werden durch bestimmte Fortpflanzungsmechanismen neue Individuen erzeugt, also die in der nächsten Generation betrachteten Punkte. Genetische Algorithmen reproduzieren Individuen in der Regel proportional zu ihrer Güte, die in Form eines skalaren **Fitness Wertes** ausgedrückt wird. In einem gewöhnlichen Verfahren mit den Fortpflanzungsmechanismen **Reproduktion**, **Kreuzung** und **Permutation** pflanzen sich Schemata H mit kleinem $\rho(H)$, kleinem $o(H)$ und guter durchschnittlicher Qualität (die sich als Durchschnitt der Fitness Werte aller Individuen, die in das Schema H passen, berechnet) von Generation zu Generation exponentiell fort. Schemata, die diese drei genannten Eigenschaften erfüllen, nennt man **Building Blocks**, die getroffene Aussage **Building Block Hypothese**.

Außer dieser Building Block Hypothese kann man bei Genetischen Algorithmen allerdings im allgemeinen keine Konvergenzaussage machen. Obwohl eine besondere Förderung der diesen Building Blocks entsprechenden Individuen plausibel erscheint, muß man bedenken, daß eventuell nicht jede sinnvolle Alternative des Mehrzielproblems sich als Individuum codieren läßt, das in Building Blocks paßt. Darüber hinaus ist das Überleben von Building Blocks teilweise gefährdet, da z.B. Kreuzung Individuen erzeugen kann, die zu Schemata

mit großem $\rho(H)$ und schlechter durchschnittlicher Qualität gehören, obwohl die Eltern in Building Blocks passen.

2.3.3.1 VEGA

Schaffer ([51]) erweiterte den einfachen Genetischen Algorithmus zu einem **Vektorwertigen Genetischen Algorithmus**, kurz **VEGA**.

In jeder Population von $N \in \mathbb{N}$ Individuen einer Generation (das sind die in einer Iteration gefundenen/erzeugten Individuen) kommt es zur Bildung so vieler Subpopulationen, wie Ziele vorhanden sind. Für die i -te Subpopulation, $i = 1, \dots, n$, werden N/n Individuen der Population proportional zu ihrem Leistungsniveau in Ziel i ausgewählt: Je kleiner der Zielwert, desto wahrscheinlicher die Wahl des Individuums. Danach werden alle n Subpopulationen zusammengemischt, woraus sich eine neue Population der Mächtigkeit N ergibt. Auf deren Individuen wendet der Genetische Algorithmus auf die übliche Weise Kreuzung und Mutationsoperatoren an. Dabei erhalten (schwach) effiziente Alternativen unter ihnen, die Minimierer für eines der Ziele sind, die beste Chance, ihre Erbinformation als Eltern der nächsten Generation zu übertragen. Aber ist dies wirklich sinnvoll?

Die Absicht hinter dieser Vorgehensweise ist, eine gute (näherungsweise) Darstellung der Effizienzmenge in R zu erarbeiten, indem man alle Ziele gleichberechtigt behandelt - es gibt ja je Ziel $i \in \{1, \dots, n\}$ genau N/n Eltern in jeder Generation, die in diesem Ziel die relativ beste Leistung bringen. Keinerlei Präferenzen drängen sich auf. Dennoch bleibt die Gefahr, daß einige Gebiete der Effizienzmenge vernachlässigt werden: Nur Individuen, die in wenigstens einem der Ziele brillieren, werden als Vererber gefördert. (Effiziente) Alternativen, die sich in keinem der Kriterien besonders auszeichnen, bleiben auf der Strecke.

Schaffer hat dieses Problem erkannt und versucht es mit verschiedenen Heuristiken zu umgehen. Er schlägt z.B. vor, den innerhalb der Generation nichtdominierten Alternativen heuristisch die Vererbung zu erleichtern, um effiziente Punkte mit mittelmäßigem Leistungsniveau in jedem Ziel am Leben zu erhalten. Eine weitere Möglichkeit ist, Individuen verschiedener Spezies (definiert durch gute Leistung in einem der Ziele) miteinander zu paaren, was durch bestimmte heuristische Paarungsmechanismen angeregt werden soll, die die herkömmliche stochastische Partnerwahl der GA ersetzen.

2.3.3.2 Evolutionäre Strategien

Kursawe formuliert in [41] eine Mehrziel-Version der Evolutionären Strategien (siehe z.B. [53]).

In jeder Generation besteht die Selektion aus so vielen Schritten, wie es Ziele gibt. In einem n -dimensionalen Vektor $r \in \mathbb{R}^n$ ist für jedes Ziel die Wahrscheinlichkeit festgehalten, mit der es Maßstab für die Elimination einiger Alternativen der aktuellen Generation sein kann. n -mal je Generation bestimmt ein Zufallsexperiment nach den Wahrscheinlichkeiten in r ein Ziel $i \in \{1, \dots, n\}$. Dann wird eine bestimmte Anzahl von Individuen mit den relativ schlechtesten Leistungen in Ziel i innerhalb der gegenwärtigen Population "weggeworfen". Nachdem n solcher Selektionsschritte durchgeführt wurden, sind von der Population dieser Generation lediglich die Individuen übrig, die Eltern für die nächste Generation werden dürfen.

Die besten Alternativen, die während eines gesamten Suchlaufes der Evolutionären Strategien erzeugt/gefunden wurden, bilden schließlich die Approximation der Effizienzmenge in R .

Sicher fördert der Algorithmus das Auffinden von Alternativen, die sich in keinem der Ziele schlecht schlagen; unter ihnen befinden sich auch viele Paretooptima. r kann allerdings Präferenzen für einige Ziele zum Ausdruck bringen, falls die einzelnen Komponenten des Vektors sich stark unterscheiden. Eine abgewandelte Version dieser Technik erlaubt es, r im Laufe des Suchvorganges stochastisch zu verändern, so daß jeglicher Präferenzeinfluß verhindert wird.

2.3.3.3 Wichtungsansatz

Hajela und Lin ([28]) verfolgen mit ihrem Genetischen Algorithmus das Ziel, in einem einzigen Lauf des Wichtungsansatzes eine Menge effizienter Alternativen zu erzeugen, die Minimierer verschiedener gewichteter Zielfunktionssummen sind. Die Idee hinter dem Ansatz ist die gleiche wie bei der Wichtungsmethode aus Abschnitt 2.3.1.1. Die Variation der Gewichte geschieht hier stochastisch, allerdings wird dabei nur eine vorher festgelegte Teilmenge aller grundsätzlich möglichen Gewichtskombinationen berücksichtigt. Die Problematik, diese Teilmenge geschickt zusammenzustellen, bleibt von den Variationsschwierigkeiten übrig. Nach wie vor werden Paretooptima aus R mit Zielwertvektoren im nichtkonvexen Teil von $F(R) + (\mathbb{R}_+^n \cup \{0\})$ von der Suchprozedur vernachlässigt.

Es sei N die Anzahl verschiedener Gewichtskombinationen, für die Paretooptima der skalaren Ersatzprobleme gefunden werden sollen.

Ansatz basierend auf gewichteten Zielwertsummen

In jeder Generation gehen aus der Population der Größe P genau N gleichmächtige Subpopulationen mit P/N Individuen hervor. Dafür gibt es zwei Varianten:

- Subpopulation $s_w, s_w \in \{1, \dots, N\}$, wird mit Alternativen der aktuellen Generation aufgefüllt, deren negative gewichtete Zielwertsummen mit Gewichtsart s_w zu den niedrigsten in dieser Generation gehören. Individuen können dabei gleichzeitig mehreren Subpopulationen angehören.
- Ein ausschließenderes Prinzip läßt ein Individuum einer Subpopulation nicht mehr als Mitglied einer weiteren zu. Reproduktion, Kreuzung und Mutation ereignen sich in jeder Subpopulation separat. Folglich kann zwischen Individuen verschiedener Subpopulationen keine Paarung stattfinden.

In einem alternativen Ansatz formen Hajela und Lin Individuen, die nicht nur eine Codierung der repräsentierten Alternative enthalten, sondern auch eine Codierung der Gewichtsvektoren, die bei der Skalarisierungsmethode der Summierung gewichteter Zielfunktionen verwendet werden. Die Belegung der Variablen $s_w \in \{1, \dots, N\}$ informiert darüber, welcher Gewichtsvektor für die Alternative, die durch den p -Vektor $(s_1, \dots, s_p) \in \mathbb{R}^p$ codiert wird, relevant ist. (s_w, s_1, \dots, s_p) ist das gesamte Individuum. Der Fitness Wert eines Individuums ist in diesem Verfahren die negative gewichtete Summe seiner Zielfunktionswerte.

Ansatz basierend auf Sharing Funktionen

Hier gibt es keine Subpopulationen zur Förderung besonders guter Individuen in Hinblick auf die Minimierung N verschiedener gewichteter Zielfunktionssummen. Um die Konzentration von Individuen künftiger Generationen um Punkte, die lediglich in einigen der N skalaren Ersatzprobleme gute Leistung bringen, d.h. die Entstehung von **Nischen**, zu verhindern, wird

das **Sharing** Prinzip eingeführt: Der Fitness Wert der einzelnen Individuen (negative Summe der gewichteten Zielwerte) wird proportional zu der Anzahl von Individuen in ihrer näheren Umgebung (ähnlicher Individuen) herabgesetzt. Dabei verwendet man sogenannte Sharing Funktionen. d_{ij} bezeichne z.B. den euklidischen Abstand zweier Individuen i und j im Raum \mathbb{R}^{P+1} . Kontrolliert vom Sharing Parameter $\sigma_{sh} > 0$, der gewöhnlich zwischen 0.01 und 0.1 gewählt wird, hat die Sharing Funktion die Form

$$\Phi(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{sh}}\right)^\alpha, & d_{ij} < \sigma_{sh}, (\alpha \geq 1), \\ 0, & \text{sonst.} \end{cases} \quad (2.66)$$

Der verminderte Fitness Wert fv_i^{sh} des Individuums i entsteht aus der Vorschrift

$$fv_i^{sh} = \frac{fv_i}{\sum_{j=1}^P \Phi(d_{ij})}, \quad (2.67)$$

wobei P die Anzahl der Individuen in der Population der aktuellen Generation und fv_i der Fitness Wert des Individuums i ist. Da beim Wichtungsansatz die Ähnlichkeit zweier Individuen i und j vor allem von der Ähnlichkeit der Gewichtsvektoren abhängt, setzt man hier $d_{ij} := \sqrt{(s_w^i - s_w^j)^2}$.

Um die Konvergenz des Verfahrens zu beschleunigen, ist es möglich, die Paarung von Individuen auf Paare mit hohem Fitness Wert und genügend kleinem d_{ij} zu beschränken.

Wir fassen zusammen: In einem einzigen Suchlauf möchte der Wichtungsansatz eine Approximation für die gesammelten Lösungen aus verschiedenen Suchläufen der Methode der Summierung der gewichteten Zielfunktionen erarbeiten. Dabei besteht die Schwierigkeit darin, eine gute Menge zu berücksichtigender Gewichtsvektoren festzulegen. Leider bleibt wie bei der Wichtungsmethode die Tendenz erhalten, daß Alternativen mit Zielwertvektoren in nichtkonvexen Teilen der Effizienzmenge in $F(R)$ von der Suche vernachlässigt werden, obwohl der Zufallscharakter der Suche diesen Alternativen durchaus eine Entdeckungs- und Überlebenschance läßt.

2.3.3.4 Pareto-basierte Fitnessbewertung

Daß man für die Selektion von Individuen eine Rangeinteilung nach Nichtdominiertheit verwendet, um die Populationen von Generation zu Generation auf die Menge aller Paretooptima hinzulenken, geht auf eine Idee von Goldberg zurück ([23]).

In jeder Generation bekommen die Alternativen, die von keiner Alternative der gegenwärtigen Population dominiert werden, einen bestimmten Rang zugewiesen. Die Menge der verbleibenden Alternativen wird wieder nach in ihr nichtdominierten Alternativen durchsucht. Sie erhalten den nächsthöheren Rang. Die nichtdominierten Alternativen innerhalb des danach noch nicht untersuchten Populationsteiles bekommen einen Rang, der höher ist als die beiden bisher vergebenen Ränge. Dies geht so weiter, bis alle Individuen der aktuellen Generation einen Rang haben.

Je niedriger der Rang eines Individuums ist, desto wahrscheinlicher kann es zu den Eltern der neuen Generation gehören. Somit bekommen die relativ effizienten (relativ in Hinblick auf die aktuelle Population) Alternativen die besten Chancen, ihre Qualitäten auf die nachfolgende Generation zu übertragen.

Um der vorzeitigen Konzentration der Populationen in einzelnen Gebieten der Effizienzmenge in R vorzubeugen, kann ein Sharing Mechanismus eingebaut werden. Die Fitness Werte (hier das Inverse der Ränge) der einzelnen Individuen werden proportional zu der Anzahl ähnlicher Individuen in der gegenwärtigen Population verringert.

Pareto-basierte Fitnessbewertung versucht sicherlich, eine Approximation der Effizienzmenge in R zu erarbeiten, ohne Präferenzen miteinzubeziehen. Im Gegensatz zu **VEGA** umgeht das Verfahren die Gefahr, daß Paretooptima ohne besonders gute Leistung in mindestens einem Ziel benachteiligt werden.

2.3.3.5 Genetischer Mehrzielalgorithmus

Der Genetische Mehrzielalgorithmus folgt einem ähnlichen Grundgedanken wie der eben beschriebene Rangordnungsmechanismus. Fonseca und Fleming ([21]) haben ein Verfahren erfunden, in dem der Rang einer bestimmten Alternative der Anzahl der Alternativen entspricht, von denen sie dominiert wird.

Eine Alternative i der Generation $t \in \mathbb{N}_0$ werde von $p_i^t \in \mathbb{N}_0$ Alternativen der gegenwärtigen Population dominiert; dann bekommt sie den Rang $1 + p_i^t$. Ist innerhalb einer Generation eine Alternative relativ paretooptimal, so erhält sie Rang 1. Nachdem alle Individuen einer Population mit einem Rang versehen sind, wird die Population nach Rang geordnet. Jedes Mitglied einer bestimmten Rang-Subpopulation bekommt den gleichen Fitness Wert $\frac{|Population|}{Rang}$. Je niedriger sein Rang, desto wahrscheinlicher wird ein Individuum für eine Elternrolle für die nächste Generation ausersehen. Wieder bekommen die relativ paretooptimalen Populationsmitglieder die besten Chancen, ihre Qualitäten der nächsten Generation zu vererben.

Wie bereits in der vorhergehenden Methode praktiziert, baut man der Nischenbildung, also der unliebsamen vorzeitigen Konzentration der Populationen in einzelnen Gebieten der Effizienzmenge in R , durch eine Sharing Methode vor. Diese basiert diesmal auf Ähnlichkeiten zwischen den Zielwertvektoren von sich gegenseitig nicht dominierenden Alternativen. Damit soll das Überleben einer großen Vielfalt von Zielwertvektoren gewährleistet werden. Der Vielfaltserhaltung unter den Alternativen ist damit nicht geholfen - der Genetische Mehrzielalgorithmus kann sich als unfähig erweisen, wenn viele verschiedene Alternativen gleiche Zielwerte besitzen.

Ansonsten gelten die Bemerkungen zur Pareto-basierten Fitnessbewertung auch hier.

2.3.3.6 Genetischer Algorithmus mit Nichtdominanz-Sortierung

Eine weitere Mehrzieltechnik, die jede Population nach Nichtdominanz sortiert, stammt von Srinivas und Deb ([54]).

Zunächst werden wieder einmal die nichtdominierten Alternativen in der Population der jeweiligen Generation identifiziert und jeweils mit einem gleichgroßen Fitness Wert versehen. In der so entstandenen Subpopulation wird sodann Sharing durchgeführt, wobei dies auf den Ähnlichkeiten der Individuen (nicht der Zielwertvektoren) basiert. Die reduzierten Fitness Werte können innerhalb der Subpopulation im Gegensatz zu den ursprünglichen Fitness Werten differieren. Nun werden im bisher noch nicht mit einem Rang versehenen Teil der Population wieder die nichtdominierten Alternativen herausgesucht und mit einem Fitness Wert versehen, der niedriger ist als der niedrigste Fitness Wert der ersten Subpopulation nach dem Sharing. Auch in

dieser zweiten Subpopulation arbeitet der Sharing Mechanismus und es entstehen neue, verringerte Fitness Werte. Analog geht dies weiter, bis alle Individuen einer Generation per Rangzuweisung zu einer Subpopulation gehören und einen Fitness Wert (nach Sharing) haben.

Alsdann erfolgt die Reproduktion entsprechend der Fitness Werte zur Erzeugung einer neuen Generation. Somit werden die relativ paretooptimalen Alternativen einer Generation wahrscheinlicher und häufiger kopiert als alle anderen Individuen dieser Generation.

Die Absicht dieses Mehrzielverfahrens ist möglichst schnelle Konvergenz gegen effiziente Alternativen bei guter Verteilung der erzeugten/gefundenen Ergebnisalternativen über die Menge aller Paretooptima.

2.3.3.7 Nixed Pareto GA

Das Selektionsschema in dieser Methode von Horn und Nafpliotis ([31]) unterscheidet sich etwas von den Selektionsschemata der letzten vorgestellten Verfahren.

Zu Beginn jedes Selektionsprozesses wird eine sogenannte "Vergleichsmenge" aus der Population der aktuellen Generation per Zufallsexperiment ausgezeichnet. Sie besteht aus einer (im Vorhinein) zu bestimmenden Anzahl $t_{dom} > 0$ von Alternativen. Ein weiteres Zufallsexperiment bestimmt zwei Alternativen, die beide den Elementen der Vergleichsmenge gegenübergestellt werden. Wird eine der beiden von keiner Alternative der Vergleichsmenge dominiert, die andere jedoch schon, so wird erstere für die Reproduktion erwählt und die andere aussortiert. Diese Art der Selektion soll den Algorithmus auf die effizienten Alternativen hinführen. Werden entweder beide Alternativen von keiner Alternative der Vergleichsmenge dominiert oder beide von einer dominiert, so entscheidet eine Art Sharing Mechanismus: Für beide Alternativen werden die Individuen der aktuellen Population gezählt, die innerhalb eines bestimmten Abstandes σ_{sh} zu ihnen liegen (die verwendete Metrik ist vom Anwender festzulegen); dies nennt man "Nischenzählung". Diejenige der beiden Alternativen mit der niedrigeren Nischenzahl wird nun für die Reproduktion auserkoren. Auf diese Weise soll die Individuenvielfalt über die Generationen hinweg erhalten bleiben. Die eben beschriebene Selektionsprozedur nennt man in englischer Sprache **pareto domination tournaments**, was man im Deutschen in etwa mit **Dominanzturnier** übersetzen kann.

Wie man sich leicht vorstellen kann, hängt der Erfolg dieses Mehrzielverfahrens stark von der Wahl von t_{dom} ab: Ein zu kleines t_{dom} könnte zum Auffinden zu weniger effizienter Alternativen, ein zu großes zu vorzeitiger Konvergenz führen.

Wieder versucht das vorgestellte Verfahren, so viele effiziente Alternativen wie möglich und dabei eine möglichst gute Repräsentation der Menge aller Paretooptima zu erreichen.

2.4 Zusammenfassung

Wie bereits zu Beginn dieses Kapitels erwähnt, ist ein wichtiges Kriterium für die Klassifizierung von Vektoroptimierungsverfahren in der Literatur die Rolle des Entscheidungsträgers vor, während und nach der Anwendung der Verfahren. Daher gliedert sich dieses Kapitel nach den drei Kategorien

- Methoden mit a priori Präferenzäußerung,
- Methoden mit fortschreitender Präferenzäußerung,

- Generierungstechniken / Methoden mit a posteriori Präferenzäußerung.

Für Mehrzielverfahren mit a priori Präferenzäußerung muß der Entscheidungsträger z.B. in einigen Zielen Oberschranken für Zielwerte oder Trade-Off Raten zwischen einigen Zielen oder eine Rangordnung nach Priorität unter allen Zielen festlegen, bevor die Prozeduren starten. Geleitet durch die Präferenzinformation suchen die Methoden eine bestimmte effiziente Alternative, die unter allen Paretooptima vom Entscheidungsträger am meisten bevorzugt wird. Die Steuerung durch Präferenzinformation führt jedoch dazu, daß große Gebiete des Suchraumes von der Suche ausgeschlossen bleiben, was wiederum dazu führt, daß viele interessante Alternativen erst gar nicht bekannt werden. Wenn vom Entscheidungsträger die Präferenzaussagen verlangt werden, ist er eventuell gar nicht in der Lage, das Vektoroptimierungsproblem gut genug einzuschätzen, da er nicht weiß, was alles möglich ist, d.h. nicht alle Paretooptima kennt. Daraus ergibt sich die Gefahr, daß diese Klasse von Mehrzielverfahren häufig irreführt wird und der Entscheidungsträger mit dem Ergebnis nicht zufrieden ist.

Methoden mit fortschreitender Präferenzäußerung basieren auf dem Gedanken, daß die Komplexität des Entscheidungsproblems den Entscheidungsträger außerstande setzt, seine Präferenzen im Vorhinein zu formulieren oder sich ihrer überhaupt bewußt zu sein. Man erhofft sich, daß er im Suchverlauf mehr und mehr über das Problem lernt und daher wenigstens von Iteration zu Iteration lokale Präferenzaussagen machen kann. In jedem Frage-Antwort Austausch jeder Iteration bekommt der Entscheidungsträger einen kleinen Ausschnitt aus der Menge aller Paretooptima zu Gesicht und versucht aufgrund dieser neuen Information über das Mehrzielproblem Hinweise zu geben, in welche Richtung die Suche weiterzuführen ist. Wieder ist der Entscheidungsträger in hohem Maße für das Gelingen der Suchprozesse verantwortlich und kann die Verfahren deshalb auch leicht zum Scheitern bringen:

- Seine Äußerungen in verschiedenen Iterationen können sich z.B. widersprechen und das Suchverfahren verwirren.
- Obwohl er während des Lösungsvorganges dazulernt, könnte die Problemeinschätzung des Entscheidungsträgers immer noch so mangelhaft sein, daß er durch ungeschickte Präferenzaussagen das Suchverfahren in ihm weniger genehme Regionen der Effizienzmenge lenkt und somit von interessanten Alternativen nie erfährt.

Generierungstechniken konzentrieren sich darauf, so viel Information wie möglich über das Vektoroptimierungsproblem bereitzustellen. Sie streben an, alle Paretooptima oder eine (gute Approximation für eine) repräsentative Teilmenge der Effizienzmenge in R zu erarbeiten. Diese Ergebnismenge vermittelt dann einen Eindruck, was in den Zielen zu erreichen möglich ist und was für welchen Gewinn zu opfern ist. Der Entscheidungsträger muß seine Präferenzen nicht formulieren, bevor er nicht den bestmöglichen Überblick über das Entscheidungsproblem hat. Natürlich kann man befürchten, daß die Ergebnisvielfalt den Entscheidungsträger überwältigt. Ist dies der Fall, so kann man auf die große Ergebnismenge immer noch ein Verfahren mit a priori oder fortschreitender Präferenzäußerung anwenden, damit der Entscheidungsträger zu seiner Lieblingslösung kommt. Präferenzen sind dann leichter zu äußern, da der Entscheidungsträger jetzt auf dem bestmöglichen Informationsstand ist.

Die eben wie auch schon früher vorgebrachten Argumente sollen dem Leser klar machen, warum diese Arbeit sich dem Gedankengang der Generierungstechniken anschließt.

Nach jeder kurzen Verfahrensbeschreibung wurden Vor- und Nachteile der jeweiligen Mehrzielmethode mit Blick auf ihre Brauchbarkeit für diskret-kontinuierliche Vektoroptimierungsprobleme, wie sie in Kapitel 3 formuliert werden, zusammengestellt. Erinnert sich der Leser der angeführten Argumente gegen die vorgestellten Generierungstechniken, so mag er verstehen, warum die Entwicklung des in Kapitel 3 erklärten Verfahrens notwendig war. Um die Nachteile, die den meisten bisher bekannten Generierungstechniken gemeinsam sind, kurz zusammenzufassen, ist zu erwähnen, daß

- einige nur auf Probleme mit stetigen Zielfunktionen und Nebenbedingungen und verfügbarer Gradienteninformation zugeschnitten sind,
- einige keine Paretooptima mit Zielwertvektoren in nichtkonvexen Teilen von $F(R) + (\mathbb{R}_+^n \cup \{0\})$ finden können,
- bei einigen die Variation von Steuerparametern, die letztlich für die Repräsentativität der Ergebnismenge verantwortlich sind, zu schwierig ist,
- Ansätze, die auf Genetischen Algorithmen basieren, im allgemeinen keine Konvergenzaussagen erlauben, d.h. Aussagen darüber, mit welcher Wahrscheinlichkeit die Menge der Paretooptima nach genügend vielen Rechenschritten erreicht werden kann.

Kapitel 3

Diskret-kontinuierliche Vektoroptimierung

Bei der Lektüre des Kapitels 2 hat der Leser erfahren, daß auf dem Gebiet der diskret-kontinuierlichen Vektoroptimierung eigentlich noch kein ausreichendes Angebot an Mehrzielverfahren existiert.

Die meisten Vektoroptimierungsverfahren sind sowieso nur für Probleme mit stetigen, eigentlich mindestens stetig differenzierbaren Zielfunktionen und Nebenbedingungsfunktionen konzipiert.

Unter den Mehrzielmethoden, die auch auf diskret-kontinuierliche Vektoroptimierungsprobleme anwendbar sind, gibt es wiederum viele, die interaktiv auf Präferenzaussagen des Entscheidungsträgers reagieren und durch diese Präferenzaussagen zu einer einzigen effizienten Alternative, nämlich der zufriedenstellendsten Lösung für den Entscheidungsträger, navigiert werden sollen. Solche Verfahren gehen leicht in die Irre, sobald die Präferenzaussagen während ihrer Laufzeit nicht völlig konsistent sind.

Somit verbleiben im Lösungsangebot für diskret-kontinuierliche Vektoroptimierungsprobleme nur noch wenige Verfahren, nämlich die Generierungstechniken. Deren Absicht ist es, dem Entscheidungsträger möglichst alle Paretooptima zu liefern, numerisch entweder als Erarbeitung einer repräsentativen endlichen Teilmenge oder einer guten Approximation einer endlichen repräsentativen Teilmenge der Menge aller Paretooptima umsetzbar. Dadurch bekommt der Entscheidungsträger das richtige Problemverständnis und wird in seiner Präferenzbildung unterstützt.

Alle deterministischen Verfahren unter den Generierungstechniken weisen entweder das Manko auf, daß sie Paretooptima, deren Zielwertvektoren in nichtkonvexen Bereichen von $F(R)$ liegen, nicht finden können oder daß für wichtige Steuerparameter des Suchverhaltens die Variation eine schwierige und heikle Angelegenheit ist. Um es auf den Punkt zu bringen: Es ist sehr fraglich, ob eine der in der Literatur bekannten deterministischen Generierungstechniken in der Lage ist, eine repräsentative Teilmenge der Effizienzmenge in R zu erzeugen, ohne daß stark einschränkende Bedingungen an die Form der Menge $F(R)$ oder die Anzahl der bei vernünftigem Rechenaufwand verkraftbaren Ziele eines Vektoroptimierungsproblems zu stellen sind.

An stochastischen Generierungstechniken gibt es im Angebot für diskret-kontinuierliche Mehrzieloptimierungsprobleme lediglich Ansätze, die auf Genetischen Algorithmen basieren. Natürlich wünscht man sich gerade bei stochastischen Verfahren, die ja ihre Absichten nie

sicher, sondern lediglich mit bestimmten Wahrscheinlichkeiten realisieren können, etwas konkretere Aussagen über diese Wahrscheinlichkeiten. Solche sogenannten Konvergenzaussagen erlauben jedoch Genetische Algorithmen im allgemeinen nicht.

Die Entwicklung des in diesem Kapitel vorgestellten stochastischen Verfahrens zur Lösung diskret-kontinuierlicher Vektoroptimierungsprobleme stößt in eine große Lücke auf dem "Optimierungsmarkt", die insofern eine schmerzliche ist, weil viele Entscheidungsprobleme der (industriellen) Praxis in mathematischer Formulierung diskret-kontinuierliche Mehrzielprobleme ergeben.

3.1 Das diskret-kontinuierliche Vektoroptimierungsproblem

In dieser Arbeit betrachten wir Vektoroptimierungsprobleme mit folgender spezieller Alternativenmenge R :

$$\begin{aligned} R &:= C \times D, \\ C &\subseteq \mathbb{R}^{m_1}, C := [a, b], a, b \in \mathbb{R}^{m_1}, a_i < b_i \quad \forall \quad i = 1, \dots, m_1, \\ D &\subseteq \mathbb{R}^{m_2}, |D| < \infty, \\ m &= m_1 + m_2, m, m_1, m_2 \in \mathbb{N}. \end{aligned} \tag{3.1}$$

Die ersten m_1 Komponenten eines $x \in R$ werden in x_C und die letzten m_2 Komponenten in x_D zusammengefaßt. x_C ist Element einer kompakten Teilmenge C von \mathbb{R}^{m_1} und x_D Element einer endlichen Teilmenge D von \mathbb{R}^{m_2} . Die Menge R ist **diskret-kontinuierlich**.

Das in dieser Arbeit entwickelte Mehrzielverfahren könnte auch diskret-kontinuierliche Vektoroptimierungsprobleme mit allgemeiner gehaltenen Nebenbedingungen als Box-Constraints lösen. Wichtig ist lediglich, daß zu jeder Festlegung $\bar{x}_D \in D$ die zulässigen kontinuierlichen Variationsmöglichkeiten aus einem Kompaktum stammen. Im Rahmen der Aufgabenstellung ist es jedoch ausreichend, das Problem möglichst unkompliziert zu formulieren.

Das auf diesem speziellen R definierte Mehrzielproblem kann folgendermaßen formuliert werden:

$$" \min_{(x_C, x_D) \in C \times D} F(x_C, x_D), F : C \times D \mapsto \mathbb{R}^n, (x_C, x_D) \mapsto \begin{pmatrix} f_1(x_C, x_D) \\ f_2(x_C, x_D) \\ \dots \\ f_{n-1}(x_C, x_D) \\ f_n(x_C, x_D) \end{pmatrix}, \tag{3.2}$$

$$f_i : C \times D \mapsto \mathbb{R}, (x_C, x_D) \mapsto f_i(x_C, x_D), i = 1, \dots, n,$$

$$f_i^{x_D} : C \mapsto \mathbb{R}, x_C \mapsto f_i(x_C, x_D) \text{ zweimal stetig differenzierbar}$$

$$\forall \quad i = 1, \dots, n \quad \text{und} \quad x_D \in D.$$

Da C kompakt ist und für jedes $x_D \in D$ die Zielfunktionen $f_i^{x_D} : C \mapsto \mathbb{R}, x_C \mapsto f_i^{x_D}(x_C), i = 1, \dots, n$, stetig sind, ist $F(R) = \cup_{x_D \in D} F^{x_D}(C)$ mit $F^{x_D} : C \mapsto \mathbb{R}^n, x_C \mapsto (f_1^{x_D}(x_C), \dots, f_n^{x_D}(x_C))^T$, als Vereinigung endlich vieler kompakter Mengen kompakt. Nach der Aussage von Satz 1.4.12 gilt $E(F(R), \leq) \neq \emptyset$, d.h. für das gegebene diskret-kontinuierliche Mehrzieloptimierungsproblem existieren effiziente Alternativen.

3.2 Suche entlang von Pfaden dominierender Alternativen

3.2.1 Präferenzfreie Bewertung von Alternativen

In Abschnitt 1.4 wurden die Schwierigkeiten, zwei Alternativen aus R sinnvoll miteinander zu vergleichen, dargelegt. Wie in der skalaren Optimierung liegen Vergleichen auch in der Mehrzieloptimierung die Zielfunktionswerte zugrunde. Vergleichbarkeit und Optimalität ergeben sich aus der topologischen Ordnung des Kriterienraumes $F(R)$. Die Tatsache, daß $F(R)$ nicht vollständig geordnet ist, macht es allerdings unmöglich, zwei beliebige Alternativen ohne weitere Definitionen miteinander zu vergleichen. Betrachtet man eine bestimmte Alternative $x \in R$, so ermöglicht es uns der Begriff **”Dominanz”**, R in drei verschiedene Alternativenmengen hinsichtlich x aufzuteilen:

- Alle Alternativen y aus R , die in jedem Ziel mindestens so gute Zielwerte aufweisen wie x und in mindestens einem Ziel sogar einen besseren Wert haben, erachtet man für besser als x . Sie bilden die Menge $\{y \in R | x \succ y\}$.
- Alle y in R , die in keinem Ziel bessere Ausprägung haben als x und sogar in mindestens einem Ziel schlechtere Werte besitzen, betrachtet man folglich als schlechter als x . Sie ergeben die Menge $\{y \in R | x \prec y\}$.
- Alle Alternativen y aus R , die nicht einer dieser beiden Mengen angehören, gelten als genauso gut wie x . Sie liegen in der Menge $\{y \in R | x \asymp y\}$.

Wie bereits im Anschluß an Bemerkung 1.4.4 erwähnt, haben wir jetzt immer noch keine vollständige Ordnung in $F(R)$ gefunden, nicht einmal eine Halbordnung, aber wir sind in der Lage, die Aussagen ”besser”, ”schlechter” oder ”genauso gut wie” innerhalb jedes Paares $(x, y) \in R \times R$ zu machen. Weder berücksichtigen diese Aussagen Prioritäten des Entscheidungsträgers unter den Zielen - jedes Ziel wird gleich behandelt - noch drücken sie verschiedene Grade von ”besser” und ”schlechter” aus. Es handelt sich also um einen präferenzfreien, neutralen Vergleich. Wie in Kapitel 1 bereits dargelegt, führt die Dominanz in der Mehrzieloptimierung zu einer speziellen Definition von Optimalität, der **Effizienz**.

Natürlich kann es sein, daß der Entscheidungsträger viel Wert auf gute Leistung der Alternativen in bestimmten Zielen legt und dafür in anderen Zielen mit einem weniger als optimalen Leistungsstand zufrieden ist, ihm einige Ziele sogar fast gleichgültig sind. Nichtsdestotrotz ist es in den meisten Entscheidungsproblemen sinnvoll, Alternativen, für die es keine Alternative gibt, die in allen Zielen mindestens genauso gute Werte und in mindestens einem Ziel einen besseren Wert aufweist, als (pareto)optimal zu betrachten.

Die höherdimensionalen Zielwertvektoren der Alternativen sind nicht vollständig geordnet. Wie kann man dann die Alternativen so bewerten, daß ein Optimierungsalgorithmus zur Menge

aller Optima, hier zur Effizienzmenge in R führt?

Falls wir versuchen, R künstlich vollständig zu ordnen, indem wir an jede Alternative irgendeine reelle Bewertungszahl vergeben, muß diese Bewertungszahl die Dominanzrelationen in jedem Paar von Alternativen in R widerspiegeln. Die Bewertungszahlen dürfen sich auch während eines Suchprozesses nach Paretooptima (mit zunehmender Information) nicht ändern. Der einzige Weg, alle Alternativen auf diese Weise zu bewerten, liefe über vollständige Enumeration. Diese ist bei großem R mit zu starkem Rechenaufwand verbunden, bei $|R| = \infty$ gar unmöglich und würde ohnehin jeden Mehrzieloptimierungsalgorithmus überflüssig machen.

Da also absolute punktweise Bewertung nicht bei sinnvollem Rechenaufwand realisierbar ist, bleibt eine weitere Möglichkeit übrig: Die Bewertung von Alternativenpaaren. Zwei Alternativen $x \neq y, x, y \in R$, können in drei verschiedenen Relationen zueinander stehen: entweder $x \succ y$ oder $x \prec y$ oder $x \asymp y$. Daher kann man dem Paar (x, y) eine aus drei möglichen Bewertungszahlen zuordnen, nämlich entweder min oder mid oder max , $min < mid < max, min, mid, max \in \mathbb{R}$. Wann auch immer während einer Suchprozedur ein bestimmtes Alternativenpaar (x, y) angelaufen wird - die Relation zwischen x und y ändert sich nicht und somit bleibt auch die Bewertungszahl unverändert erhalten. Die Bewertungszahl ist nicht vom Verhältnis von x und y zu anderen Alternativen abhängig, so daß sie bei weiterer Erforschung des Suchraumes nicht angepaßt werden muß. Ein skalarer Optimierungsalgorithmus, der noch zu erarbeiten ist, kann somit auf Alternativenpaaren und deren Bewertungszahlen arbeiten. Die Bewertungszahlen sind Funktionswerte einer Funktion, die sich definiert durch

$$Z : R \times R \mapsto \{min, mid, max\} \subset \mathbb{R}, (x, y) \mapsto Z[(x, y)] := \begin{cases} min, & \text{falls } y \prec x, \\ mid, & \text{falls } y \asymp x, \\ max, & \text{falls } y \succ x. \end{cases} \quad (3.3)$$

3.2.2 Suchstrategie "Sequenzielle Suche" (SS)

Wie kann eine Suchprozedur auf Alternativenpaaren $(x, y) \in R \times R, x \neq y$, zu einer nichtdominierten Alternative führen? Sie sollte irgendwie eine Folge von Alternativen $x_i \in R, i = 0, 1, \dots, k, k \in \mathbb{N}_0$, hervorbringen mit $x_0 \succ x_1 \succ \dots \succ x_k$, die in einem Paretooptimum x_k endet. Erzeugt sie mehrere solcher Folgen hintereinander, sind mehrere effiziente Alternativen gefunden.

Wir gehen von einem beliebigen Startpunkt $x_0 \in R$ aus. Ein skalarer Optimierungsalgorithmus kann auf Alternativenpaaren $(x_0, y) \in \{x_0\} \times (R \setminus \{x_0\})$ mit jeweils gleichem "ersten Partner" oder **Referenzpunkt** x_0 arbeiten und dabei die Zielfunktion $Z : R \times R \mapsto \{min, mid, max\}, (x, y) \mapsto Z[(x, y)]$ laut (3.3) minimieren, d.h. nach Paaren $(x_0, y_0) \in \{x_0\} \times (R \setminus \{x_0\})$ mit minimalem Zielfunktionswert $Z[(x_0, y_0)]$ suchen. Ein (globaler) Minimierer von Z auf dem Suchraum $\{x_0\} \times (R \setminus \{x_0\})$ liefert zwei Arten von Information:

1. Ist $Z[(x_0, y_0)] = min$, dann gibt es offensichtlich eine bessere Alternative y_0 als x_0 , denn $y_0 \prec x_0$, und x_0 ist sicher nicht effizient.
2. Falls $Z[(x_0, y_0)] > min$ ist, wird x_0 von keiner Alternative aus R dominiert, x_0 ist also ein Paretooptimum. In der numerischen Praxis ist meist nicht sicher, ob der Optimierungsalgorithmus wirklich einen globalen Minimierer des zu bearbeitenden Problems gefunden

hat, womit sich die Aussage reduziert auf: Der skalare Optimierungsalgorithmus konnte keine dominierende Alternative für x_0 finden; x_0 scheint daher paretooptimal zu sein.

Bei $Z[(x_0, y_0)] = \min$ bildet $x_1 = y_0$ dann den Referenzpunkt für eine weitere skalare Suchprozedur: Ein skalarer Optimierungsalgorithmus betrachtet Punktepaare $(x_1, y) \in \{x_1\} \times (R \setminus \{x_1\})$ und minimiert auf diesem Suchraum erneut die Zielfunktion Z gemäß (3.3). Wieder kann ein global minimales Ergebnis (x_1, y_1) zweierlei ausdrücken:

1. Im Falle $Z[(x_1, y_1)] = \min$ konnte eine weitere Verbesserung $y_1 \prec x_1 \prec x_0$ erreicht werden. Darüber hinaus erfahren wir, daß x_1 nicht paretooptimal ist.
2. Mit $Z[(x_1, y_1)] > \min$ darf x_1 als effizient angesehen werden.

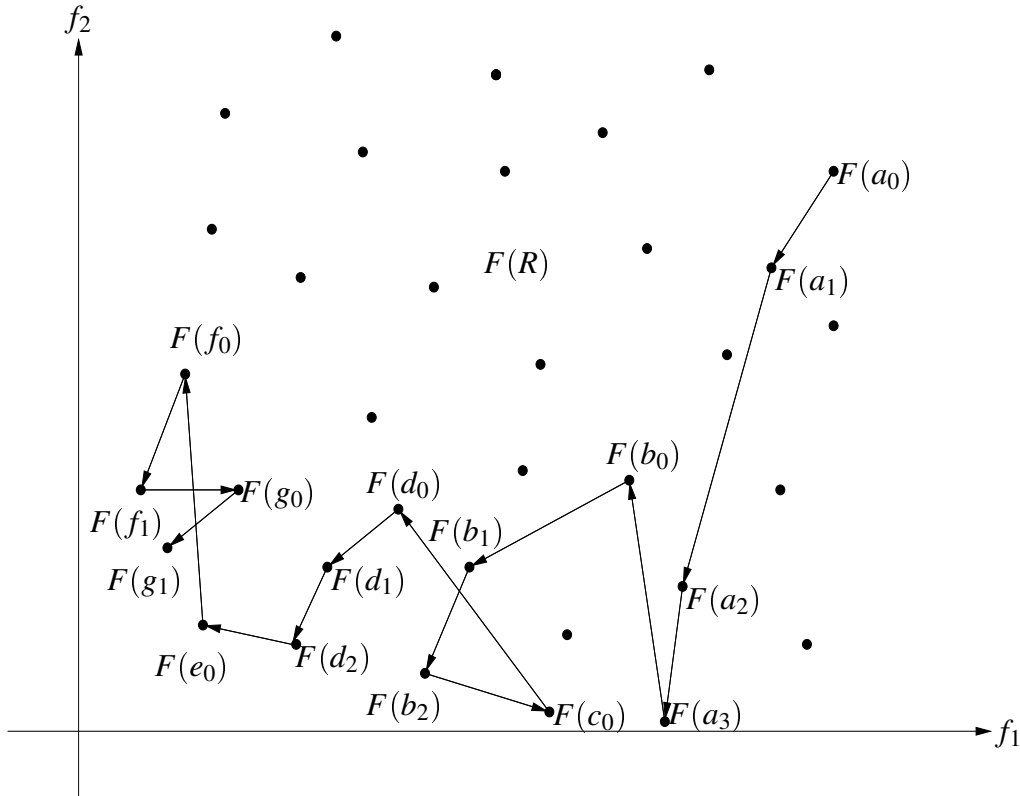
Durch $Z[(x_1, y_1)] = \min$ sind wir zu einem neuen Referenzpunkt $x_2 = y_1$ für eine weitere Suche nach Minima von Z , diesmal auf Paaren $(x_2, y) \in \{x_2\} \times (R \setminus \{x_2\})$, gekommen usw.

Durch weiteres Hintereinanderausführen von skalaren Suchprozeduren auf Alternativenpaaren erarbeiten wir uns die gewünschte Folge $x_0 \succ x_1 \succ \dots \succ x_k$ von Alternativen aus R und entdecken dabei ein Paretooptimum $x_k, k \in \mathbb{N}_0$ (oder genauer eine Alternative, für die der arbeitende skalare Optimierungsalgorithmus keine dominierende Alternative finden konnte).

Da wir ja an viele Paretooptima heranzukommen wünschen, wollen wir viele solcher Folgen dominierender Alternativen mit Ende in einer effizienten erzeugen. Es gibt nun zwei Möglichkeiten fortzufahren:

1. Entweder wählen wir uns eine beliebige neue Startalternative bzw. erste Referenzalternative $\hat{x}_0 \in R$ ungleich der vorhergehenden Startalternative x_0 und erarbeiten, wie bereits beschrieben, eine Folge $\hat{x}_0 \succ \hat{x}_1 \succ \dots \succ \hat{x}_p, p \in \mathbb{N}_0$, von dominierenden Alternativen aus R , die in einem Paretooptimum \hat{x}_p gipfelt,
2. oder verwenden aus dem Ergebnis (x_k, y_k) der skalaren Optimierungsprozedur, die x_k als effizient einstufte, den "zweiten Partner" y_k als neue Startalternative. Ist (x_k, y_k) wirklich ein globaler Minimierer von Z auf $\{x_k\} \times (R \setminus \{x_k\})$, ergeben sich zwei verschiedene Szenarien:
 - Falls $Z[(x_k, y_k)] = \min$ ist, also $y_k \prec x_k$, gehört $\hat{x}_0 = y_k$ bereits zu einer Obermenge $Ind_{x_k} := \{x \in R : x \prec x_k\} \subset R$ der Effizienzmenge in R ; vielleicht ist bereits eine weitere effiziente Alternative gefunden. Jedenfalls bleiben wir mit unserer Wahl von \hat{x}_0 so nahe wie möglich bei der Effizienzmenge in R und starten nicht erst bei einem von x_k dominierten Punkt, der sicher nicht paretooptimal ist.
 - Ist $Z[(x_k, y_k)] = \max$, also $y_k \succ x_k$, so gibt es kein $x \in R$ mit $x \prec x_k$, d.h. die Obermenge $Ind_{x_k} \subset R$ der Effizienzmenge in R ist leer; außer x_k existiert kein weiteres Paretooptimum und jeder weitere Suchaufwand in R ist überflüssig.

Abbildung 3.1: Suchstrategie SS erzeugt in diesem Beispiel die Folge $a_0 \succ a_1 \succ a_2 \succ a_3 \prec b_0 \succ b_1 \succ b_2 \prec c_0 \prec d_0 \succ d_1 \succ d_2 \prec e_0 \prec f_0 \succ f_1 \prec g_0 \succ g_1$ in R . Dabei gelten: $Z[(a_0, a_1)] = \min, Z[(a_1, a_2)] = \min, Z[(a_2, a_3)] = \min, Z[(a_3, b_0)] = \text{mid}, Z[(b_0, b_1)] = \min, Z[(b_1, b_2)] = \min, Z[(b_2, c_0)] = \text{mid}, Z[(c_0, d_0)] = \text{mid}, Z[(d_0, d_1)] = \min, Z[(d_1, d_2)] = \min, Z[(d_2, e_0)] = \text{mid}, Z[(e_0, f_0)] = \text{mid}, Z[(f_0, f_1)] = \min, Z[(f_1, g_0)] = \text{max}, Z[(g_0, g_1)] = \min$. Hintereinander werden Folgen dominierender Punkte erarbeitet, nämlich a_0, a_1, a_2, a_3 ; b_0, b_1, b_2 ; c_0 ; d_0, d_1, d_2 ; e_0 ; f_0, f_1 ; g_0, g_1 . Dabei sammelt SS die Paretooptima $a_3, b_2, c_0, d_2, e_0, f_1, g_1$ auf.



3.2.3 Zusammenspiel von diskreten und kontinuierlichen Suchschritten

Im Suchraum R des Problems (3.2) kann man nicht mit Ableitungsinformation arbeiten. Da $|R|$ als sehr groß oder sogar als unendlich angenommen wird, liegt es nahe, bei der Suche nach Paretooptima ein diskretes stochastisches Verfahren anzuwenden. Dennoch steckt in (3.2) wertvolle Gradienteninformation, da für alle $x_D \in D$ sämtliche Funktionen $f_i^{x_D}, i = 1, \dots, n$, zweimal stetig differenzierbar sind. Wie kann man sich stochastisch in R bewegen, ohne diese Gradienteninformation ungenutzt zu lassen?

Der Einfachheit halber betrachten wir zunächst den eindimensionalen Fall $n = 1$, d.h. das zu lösende Problem lautet

$$\min_{(x_C, x_D) \in C \times D} f_1(x_C, x_D). \quad (3.4)$$

Die stochastische Generierung eines Punktes \bar{x} in R ist folgendermaßen möglich:

1. Zunächst kann man stochastisch eine zulässige diskrete Variation $\bar{x}_D \in D$ wählen.
2. Dann minimiert man die Zielfunktion $f_1^{\bar{x}_D}$ kontinuierlich über C :

$$\min_{x_C \in C} f_1^{\bar{x}_D}(x_C) = \min_{x_C \in C} f_1(x_C, \bar{x}_D). \quad (3.5)$$

Die Kompaktheit von C und Stetigkeit von $f_1^{\bar{x}_D}$ garantieren die Existenz einer nichtleeren Lösungsmenge $L^{\bar{x}_D} \subseteq C$ von Problem (3.16). Es ist nützlich, die Menge $E^{\bar{x}_D} := \{(x_C, \bar{x}_D) \in C \times D \mid x_C \in L^{\bar{x}_D}\}$ einzuführen.

3. Sodann wählt man stochastisch eine Alternative (\bar{x}_C, \bar{x}_D) aus $E^{\bar{x}_D}$

Die Schritte 1 und 3 bei der stochastischen Generierung von Alternativen in R müssen dazu führen, daß schließlich das Problem

$$\min_{x \in \cup_{x_D \in D} E^{x_D}} f_1(x) \quad (3.6)$$

gelöst wird. Die Lösungsmenge von (3.6) entspricht genau der von Problem (3.4). Bei der Lösung von (3.6) bewegt man sich nur auf den "besten" Punkten zu den jeweiligen festen diskreten Werten \bar{x}_D , nämlich den Mengen $E^{\bar{x}_D}$, $\bar{x}_D \in D$, zu denen man unter Einbeziehung der vorhandenen Ableitungsinformationen kommt. Es wurde hervorgehoben, daß die Probleme (3.5) für alle $x_D \in D$ Lösungen besitzen. Dies scheint aufgrund der Lösbarkeit von Problem (3.4) trivial zu sein. Eine Vergewisserung, ob die kontinuierlichen Subprobleme Lösungen besitzen, ist nicht überflüssig, auch wenn man weiß, daß das diskret-kontinuierliche Gesamtproblem lösbar ist. Um dies zu demonstrieren, denken wir im folgenden Beispiel an ein Problem mit nichtkompakter Menge C :

Beispiel 3.2.1. Gegeben sei ein spezielles diskret-kontinuierliches Optimierungsproblem

$$\begin{aligned} \min_{(x_C, x_D) \in C \times D} f_1(x_C, x_D), f_1(x_C, x_D) &:= x_D \exp(x_C), \\ C &:=]-\infty; 2], D := \{0, 1\}. \end{aligned} \quad (3.7)$$

Es gelten $f_1(x_C, x_D) \geq 0 \quad \forall (x_C, x_D) \in C \times D$ und $f_1(x_C, x_D) = 0$ für alle $(x_C, 0), x_C \in C$. Diese Punkte $(x_C, 0), x_C \in C$, sind also die globalen Minimierer des gegebenen Problems.

Bei der Konstruktion der beiden möglichen kontinuierlichen Subprobleme hat

- 1.

$$\min_{x_C \in]-\infty; 2]} f_1^0(x_C), f_1^0(x_C) := 0 \exp(x_C), \quad (3.8)$$

die gesamte Menge C als Lösungsmenge und

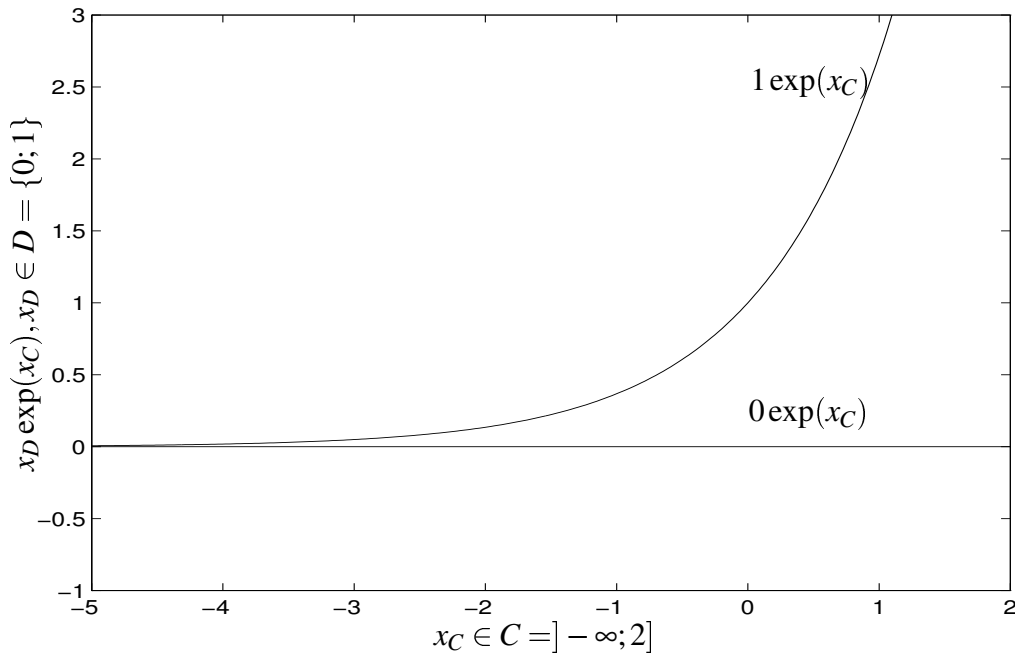
- 2.

$$\min_{x_C \in]-\infty; 2]} f_1^1(x_C), f_1^1(x_C) := 1 \exp(x_C), \quad (3.9)$$

keine Lösungen.

△

Abbildung 3.2: In Beispiel 3.2.1 hat die kontinuierliche Sub-Zielfunktion $f_1^1 = 1 \exp(x_C)$ kein Minimum, wohingegen jedes Element aus C ein globaler Minimierer von $f_1^0 = 0 \exp(x_C)$ ist.



Die bei der skalaren diskret-kontinuierlichen Optimierung erworbenen Erkenntnisse lassen sich auf das Mehrzielproblem (3.2) übertragen. Das kontinuierliche Subproblem kann für jedes $\bar{x}_D \in D$ so formuliert werden:

$$\begin{aligned}
 & \text{"min"}_{x_C \in C} F(x_C, \bar{x}_D) = \text{"min"}_{x_C \in C} F^{\bar{x}_D}(x_C), \\
 & F(x_C, \bar{x}_D) := \begin{pmatrix} f_1(x_C, \bar{x}_D) \\ f_2(x_C, \bar{x}_D) \\ \dots \\ f_{n-1}(x_C, \bar{x}_D) \\ f_n(x_C, \bar{x}_D) \end{pmatrix}, F^{\bar{x}_D}(x_C) := \begin{pmatrix} f_1^{\bar{x}_D}(x_C) \\ f_2^{\bar{x}_D}(x_C) \\ \dots \\ f_{n-1}^{\bar{x}_D}(x_C) \\ f_n^{\bar{x}_D}(x_C) \end{pmatrix}, \\
 & f_i : C \times D \mapsto \mathbb{R}, (x_C, x_D) \mapsto f_i(x_C, x_D), \\
 & f_i^{\bar{x}_D} : C \mapsto \mathbb{R}, x_C \mapsto f_i(x_C, \bar{x}_D), \text{ zweimal stetig differenzierbar} \\
 & \quad \forall \quad i = 1, \dots, n.
 \end{aligned} \tag{3.10}$$

Für jedes $\bar{x}_D \in D$ hat das zugehörige kontinuierliche Subproblem (3.10) nach Satz 1.4.12 eine nichtleere Effizienzmenge in R - das Problem ist lösbar.

Die Aussage dieses letzten Satzes darf man wiederum nicht bei allen lösbaren diskret-kontinuierlichen Mehrzieloptimierungsproblemen für selbstverständlich halten. Wie bereits im Einzelfall ist es auch hier möglich, ein diskret-kontinuierliches Problem mit Paretooptima zu konstruieren, bei dem ein kontinuierliches Subproblem keine paretooptimalen Lösungen hat, z.B. wenn C nicht kompakt ist:

Beispiel 3.2.2. Wir wollen ein diskret-kontinuierliches Problem der Form

$$\begin{aligned} \text{"min"}_{(x_C, x_D) \in C \times D} F(x_C, x_D), F(x_C, x_D) &:= \begin{pmatrix} x_D \exp(x_C) \\ -x_D x_C^2 \end{pmatrix}, \\ C &:=]-\infty, 2], D := \{0, 1\}, \end{aligned} \quad (3.11)$$

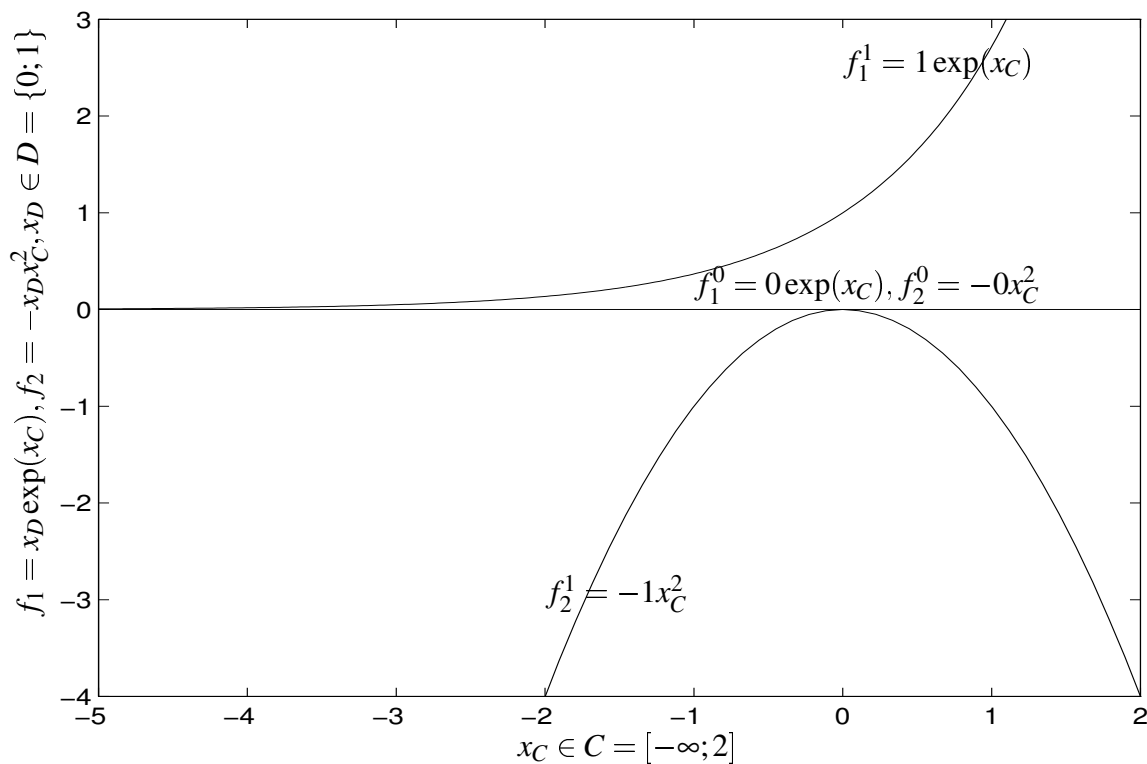
betrachten. Alle $(x_C, 0), x_C \in C$, sind paretooptimal. Für $(x_C, 1), x_C \in C$, erhalten wir

$$\begin{aligned} \lim_{x_C \rightarrow -\infty} f_1^1(x_C) = 0, \quad \lim_{x_C \rightarrow -\infty} f_2^1(x_C) = -\infty, \\ f_1^1(x_C) > 0, f_2^1(x_C) \leq 0 \quad \forall x_C \in C. \end{aligned} \quad (3.12)$$

Zu jedem $(\bar{x}_C, 1) \in C \times D$ gibt es ein $(\hat{x}_C, 1) \in C \times D$ mit $\hat{x}_C < \bar{x}_C$, das $(\hat{x}_C, 1) \prec (\bar{x}_C, 1)$ hervorbringt. Alle $(\hat{x}_C, 0), \hat{x}_C \in C$, erfüllen jedoch

$$\begin{aligned} f_1(\bar{x}_C, 1) > f_1(\hat{x}_C, 0), f_2(\bar{x}_C, 1) \leq f_2(\hat{x}_C, 0) \iff \text{entweder } (\bar{x}_C, 1) \succ (\hat{x}_C, 0) \\ \text{oder } (\bar{x}_C, 1) \prec (\hat{x}_C, 0) \quad \forall \bar{x}_C \in C. \end{aligned} \quad (3.13)$$

Abbildung 3.3: In Beispiel 3.2.2 kann man zu jeder Alternative $(\bar{x}_C, 1) \in R$ eine dominierende Alternative $(\hat{x}_C, 1) \in R$ mit $\hat{x}_C < \bar{x}_C$ finden. Jede Alternative $(x_C, 0) \in R$ jedoch ist effizient.



Es gibt zwei kontinuierliche Subprobleme:

1.

$$"min_{x_C \in C} F^0(x_C), F^0(x_C) := \begin{pmatrix} 0 \exp(x_C) \\ -0x_C^2 \end{pmatrix}, \quad (3.14)$$

dessen Effizienzmenge in C ganz C ist und

2.

$$"min_{x_C \in C} F^1(x_C), F^1(x_C) := \begin{pmatrix} 1 \exp(x_C) \\ -1x_C^2 \end{pmatrix}, \quad (3.15)$$

das keine paretooptimalen Lösungen besitzt, da zu jedem $\bar{x}_C \in C$ ein $\hat{x}_C \in C$ mit $\hat{x}_C < \bar{x}_C$ gefunden werden kann, so daß $\hat{x}_C \prec \bar{x}_C$.

△

Wir wollen nun allerdings zu Problem (3.2) zurückkehren, bei dem alle kontinuierlichen Subprobleme (3.10) lösbar sind:

Analog zum skalaren Fall ist die stochastische Generierung eines Punktes \bar{x} in R folgendermaßen vorstellbar:

1. Zunächst wählt man stochastisch eine zulässige diskrete Variation $\bar{x}_D \in D$.
2. Dann minimiert man die Zielfunktion $F^{\bar{x}_D}$ kontinuierlich über C :

$$\min_{x_C \in C} F^{\bar{x}_D}(x_C) = \min_{x_C \in C} F(x_C, \bar{x}_D). \quad (3.16)$$

Von der zugehörigen Lösungsmenge $L^{\bar{x}_D} \subseteq C$ leiten wir die Menge aller Paretooptima in $C \times \{\bar{x}_D\} \subseteq R$ hinsichtlich der Mehrzielfunktion F her:

$$E^{\bar{x}_D} = \{(x_C, \bar{x}_D) | x_C \in L^{\bar{x}_D}\} \subseteq R, \quad (3.17)$$

die Menge aller **kontinuierlich-effizienten Punkte** zu festem diskreten $\bar{x}_D \in D$.

3. Sodann wählt man stochastisch eine Alternative (\bar{x}_C, \bar{x}_D) aus $E^{\bar{x}_D}$.

Die Schritte 1 und 3 bei der stochastischen Generierung von Alternativen in R sollen schließlich zur Lösung des Problems

$$\min_{x \in \cup_{x_D \in D} E^{x_D}} F(x) \quad (3.18)$$

führen.

Unter der Annahme, daß E^{x_D} für jedes $x_D \in D$ beschaffbar ist, ergibt sich also als Suchraum R_D für ein stochastisches diskretes Mehrzielverfahren die Menge

$$R_D := \cup_{x_D \in D} E^{x_D} \subseteq R. \quad (3.19)$$

Die Lösung von

$$\begin{aligned} \text{"min"}_{(x_C, x_D) \in R_D} F(x_C, x_D), F(x_C, x_D) &:= \begin{pmatrix} f_1(x_C, x_D) \\ f_2(x_C, x_D) \\ \dots \\ f_{n-1}(x_C, x_D) \\ f_n(x_C, x_D) \end{pmatrix}, & (3.20) \\ f_i : C \times D &\mapsto \mathbb{R}, (x_C, x_D) \mapsto f_i(x_C, x_D) \quad \forall i = 1, \dots, n. \end{aligned}$$

ist identisch mit der Lösung von Problem (3.2):

- Die Menge aller Paretooptima von (3.2) ist sicher eine Teilmenge von R_D , weil jeder Punkt $(x_C, x_D) \notin R_D$ von mindestens einem Punkt mit denselben diskreten Komponenten $x_D \in D$ dominiert wird.
- Jeder Lösungspunkt $(\bar{x}_C, \bar{x}_D) \in R_D$ von (3.20) ist effizient. Sollte nämlich ein dominierender Punkt $(\hat{x}_C, \hat{x}_D) \in (C \times D)$ existieren, müßte dieser von einem Punkt $(\tilde{x}_C, \tilde{x}_D) \in R_D$ dominiert werden, der auch (\hat{x}_C, \hat{x}_D) dominiert. Dies widerspricht jedoch der Tatsache, daß (\bar{x}_C, \bar{x}_D) die Lösung des diskreten Problems (3.20) ist.

Da ein guter stochastischer Algorithmus zur Lösung kontinuierlicher Mehrzielprobleme, der uns eine sehr repräsentative Teilmenge bzw. Näherungsmenge der Effizienzmenge jedes durch ein zufällig gewähltes \bar{x}_D angesprochenen Problems (3.10) erzeugt, zur Verfügung steht (kurze Beschreibung siehe Abschnitt 2.3.2.3), verbleibt als Aufgabe dieser Arbeit, das Vektoroptimierungsproblem (3.20) zu lösen. Gegenüber R hat R_D den Vorteil, daß es eine Menge mit nützlicher Struktur ist, wie in Abschnitt 3.2.7 noch deutlich werden wird.

Natürlich ist die numerisch erzeugte Lösungsmenge $\bar{L}^{\bar{x}_D}$ des Problems (3.10) für ein \bar{x}_D nur eine endliche Teilmenge oder Näherungsmenge der tatsächlichen Menge $L^{\bar{x}_D}$ und somit das numerisch erzeugte $\bar{E}^{\bar{x}_D}$ ebenfalls nur eine (Approximation für eine) endliche Teilmenge der eigentlichen Menge $E^{\bar{x}_D}$. Folglich wird auch die Lösung von Problem (3.20) nur zu einer endlichen Teilmenge oder Näherungsmenge einer endlichen Teilmenge der Menge aller Paretooptima von Problem (3.2) führen, deren Güte letztlich auch von der Leistungsfähigkeit des benützten kontinuierlichen Mehrzieloptimierungsverfahrens abhängt.

Auch wenn einem numerischen Lösungsverfahren also nie das wahre R_D zur Verfügung steht, wird in den folgenden Ausführungen auch dann von R_D die Rede sein, wenn eigentlich nur numerische Approximationen bestimmter Teile von R_D zur Verfügung stehen. Analog wird der Einfachheit halber die Bezeichnung $E^{\bar{x}_D}$ auch für $\bar{E}^{\bar{x}_D}$ verwendet werden.

3.2.4 Bearbeitung der diskreten Suchschritte

Bemerkung 3.2.3. Bei **diskreten Optimierungsproblemen** wollen wir unter einem **lokalen Minimierer** einen Punkt des zulässigen Bereiches verstehen, dessen Zielfunktionswert weder das Maximum, noch das Minimum der Wertemenge darstellt. Analog gelte bei **diskreten Vektoroptimierungsproblemen** eine Alternative, die nicht paretooptimal ist, jedoch mindestens eine Alternative der Alternativenmenge dominiert, als **lokales Paretooptimum**.

Diese Begriffsdefinitionen weichen von denen in kontinuierlichen Optimierungsproblemen und Vektoroptimierungsproblemen ab, da man in diskreten (Vektor)optimierungsproblemen über keinen topologischen Umgebungsbegriff verfügt. \triangle

In Suchstrategie SS sind skalare diskrete Minimierungsprobleme (3.21) für verschiedene Referenzalternativen \bar{x} , nämlich

$$\min_{(\bar{x}, y) \in \{\bar{x}\} \times (R_D \setminus \{\bar{x}\})} Z[(\bar{x}, y)],$$

$$Z[(\bar{x}, y)] := \begin{cases} \min, & \text{falls } y \prec \bar{x}, \\ \text{mid}, & \text{falls } y \asymp \bar{x}, \\ \max, & \text{falls } y \succ \bar{x}, \end{cases} \quad (3.21)$$

in Folge zu lösen. Wir müssen bei den Problemen (3.21) mit drei Eigenschaften rechnen:

- Außer den Zielfunktionswerten steht über Z keine Information zur Verfügung.
- Es gibt sehr viele lokale (alle Alternativenpaare des Suchraumes mit Zielwert *mid*) und viele verschiedene globale Minimierer.
- Für jeden Referenzpunkt $\bar{x} \in R_D$ kann der Suchraum $\{\bar{x}\} \times (R_D \setminus \{\bar{x}\})$ sehr groß sein (theoretisch sogar unendlich, in der Numerik immer endlich).

Diese drei Eigenschaften weisen auf die Nützlichkeit eines stochastischen diskreten Verfahrens hin, da ein solches nicht alle Punkte des Suchraumes anlaufen muß, lediglich mit Zielfunktionswerten arbeitet und eventuell per Zufall auch aus lokalen wie globalen Minima wieder herausfindet, um (weitere) globale Minima zu suchen. Die bekanntesten der stochastischen diskreten skalaren Optimierungstechniken sind **Evolutionäre Algorithmen** und der **Metropolis Algorithmus** als eine Art Spezialfall unter ihnen. Sie alle haben die Vorteile, daß

- sie Ergebnisse mehr oder weniger unabhängig von der Wahl des Startpunktes / der Startpunkte liefern und
- im Laufe der Suche auch Verschlechterungen im Zielfunktionswert akzeptieren, d.h. auch Punkte im Suchraum anlaufen, deren Zielfunktionswerte höher als die der bereits aufgesuchten Punkte sind. Dies verhindert das Hängenbleiben in einem gefundenen lokalen oder globalen Minimierer.

Ein weiterer wichtiger Aspekt liegt in der Frage, ob es möglich ist, das Erreichen der globalen Minima durch den verwendeten stochastischen Algorithmus mit einer bestimmten Wahrscheinlichkeit vorherzusagen. Hier liegt ein Schwachpunkt der Evolutionären Algorithmen: Obwohl die Gesetze ihrer Suchprozeduren die Entdeckung vielversprechender Punkte des Suchraumes fördern, kann man keinerlei konkrete Wahrscheinlichkeitsaussagen über ihre Konvergenz gegen die gewünschte Punktmenge machen. Wie spätere Abschnitte verdeutlichen werden, erlaubt hingegen der Metropolis Algorithmus eine Art Konvergenzaussagen.

All die eben genannten Argumente sprechen dafür, den Metropolis Algorithmus als Löser der Probleme (3.21) und somit als Werkzeug der Suchstrategie SS einzusetzen.

3.2.5 Markov Ketten

Der **Metropolis Algorithmus** ist eine stochastische Methode, die in abzählbaren Suchräumen nach globalen Minimierern einer skalaren Zielfunktion sucht. Wie bereits erwähnt, akzeptiert er nicht nur Punkte mit einem besseren Zielfunktionswert als die Zielfunktionswerte der zuvor entdeckten Punkte des Suchraumes, sondern erlaubt auch mit einer Wahrscheinlichkeit größer null aber kleiner eins, Punkte mit einem schlechteren Zielfunktionswert anzulaufen. Dies ermöglicht dem Metropolis Algorithmus, lokale Minima wieder zu verlassen, um in weiteren Suchschritten zu einem globalen Minimum zu gelangen.

Wir wollen den **Suchraum**, den der Metropolis Algorithmus nach globalen Minimierern erforscht, S nennen. S sei im Rahmen dieser Arbeit **endlich**. Die Suchprozedur des Metropolis Algorithmus besteht aus einer Folge von Zufallsexperimenten. Jedes einzelne dieser Zufallsexperimente beschreibt den Übergang von einem Standpunkt $x_{t-1} \in S, t \in \mathbb{N}$, zum nächsten zu betrachtenden Punkt $x_t \in S$. Dieser Übergang werde von der diskreten Zufallsvariable X_t verkörpert, deren Realisierungen Elemente aus S sind. Der stochastische Prozeß $\{X_t\}_{t \in \mathbb{N}_0}$ (Folge der Zufallsexperimente $X_t, t \in \mathbb{N}_0$) repräsentiert die Handlungen des **idealisierten Metropolis Algorithmus**, der unendlich viele Übergänge vollführen darf, $\{x_t\}_{t \in \mathbb{N}_0}, x_t \in S$, die Folge von entdeckten und untersuchten Punkten innerhalb eines Suchlaufes. $\{X_t\}_{t \in \mathbb{N}_0}$ hat spezielle Eigenschaften - der stochastische Prozeß ist eine **Markov Kette**.

Die im Folgenden präsentierten Auszüge aus der Theorie der Markov Ketten basieren auf [19].

3.2.5.1 Definition der Markov Ketten

In stochastischen Prozessen ist die Zukunft nicht eindeutig festgelegt, aber Wahrscheinlichkeiten für bestimmte Zusammenhänge ermöglichen uns gewisse Vorhersagen. Bei den jetzt betrachteten stochastischen Prozessen, den Markov Ketten, hängen Wahrscheinlichkeitsverhältnisse der Zukunft lediglich vom gegenwärtigen Zustand ab und nicht davon, wie der gegenwärtige Zustand aus der Vergangenheit entstanden ist.

Definition 3.2.4. Eine Folge diskreter Zufallsvariablen $\{Y_t\}_{t \in \mathbb{Z}}$ ist ein **Markov Prozeß** (eine Markov Kette), falls für jede endliche Menge ganzer Zahlen $t_1 < t_2 < \dots < t_k, k \in \mathbb{N}$, die bedingte Wahrscheinlichkeit für das Ergebnis $Y_{t_k} = y_k$ unter der Bedingung $Y_{t_1} = y_1, \dots, Y_{t_{k-1}} = y_{k-1}$ identisch ist mit der bedingten Wahrscheinlichkeit für $Y_{t_k} = y_k$ unter der Bedingung $Y_{t_{k-1}} = y_{k-1}$:

$$P\{Y_{t_k} = y_k | Y_{t_1} = y_1, \dots, Y_{t_{k-1}} = y_{k-1}\} = P\{Y_{t_k} = y_k | Y_{t_{k-1}} = y_{k-1}\}. \quad (3.22)$$

△

Folglich erfüllt die Markov Kette $\{X_t\}_{t \in \mathbb{N}_0}$ für jedes $k \in \mathbb{N}$ die Gleichung

$$P\{X_k = x_k | X_0 = x_0, \dots, X_{k-1} = x_{k-1}\} = P\{X_k = x_k | X_{k-1} = x_{k-1}\}. \quad (3.23)$$

Wir wollen zwei Elemente $i \neq j$ aus S betrachten. Die Markov Kette, die den Metropolis Algorithmus modelliert, hat die zusätzliche Eigenschaft, daß die **Übergangswahrscheinlichkeit** $P\{X_k = j | X_{k-1} = i\}, k \in \mathbb{N}$, **zeithomogen (stationär)** ist, d.h. es gilt

$$P_{ij}(k) \equiv P_{ij} := P\{X_k = j | X_{k-1} = i\} \quad \forall k \in \mathbb{N}. \quad (3.24)$$

Alle weiteren Aussagen über **Markov Prozesse** beziehen sich auf **zeithomogene mit diskretem Zeitindex** $t \in \mathbb{N}_0$, bestehend aus **diskreten Zufallsvariablen** mit **Realisierungen in**

S. Einige Aussagen mögen auch für eine allgemeinere Form von Markov Ketten gelten, was jedoch über den Rahmen dieser Arbeit hinausgeht.

Für jeden Punkt $i \in S$ gibt es eine Wahrscheinlichkeit, mit der er Ergebnis des ersten Zufallsexperimentes X_0 der Markov Kette werden kann:

$$P_i := P\{X_0 = i\}. \quad (3.25)$$

Die Wahrscheinlichkeit für das Ergebnis $(x_0, x_1, \dots, x_k) = (i_0, i_1, \dots, i_k) \in S^k$, $k \in \mathbb{N}$, einer Folge $\{X_t\}_{0 \leq t \leq k}$ der Länge k von Zufallsexperimenten in der Markov Kette $\{X_t\}_{t \in \mathbb{N}_0}$ lautet:

$$P\{X_0 = i_0, X_1 = i_1, \dots, X_k = i_k\} = P_{i_0} P_{i_0 i_1} \dots P_{i_{k-1} i_k}. \quad (3.26)$$

Da sich die Ausgänge $X_0 = j$ ($X_k = j \wedge X_{k-1} = i$, $k \in \mathbb{N}$) für unterschiedliche $j \in S$ gegenseitig ausschließen, aber ein Anfangszustand (für jedes $i \in S$ ein Folgezustand) der Markov Kette existieren muß, gilt offensichtlich

$$\sum_{j \in S} P_j = 1, \quad (3.27)$$

$$\sum_{j \in S} P_{ij} = 1 \quad \forall i \in S. \quad (3.28)$$

Indem man die Elemente von S durchnumeriert (S ist ja abzählbar, sogar endlich), läßt sich die **Wahrscheinlichkeitsverteilung des Anfangszustandes der Markov Kette** als $|S|$ -Vektor niederschreiben:

$$P_{init} = (P_1, P_2, P_3, \dots). \quad (3.29)$$

Alle Übergangswahrscheinlichkeiten, d.h. bedingte Wahrscheinlichkeiten eines Zustandes zu einem Zeitpunkt unter der Bedingung eines bestimmten unmittelbar vorhergehenden Zustandes, können in einer $|S| \times |S|$ -Matrix zusammengefaßt werden:

$$P_{trans} = \begin{pmatrix} P_{11} & P_{12} & P_{13} & \dots \\ P_{21} & P_{22} & P_{23} & \dots \\ P_{31} & P_{32} & P_{33} & \dots \\ \dots & & & \dots \end{pmatrix}. \quad (3.30)$$

P_{init} ist ein stochastischer Vektor, weil sich seine Komponenten auf 1 aufsummieren. Die **Übergangsmatrix** P_{trans} ist eine stochastische Matrix, da je Zeile die Summe über alle Komponenten 1 ergibt.

3.2.5.2 Höhere Übergangswahrscheinlichkeiten

Bei der Beschreibung **höherer Übergangswahrscheinlichkeiten** soll P_{ij}^n für alle $i, j \in S$ und $n \in \mathbb{N}$ die Wahrscheinlichkeit darstellen, den Punkt $j \in S$ vom Punkt $i \in S$ aus in n Schritten zu erreichen. Induktion führt zur folgenden Identität:

$$P_{ij}^{m+n} = \sum_{l \in S} P_{il}^m P_{lj}^n \quad \forall i, j \in S, m, n \in \mathbb{N}. \quad (3.31)$$

Auch die höheren Übergangswahrscheinlichkeiten können in einer stochastischen Matrix arrangiert werden:

$$P_{trans}^n = \begin{pmatrix} P_{11}^n & P_{12}^n & P_{13}^n & \dots \\ P_{21}^n & P_{22}^n & P_{23}^n & \dots \\ P_{31}^n & P_{32}^n & P_{33}^n & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix}. \quad (3.32)$$

Aus Gleichung (3.31) folgt

$$P_{trans}^n = (P_{trans})^n, P_{trans}^{m+n} = P_{trans}^m P_{trans}^n. \quad (3.33)$$

Die einzigen absoluten Wahrscheinlichkeiten, die bisher eingeführt wurden, bilden den Vektor P_{init} und betreffen die möglichen Anfangszustände der Markov Kette. Weiter von Interesse sind für alle $j \in S$ die **(unbedingten) Wahrscheinlichkeiten, einen Punkt aus S nach $n \in \mathbb{N}$ Übergängen zu erreichen**:

$$P_j^n = \sum_{i \in S} P_i P_{ij}^n. \quad (3.34)$$

Mit dem stochastischen Vektor

$$(P_{abs}^n) = (P_1^n, P_2^n, P_3^n, \dots) \quad (3.35)$$

erhalten wir:

$$P_{abs}^n = P_{init} P_{trans}^n. \quad (3.36)$$

Wenn die Markov Kette bestimmte Eigenschaften hat, verliert die Verteilung der Anfangszustände P_{init} bei wachsendem n mehr und mehr ihren Einfluß auf P_{abs}^n , wie später noch gezeigt wird. Das bedeutet, daß für alle $j \in S$ eine reelle Zahl $0 < q_j < 1$ existiert mit $\lim_{n \rightarrow \infty} P_{ij}^n = q_j \quad \forall i \in S$, also P_{trans}^n gegen eine Matrix konvergiert, deren Zeilen alle das Aussehen $(q_1, \dots, q_{|S|})$ haben.

3.2.5.3 Eigenschaften einer Markov Kette

Der Ausdruck **”Zustand j (der Markov Kette)”** umschreibt das Szenario, daß das Ergebnis des momentanen Zufallsexperimentes $X_t, t \in \mathbb{N}_0$, der Punkt $j \in S$ ist. Der Suchraum S besteht aus allen möglichen Zuständen der Markov Kette. Diese Zustände lassen sich in verschiedene Klassen untergliedern. Eine Markov Kette, die auf einer bestimmten Klasse von Zuständen definiert ist, hat bestimmte Eigenschaften.

Definition 3.2.5. Die **Periode** $p \in \mathbb{N}$ eines Zustandes $j \in S$ ist der größte gemeinsame Teiler aller $n \in \mathbb{N}$ mit $P_{jj}^n > 0$. Ist $p = 1$, dann heißt j **aperiodisch**, im Falle $p > 1$ **periodisch**. \triangle

Die Periodizität von $j \in S$ verhindert die Konvergenz von P_{ij}^n gegen eine reelle Zahl $0 < q_j < 1$ für alle $i \in S$. Man betrachte eine Periode $p > 1$ von j . Falls $\bar{n} \neq \lambda p, \bar{n} \in \mathbb{N}$, für alle $\lambda \in \mathbb{N}$ gilt, ist sicher $P_{jj}^{\bar{n}} = 0$, wohingegen $P_{ij}^{\bar{n}} > 0$ für jedes $i \neq j, i \in S$, möglich ist. Diese Tatsache gilt unabhängig davon, wie groß \bar{n} ist. Offensichtlich bleibt hier die Entwicklung von $\lim_{n \rightarrow \infty} P_{ij}^n$ von i beeinflußt.

Zur Einführung zweier weiterer Klassen von Zuständen benötigen wir zwei neue Bezeichnungen:

Notation 3.2.6. Die Wahrscheinlichkeit, in einem Prozeß, der in $i \in S$ startet, nach $n \in \mathbb{N}_0$ Übergängen erstmalig in den Zustand $j \in S$ zu gelangen, soll mit f_{ij}^n bezeichnet werden, wobei $f_{ij}^0 = 0$ für alle $i, j \in S$ gilt.

$$f_{ij} := \sum_{n=1}^{\infty} f_{ij}^n \quad (3.37)$$

steht daher für die Wahrscheinlichkeit, daß die Folge von Zufallsexperimenten der Markov Kette, die im Zustand $i \in S$ beginnt, irgendwann Zustand $j \in S$ erreichen wird. \triangle

Folgende Notation macht nur für $j \in S$ mit $f_{jj} = 1$ Sinn:

Notation 3.2.7.

$$\mu_j := \sum_{n=1}^{\infty} n f_{jj}^n, \mu_j \leq \infty, \quad (3.38)$$

ist die **mittlere Wiederkehrzeit** für den Punkt $j \in S$. \triangle

Bei Zuständen $j \in S$ ist von Interesse, ob die Rückkehr zu ihnen sicher ist oder nicht, und, falls ja, ob ihre mittlere Wiederkehrzeit μ_j endlich oder unendlich ist.

Definition 3.2.8. Ein Zustand $j \in S$ mit $f_{jj} = 1$ heißt **rekurrent**, mit $f_{jj} < 1$ **transient**. Ein rekurrenter Zustand $j \in S$ wird **null-rekurrent** genannt, falls $\mu_j = \infty$ ist. \triangle

Um von i unabhängige Grenzwerte $\lim_{n \rightarrow \infty} P_{ij}^n, i, j \in S$, zu erhalten, müssen wir aperiodische rekurrente Zustände j mit endlichem μ_j betrachten:

Definition 3.2.9. Ein aperiodischer rekurrenter Zustand $j \in S$ mit $\mu_j < \infty$ heißt **ergodisch**. \triangle

Das gleich zitierte Theorem weist auf die Beziehungen zwischen der Art eines Zustandes $j \in S$ und dem Verhalten der Übergangswahrscheinlichkeiten $P_{jj}^n, n \in \mathbb{N}_0$, hin.

Theorem 3.2.10. $j \in S$ ist genau dann transient, wenn

$$\sum_{n=0}^{\infty} P_{jj}^n < \infty. \quad (3.39)$$

In diesem Fall gilt

$$\sum_{n=1}^{\infty} P_{ij}^n < \infty \quad \forall i \in S. \quad (3.40)$$

$j \in S$ ist rekurrent mit unendlichem μ_j (null-rekurrent) genau dann, wenn

$$\sum_{n=0}^{\infty} P_{jj}^n = \infty, \quad \text{aber} \quad \lim_{n \rightarrow \infty} P_{jj}^n = 0 \quad (3.41)$$

erfüllt sind. Ist dies der Fall, dann gilt

$$\lim_{n \rightarrow \infty} P_{ij}^n = 0 \quad \forall i \in S. \quad (3.42)$$

Ein aperiodischer rekurrenter Zustand $j \in S$ ist ergodisch genau dann, wenn $\mu_j < \infty$ ist. In diesem Fall ergibt sich

$$\lim_{n \rightarrow \infty} P_{ij}^n = f_{ij} \mu_j^{-1} \quad \forall i \in S. \quad (3.43)$$

Die Klassifizierung der Zustände einer Markov Kette bringt drei wichtige Eigenschaften von Markov Ketten hervor:

Definition 3.2.11. Eine Markov Kette $\{X_t\}_{t \in \mathbb{N}_0}$ heißt **irreduzibel**, wenn jeder Zustand $j \in S$ der Markov Kette von jedem Zustand $i \in S$ aus in endlicher Zeit mit positiver Wahrscheinlichkeit erreichbar ist:

$$\forall i, j \in S \quad \exists n \in \mathbb{N} : P_{ij}^n > 0. \quad (3.44)$$

△

Definition 3.2.12. Sind alle $j \in S$, d.h. alle Zustände der Markov Kette, aperiodisch, dann nennt man die gesamte Markov Kette **aperiodisch**.

△

Definition 3.2.13. Sind alle $j \in S$, d.h. alle Zustände der Markov Kette, ergodisch, dann wird die gesamte Markov Kette als **ergodisch** bezeichnet.

△

3.2.5.4 Die stationäre Verteilung

Bei irreduziblen und ergodischen Markov Ketten kann man beweisen, daß P_{ij}^n für jedes $j \in S$ mit wachsendem n gegen einen Wert konvergiert, der von $i \in S$ unabhängig ist.

Theorem 3.2.14. In einer irreduziblen und ergodischen Markov Kette mit P_{trans} gemäß (3.30) und P_{trans}^n gemäß (3.32) existieren die Grenzwerte

$$q_j := \lim_{n \rightarrow \infty} P_{ij}^n \quad \forall i \in S \quad (3.45)$$

für alle $j \in S$ und sind vom Anfangszustand $i \in S$ unabhängig. Darüber hinaus sind für alle $j \in S$ die Grenzwerte $q_j > 0$ und erfüllen

$$\sum_{j \in S} q_j = 1, \quad (3.46)$$

$$q_j = \sum_{i \in S} q_i P_{ij} \quad \forall j \in S. \quad (3.47)$$

Umgekehrt lauten die Aussagen: Falls für eine irreduzible und aperiodische Markov Kette mit P_{trans} gemäß (3.30) und P_{trans}^n gemäß (3.32) Zahlen $q_j > 0$ existieren, welche die Gleichungen (3.46) und (3.47) erfüllen, dann sind alle Zustände der Markov Kette ergodisch, die q_j definieren sich für alle $j \in S$ aus (3.45) und es gilt

$$q_j = 1/\mu_j \quad \forall j \in S. \quad (3.48)$$

Definition 3.2.15. Eine Wahrscheinlichkeitsverteilung $q := (q_1, \dots, q_{|S|})$, die (3.47) erfüllt, heißt **stationäre Verteilung** (der gegebenen Markov Kette).

△

3.2.5.5 Der Metropolis Algorithmus

Wie bereits erwähnt, können die Handlungen des idealisierten Metropolis Algorithmus (unendlich viele Übergänge je Suchlauf) im Suchraum S mit Hilfe einer zeithomogenen Markov Kette $\{X_t\}_{t \in \mathbb{N}_0}$, deren mögliche Zustände alle Elemente aus S sind, modelliert werden. Theorem 3.2.14 ermöglicht es uns, Wahrscheinlichkeitsaussagen über das Konvergenzverhalten des Algorithmus zu machen. Eine gute Einführung in den Metropolis Algorithmus ist z.B. in [1] zu finden; hiernach orientieren sich auch die Definitionen und Sätze dieses Abschnittes.

Bemerkung 3.2.16. Der Name "Metropolis Algorithmus" wird in dieser Arbeit für zwei verschiedene Konstrukte verwendet:

- als Name für einen "idealisierten" Metropolis Algorithmus mit unendlich vielen Übergängen; sein Suchlauf durch S kann durch die Markov Kette $\{X_t\}_{t \in \mathbb{N}_0}$ modelliert werden. Ist diese Markov Kette irreduzibel und ergodisch, so besitzt sie nach Theorem 3.2.14 eine stationäre Verteilung.

Wir wollen diese Variante des Metropolis Algorithmus mit der Abkürzung "IMA" kennzeichnen.

- als Name für einen "numerischen" Metropolis Algorithmus mit endlich vielen Übergängen $\bar{t} \in \mathbb{N}$; seine Suche in S läßt sich mit Hilfe der ersten $\bar{t} + 1$ Experimente der Markov Kette $\{X_t\}_{t \in \mathbb{N}_0}$ modellieren, nämlich mit $\{X_t\}_{0 \leq t \leq \bar{t}}$. Falls $\{X_t\}_{t \in \mathbb{N}_0}$ eine stationäre Verteilung $(q_1, \dots, q_{|S|})$ besitzt, nehmen wir von \bar{t} an, daß es groß genug sei, um für alle $i \in S$ die Identität $P_i^{\bar{t}} = P\{X_{\bar{t}} = i\} = q_i$ zu rechtfertigen.

Diese Variante des Metropolis Algorithmus soll mit der Abkürzung "NMA" gekennzeichnet werden.

Da $\{X_t\}_{0 \leq t \leq \bar{t}}$ NMA beschreibt und $\{X_t\}_{t \in \mathbb{N}_0}$ IMA, interpretieren wir NMA als **Ausschnitt von IMA**. △

Bemerkung 3.2.17. Jedes Zufallsexperiment $X_{t+1}, t \in \mathbb{N}_0$, das innerhalb eines Metropolis Suchlaufes (IMA und NMA) den Übergang von einem Punkt $x_t \in S$ zu einem Punkt $x_{t+1} \in S$ beschreibt, ist eine Kombination aus

1. einem Generierungsmechanismus und
2. einem Test auf Akzeptanz,

beides wiederum Zufallsexperimente. △

Vom jeweils aktuellen Standpunkt $x_t \in S, t \in \mathbb{N}_0$, der Suchprozedur aus wird ein **Nachbar** $x \in S$ zufällig als neuer Standpunktkandidat generiert. Dieses Generierungsexperiment folgt einer bestimmten vom Anwender des Metropolis Algorithmus im Voraus zu bestimmenden Wahrscheinlichkeitsverteilung. "Nachbar" bedeutet, daß x aus einer besonderen Teilmenge $N_{x_t} \subseteq S$ des Suchraumes stammt, die in gewisser Weise die "nähere Umgebung" von x_t beschreibt, der sogenannten **Nachbarschaft** von x_t . N_{x_t} enthält alle Punkte $x \neq x_t$ aus S , die für den Algorithmus von x_t aus in einem Schritt mit positiver Wahrscheinlichkeit erreichbar sind. Die **Nachbarschaftsstruktur** muß definiert werden, bevor die Sucharbeit beginnt, d.h.

es muß im Vorhinein festgelegt werden, welche Punkte von jedem beliebigen $x \in S$ aus als Kandidaten für einen neuen Standpunkt des Suchlaufes vorgeschlagen werden können, sobald x der aktuelle Standpunkt ist.

Bemerkung 3.2.18. Wie der Anwender diese Nachbarschaftsstruktur genau zu definieren hat, ist problemabhängig. Beispiele für problemspezifische Nachbarschaftsstrukturen werden in Kapitel 4 gegeben. \triangle

Sobald vom momentanen Standpunkt x_t aus ein neuer Standpunktkandidat $x \in N_{x_t}$ vorgeschlagen wurde, muß überprüft werden, ob der Algorithmus ihn als neuen Standpunkt akzeptiert. Diese Prüfung auf Akzeptanz läuft wieder in Form eines Zufallsexperimentes ab. Diesem liegt eine Wahrscheinlichkeitsverteilung zugrunde, die von den Zielwerten $Z(x_t)$ und $Z(x)$ abhängt, wobei $Z : S \mapsto \mathbb{R}, x \mapsto Z(x)$, die Bewertungs- bzw. Zielfunktion des Suchraumes S sei, die der Metropolis Algorithmus (IMA und NMA) optimiert.

Die bisherigen Aussagen über den Generierungsmechanismus des Metropolis Algorithmus für jeweils neue Standpunktkandidaten lassen sich in folgender Definition zusammenfassen:

Definition 3.2.19. Während seiner Suche in S sei der Metropolis Algorithmus (IMA und NMA) im Punkt $i \in S$ als aktueller Standpunkt angekommen. Die Wahrscheinlichkeit, mit der ein Punkt $j \in S$ als Kandidat für einen neuen Standpunkt von $i \in S$ aus vorgeschlagen bzw. generiert wird, heißt **Generierungswahrscheinlichkeit** G_{ij} .

Alle $j \in S$ mit $G_{ij} > 0$ werden als **Nachbarn von i** bezeichnet. Sie bilden die Menge $N_i \subseteq S$, die **Nachbarschaft von i** . \triangle

Da die Markov Kette, die das Suchverhalten des Metropolis Algorithmus (IMA und NMA) modelliert, zeithomogen ist, hängt G_{ij} für alle $i, j \in S$ nicht vom Zeitpunkt $t \in \mathbb{N}_0$ ab, zu dem i der aktuelle Standpunkt ist. Da die Nachbarschaftsstruktur, die dem Metropolis Algorithmus zugrunde liegt, vor dem Start der Suchprozedur zu definieren ist, muß G_{ij} für alle $i, j \in S$ a priori festgelegt werden, denn die **Gesamtheit aller $G_{ij}, i, j \in S$, bildet die Nachbarschaftsstruktur**. Um später für eine bestimmte Wahrscheinlichkeitsverteilung nachweisen zu können, daß sie die stationäre Verteilung der Markov Kette ist, die den Metropolis Algorithmus (IMA) modelliert, benötigen wir außerdem

$$G_{ij} = G_{ji} \quad \forall i, j \in S, \quad (3.49)$$

also die Symmetrie der Generierungswahrscheinlichkeiten. Daraus ergibt sich

$$j \in N_i \iff i \in N_j. \quad (3.50)$$

Das Zufallsexperiment, das immer nach dem Generierungsexperiment durchgeführt wird, entscheidet, ob das vorgeschlagene $j \in S$ als neuer Standpunkt des Algorithmus akzeptiert wird:

Definition 3.2.20. Sei $i \in S$ der aktuelle Standpunkt des Metropolis Algorithmus (IMA und NMA) im Suchraum S und $j \in N_i$ ein neuer Standpunktkandidat, der mit Wahrscheinlichkeit G_{ij} vorgeschlagen wurde. Die Wahrscheinlichkeit, daß j als neuer Standpunkt akzeptiert wird, heißt **Akzeptanzwahrscheinlichkeit** und ist definiert durch

$$A_{ij}(c) := \exp\left(-\frac{(Z(j) - Z(i))^+}{c}\right), c \in \mathbb{R}^+ \quad \forall i \in S, j \in N_i, \quad (3.51)$$

wobei gilt

$$(Z(j) - Z(i))^+ := \begin{cases} Z(j) - Z(i), & \text{falls } Z(j) - Z(i) > 0, \\ 0, & \text{sonst.} \end{cases} \quad (3.52)$$

Bei allen $j \notin N_i$ wird $A_{ij}(c) = 0$ festgelegt - für solche j wird sich die Akzeptanzfrage nie stellen, da $G_{ij} = 0$ ist. \triangle

Die Konstante $c \in \mathbb{R}^+$ wird meist **Kontrollparameter** des Metropolis Algorithmus (IMA und NMA) genannt. Liegt $Z(j) \leq Z(i)$, also eine Verbesserung oder keine Veränderung des Zielfunktionswertes $Z(\cdot)$ vor, so kann der Kontrollparameter den Wert von $A_{ij}(c)$ nicht beeinflussen, da der Zähler im Exponenten gleich null ist. Ein Punkt, der im Zielfunktionswert nicht schlechter als der momentane Standpunkt abschneidet, wird sicherlich als neuer Standpunkt akzeptiert. Ist jedoch $Z(j) > Z(i)$, so gilt $0 < A_{ij}(c) < 1$. Wie nahe $A_{ij}(c)$ dann bei null oder eins liegt, hängt von c ab: Je kleiner c ist, desto näher ist der Wert $A_{ij}(c)$ der Null und umso weniger wahrscheinlich wird die Verschlechterung von $Z(i)$ auf $Z(j)$ akzeptiert. Ein größeres c macht die Hinnahme eines schlechteren Zielwertes $Z(j)$ wahrscheinlicher.

In der Wahl von $c \in \mathbb{R}^+$ liegen die Chancen und Gefahren des Metropolis Algorithmus (IMA und NMA). Indem er im Vergleich zu bereits entdeckten Punkten des Suchraumes auch Verschlechterungen in der Leistung (=Zielfunktionswert) erlaubt, bekommt der Algorithmus die Chance, aus "Tälern" von lokalen Minima, in die er hinabgestiegen ist, wieder "bergaufzusteigen" und weiterzuwandern, bis er die "Täler" der globalen Minima gefunden hat. Je größer die zu akzeptierenden Zielwertverschlechterungen sind, um aus einem Minimum wieder herauszukommen, desto mehr muß der Algorithmus annehmen, in oder nahe einem globalen Minimum zu sein (je höher der "Berg", desto tiefer das "Tal"). In Übereinstimmung damit ist für festes $c \in \mathbb{R}^+$ die Akzeptanzwahrscheinlichkeit $A_{ij}(c)$ umso kleiner, je größer $(Z(j) - Z(i))^+$ ist. Wird der Kontrollparameter zu klein gewählt, könnte die Chance, lokalen Minima zugunsten der Weitersuche nach globalen Minima wieder zu entkommen, fast zerstört werden. Ein zu großes c hingegen beraubt den Algorithmus seiner Tendenz, bessere Zielwerte den schlechteren vorzuziehen und damit all seiner Steuerung.

Wir fassen zusammen: Der Übergang von einem Standpunkt $i \in S$ zu einem Punkt $j \in S$ innerhalb eines Metropolis-Suchlaufs (IMA und NMA) ist das Ergebnis aus der Hintereinanderausführung eines Generierungsexperiments, d.h. der Zufallserzeugung eines neuen Standpunktkandidaten $j \in N_i$, und eines Akzeptanzexperiments, d.h. der zufallsgesteuerten Entscheidung, ob der Kandidat $j \in S$ als neuer Standpunkt akzeptiert wird. Hieraus ergibt sich die Definition der Übergangswahrscheinlichkeit:

Definition 3.2.21. Sei $i \in S$ der momentane Standpunkt des Metropolis Algorithmus (IMA und NMA). Die **Übergangswahrscheinlichkeit** von i zu einem beliebigen Punkt $j \in S$ ist für ein $c \in \mathbb{R}^+$ gegeben durch

$$(P_{trans})_{ij}(c) = P_{ij}(c) := \begin{cases} G_{ij}A_{ij}(c), & \text{falls } i \neq j, \\ 1 - \sum_{l \in S, l \neq i} P_{il}(c), & \text{falls } i = j. \end{cases} \quad (3.53)$$

\triangle

(3.53) entspricht für alle $t \in \mathbb{N}$ der Übergangswahrscheinlichkeit $P\{X_t = j | X_{t-1} = i\}$, es ist die Metropolis-Definition von (3.24). Die Schreibweise in Abhängigkeit von c betont den

Einfluß des Kontrollparameters auf das Verhalten der speziellen Markov Kette $\{X_t\}_{t \in \mathbb{N}_0}$, die den Metropolis Algorithmus (IMA) modelliert.

Alle drei definierten Wahrscheinlichkeitsverteilungen füllen jeweils eine $|S| \times |S|$ -Matrix:

- $\mathcal{G} := (G_{ij})_{i,j \in S}$ ist die **Generierungsmatrix**,
- $\mathfrak{A}(c) := (A_{ij}(c))_{i,j \in S}$ ist die **Akzeptanzmatrix**,
- $P_{trans}(c) = \mathfrak{P}(c) := (P_{ij}(c))_{i,j \in S}$ ist die **Übergangsmatrix**.

\mathcal{G} und $\mathfrak{P}(c)$ sind stochastisch.

Die zeithomogene Markov Kette $\{X_t\}_{t \in \mathbb{N}_0}$ mit $P\{X_t = j | X_{t-1} = i\} = P_{ij}(c) \quad \forall i, j \in S$ aus (3.53) modelliert für festes $c \in \mathbb{R}^+$ den Suchlauf des Metropolis Algorithmus (IMA), der wie beschrieben Punkt zu Punkt Übergänge vollführt und dabei den Wahrscheinlichkeiten in Matrix $\mathfrak{P}(c)$ gehorcht. Falls $\{X_t\}_{t \in \mathbb{N}_0}$ irreduzibel und aperiodisch ist und reelle Zahlen $q_i > 0$ existieren, die für alle $i \in S$ die Gleichungen (3.46) und (3.47) erfüllen, dann weiß man durch Theorem 3.2.14, daß die Markov Kette ergodisch ist und die Zahlen $q_i, i \in S$, die stationäre Verteilung bilden.

Kandidaten für solche Zahlen sind:

$$q_i(c) := \frac{1}{N_0(c)} \exp\left(-\frac{Z(i)}{c}\right) \quad \forall i \in S, c \in \mathbb{R}^+,$$

$$N_0(c) := \sum_{j \in S} \exp\left(-\frac{Z(j)}{c}\right).$$
(3.54)

Satz 3.2.22. *Der Metropolis Algorithmus (IMA), der S durchsucht, werde von einer Markov Kette mit Übergangsmatrix $P_{trans}(c) = \mathfrak{P}(c)$ für einen festen Kontrollparameter $c \in \mathbb{R}^+$ modelliert. $\mathfrak{P}(c)$ sei gemäß Definition 3.2.19, Gleichung (3.49), Definition 3.2.20 und Definition 3.2.21 definiert. Ist die Markov Kette irreduzibel und aperiodisch, dann ist die Wahrscheinlichkeitsverteilung $q(c) := (q_1(c), \dots, q_{|S|}(c)), c \in \mathbb{R}^+$, mit*

$$q_i(c) := \frac{1}{N_0(c)} \exp\left(-\frac{Z(i)}{c}\right) \quad \forall i \in S,$$

$$N_0(c) := \sum_{j \in S} \exp\left(-\frac{Z(j)}{c}\right),$$
(3.55)

die stationäre Verteilung der gegebenen Markov Kette.

Beweis: Nach Theorem 3.2.14 reicht es zu überprüfen, ob der Kandidat $q(c)$ den Gleichungen (3.46) und (3.47) genügt.

$$\sum_{i \in S} q_i(c) = \frac{1}{N_0(c)} \left[\sum_{i \in S} \exp\left(-\frac{Z(i)}{c}\right) \right] = \frac{N_0(c)}{N_0(c)} = 1,$$
(3.56)

d.h. Gleichung (3.46) ist erfüllt. Falls $\forall i, j \in S$ gilt

$$q_i(c)P_{ij}(c) = q_j(c)P_{ji}(c), \quad (3.57)$$

dann erfüllen die $q_i(c)$ für alle $i \in S$ Gleichung (3.47):

$$q_i(c)P_{ij}(c) = q_j(c)P_{ji}(c) \quad (3.58)$$

$$\implies \sum_{j \in S} q_i(c)P_{ij}(c) = \sum_{j \in S} q_j(c)P_{ji}(c) \iff q_i(c) = \sum_{j \in S} q_j(c)P_{ji}(c). \quad (3.59)$$

Mit den Voraussetzungen aus Satz 3.2.22 ist die Gültigkeit von (3.57) leicht nachzuweisen:

Da $G_{ij} = G_{ji} \quad \forall i, j \in S$ gilt, reduziert sich (3.57) auf

$$q_i(c)A_{ij}(c) = q_j(c)A_{ji}(c) \quad \forall i, j \in S, i \neq j. \quad (3.60)$$

$$\begin{aligned} q_i(c)A_{ij}(c) &= \frac{1}{N_0} \exp\left(-\frac{Z(i)}{c}\right) \exp\left(-\frac{(Z(j) - Z(i))^+}{c}\right) \\ &= \frac{1}{N_0} \exp\left(-\frac{Z(j)}{c}\right) \exp\left(-\frac{(Z(i) - Z(j)) + (Z(j) - Z(i))^+}{c}\right) \\ &= \frac{1}{N_0} \exp\left(-\frac{Z(j)}{c}\right) \exp\left(-\frac{(Z(i) - Z(j))^+}{c}\right) \\ &= q_j(c)A_{ji}(c) \quad \forall j \in N_i. \end{aligned} \quad (3.61)$$

Für alle $j \in S \setminus N_i$ lautet (3.60) $q_i(c)A_{ij}(c) = 0 = q_j(c)A_{ji}(c)$.

Die Zahlen $q_i(c)$ genügen also auch Gleichung (3.47). ■

Vergleicht man unter allen Punkten $j \in S$ die Wahrscheinlichkeiten, mit denen sie nach unendlich vielen Übergängen in einem Metropolis-Suchlauf erreicht werden können, so fällt auf, daß die Wahrscheinlichkeiten für die Punkte mit dem kleinsten Zielfunktionswert, also für die globalen Minimierer, am größten sind. Gibt man dem Metropolis Algorithmus (NMA) Gelegenheit, ausreichend viele Übergänge $\bar{t} \in \mathbb{N}$ zu vollführen, so daß für alle $j \in S$ $q_j(c)$ gut genug durch $P_{ij}^{\bar{t}}(c) \quad \forall i \in S$ approximiert wird, so ist die Wahrscheinlichkeit, daß der Algorithmus nach diesen \bar{t} Übergängen in einem bestimmten globalen Minimierer landet, höher, als dafür, daß er in einem bestimmten Punkt aus S ankommt, der nicht zu den globalen Minimierern der Zielfunktion Z gehört.

3.2.6 Suchstrategie "Parallele Suche" (PS)

In Abschnitt 3.2.2 wurde eine Generierungsstrategie zur Lösung von Problem 3.20 vorgestellt: Man startet von einer beliebigen Referenzalternative $x_0 \in R_D$ aus und sucht nach einer Alternative $x_1 \in R_D$, von der sie dominiert wird. Wird so ein x_1 entdeckt, sucht man weiter nach einer Alternative $x_2 \in R_D$, die wiederum x_1 dominiert usw., bis man für eine Alternative $x_k \in R_D, k \in \mathbb{N}_0$, keine dominierenden Punkte mehr aufspüren kann. x_k wird dann als das erste gefundene Paretooptimum betrachtet. Die eben beschriebene Suche nach immer den Vorpunkt dominierenden Punkten läßt sich dann von einer weiteren Start-Referenzalternative $\hat{x}_0 \in R_D$ aus wiederholen, wobei \hat{x}_0 entweder zufällig aus $R_D \setminus \{x_0\}$ gewählt wird oder einfach der zweite Partner des

Ergebnispaares der Suche mit Referenzpunkt x_k dafür hergenommen wird. Indem man immer weitere Suchvorgänge wie beschrieben ausführt, also mehrmals über je einen Pfad dominierender Punkte zu jeweils einem Paretooptimum strebt, kann man viele verschiedene effiziente Alternativen aufsammeln und somit eine repräsentative Teilmenge der Effizienzmenge in R_D oder Approximation einer solchen erarbeiten. Bei jeder einzelnen Suche nach einem dominierenden Punkt für eine Referenzalternative \bar{x} ist ein skalares diskretes Minimierungsproblem der Form (3.21) mit $S := \{\bar{x}\} \times (R_D \setminus \{\bar{x}\})$ zu lösen.

Die stochastische Generierung einer Alternative aus R_D erfolgt jeweils in drei Schritten:

1. stochastische Generierung eines \bar{x}_D aus D ,
2. Lösung des zu \bar{x}_D gehörigen kontinuierlichen Subproblems (3.10) mit Hilfe eines kontinuierlichen Mehrzielverfahrens, das die numerische Näherung $\bar{L}^{\bar{x}_D}$ für die Lösungsmenge $L^{\bar{x}_D}$ von (3.10) erzeugt (sofern dieses Subproblem nicht schon während des Suchlaufes gelöst wurde; $\bar{L}^{\bar{x}_D}$ wird für weitere Verwendung gespeichert),
3. stochastische Wahl einer Alternative (\bar{x}_C, \bar{x}_D) aus der numerischen Näherung $\bar{E}^{\bar{x}_D} := \{\bar{x}\} \times \bar{L}^{\bar{x}_D}$ von $E^{\bar{x}_D}$.

Betrachten wir Problem (3.21) für die erste Referenzalternative $x_0 \in R_D$ näher. Wir gehen davon aus, daß der Metropolis Algorithmus (NMA), der das Problem bearbeitet, viele Übergänge von Paar zu Paar $(x_0, y) \in \{x_0\} \times (R_D \setminus \{x_0\})$ machen darf und alle im Laufe seiner Tätigkeit entdeckten Punkte $y \in R_D$ mit $y \prec x_0$ oder $y \succ x_0$ nach folgenden Regeln in einer Liste $L_0 \subseteq R_D$ speichert:

- Falls ein $\bar{y} \in R_D$ Kandidat für L_0 ist, jedoch von einem Element aus L_0 dominiert wird, wird \bar{y} nicht in L_0 aufgenommen.
- Ansonsten wird der Kandidat Mitglied von L_0 .
- Sollte das neu aufgenommene \bar{y} einen Listeneintrag $\hat{y} \in L_0$ dominieren, so wird \hat{y} aus L_0 entfernt.

Da der Metropolis Algorithmus (IMA und NMA) auch Verschlechterungen in den Zielfunktionswerten von Z hinnimmt, kann er ein Paar $(x_0, \bar{y}) \in \{x_0\} \times (R_D \setminus \{x_0\})$ mit $\bar{y} \prec x_0$ zugunsten von "schlechteren" Paaren im Suchraum $\{x_0\} \times (R_D \setminus \{x_0\})$ verlassen, um von diesen aus weitere Paare $(x_0, \hat{y}) \in \{x_0\} \times (R_D \setminus \{x_0\})$ mit $\hat{y} \prec x_0$ zu finden. Bei langer Laufzeit ($\bar{t} \in \mathbb{N}$ Übergänge) können auf diese Weise viele Paare mit Zielfunktionswert *min* oder *mid* aufgesammelt werden. Daher füllt sich L_0 mit sehr vielen oder eventuell sogar allen effizienten Alternativen. Die Nachbarschaftsstruktur des Suchraumes von Problem (3.21) sollte natürlich so gewählt werden, daß die Markov Kette $\{X_t\}_{t \in \mathbb{N}_0}$, die den IMA modelliert und deren erste $\bar{t} + 1$ Experimente $\{X_t\}_{0 \leq t \leq \bar{t}}$ das Suchverhalten des NMA beschreiben, irreduzibel und ergodisch ist. Dann existiert eine stationäre Verteilung, zusammengesetzt aus den Zahlen

$$q_{(x_0, y)}(c) = \frac{\exp\left(-\frac{Z[(x_0, y)]}{c}\right)}{\sum_{z \in R_D \setminus \{x_0\}} \exp\left(-\frac{Z[(x_0, z)]}{c}\right)}, (x_0, y) \in S_0 := \{x_0\} \times (R_D \setminus \{x_0\}). \quad (3.62)$$

Drei neue Bezeichnungen sind im weiteren Gebrauch von Nutzen:

Notation 3.2.23. Betrachtet man einen Punkt $x \in R_D$, so erlaubt es das Prinzip der Dominanz, R_D in drei Teilmengen zu zerlegen:

- $Dom_x \subset R_D$ repräsentiere alle $y \in R_D$ mit $y \prec x$.
- $Ind_x \subseteq R_D$ enthalte alle $y \in R_D, y \neq x$, mit $y \asymp x$.
- $NDom_x := R_D \setminus (Dom_x \cup Ind_x \cup \{x\})$ sei schließlich die Menge aller Alternativen $y \succ x$.

△

Mit Hilfe dieser drei Bezeichnungen/Mengen ist es möglich, die drei verschiedenen Wahrscheinlichkeiten auszudrücken, die in der stationären Verteilung der Markov Kette vorkommen, die für einen Kontrollparameter $c \in \mathbb{R}^+$ den Metropolis Algorithmus (IMA) modelliert, dessen numerische Version NMA das Minimierungsproblem (3.21) mit Referenzalternative $x_0 \in R_D$ löst:

$$q_{(x_0,y)}(c) := \begin{cases} \frac{\exp\left(\frac{-min}{c}\right)}{|Dom_{x_0}| \exp\left(\frac{-min}{c}\right) + |Ind_{x_0}| \exp\left(\frac{-mid}{c}\right) + |NDom_{x_0}| \exp\left(\frac{-max}{c}\right)}, & \text{falls } y \prec x_0, \\ \frac{\exp\left(\frac{-mid}{c}\right)}{|Dom_{x_0}| \exp\left(\frac{-min}{c}\right) + |Ind_{x_0}| \exp\left(\frac{-mid}{c}\right) + |NDom_{x_0}| \exp\left(\frac{-max}{c}\right)}, & \text{falls } y \asymp x_0, \\ \frac{\exp\left(\frac{-max}{c}\right)}{|Dom_{x_0}| \exp\left(\frac{-min}{c}\right) + |Ind_{x_0}| \exp\left(\frac{-mid}{c}\right) + |NDom_{x_0}| \exp\left(\frac{-max}{c}\right)}, & \text{falls } y \succ x_0 \end{cases} \quad (3.63)$$

$\forall (x_0, y) \in S_0$.

Nehmen wir an, $\underline{t} \in \mathbb{N}$ sei so groß, daß in etwa die Gleichung $P\{X_{\underline{t}} = (x_0, y)\} = q_{(x_0,y)}(c)$ für alle $(x_0, y) \in S_0$ stimmt, so besteht für unseren Metropolis Algorithmus bei insgesamt $\bar{t} > \underline{t}$ Übergängen begründete Hoffnung, daß er viele effiziente Alternativen in L_0 anhäuft, da ja gilt

$$\begin{aligned} q_{(x_0,y)}(c) & \text{ mit } y \prec x_0 \\ & > \\ q_{(x_0,y)}(c) & \text{ mit } y \asymp x_0 \\ & > \\ q_{(x_0,y)}(c) & \text{ mit } y \succ x_0 \\ & \forall (x_0, y) \in S. \end{aligned} \quad (3.64)$$

Befindet sich die Markov Kette $\{X_t\}_{t \in \mathbb{N}_0}$, die den IMA modelliert, im Experiment $X_{\underline{t}}$ fast in ihrem stationären Gleichgewicht, d.h. liegt für ihre möglichen Zustände zum Zeitpunkt \underline{t} in etwa die stationäre Verteilung zugrunde, so läuft die Metropolis-Suchprozedur (NMA) in den Experimenten $X_{k\underline{t}}, k \in \{l \in \mathbb{N} | l \leq \bar{t}\}$, mit größerer Wahrscheinlichkeit jeweils einen einzelnen Punkt aus Dom_{x_0} als einen einzelnen Punkt aus Ind_{x_0} und wiederum mit größerer Wahrscheinlichkeit einen einzelnen Punkt aus $Dom_{x_0} \cup Ind_{x_0}$ als einen aus $NDom_{x_0}$ an. Enthalten die Mengen Dom_{x_0} und Ind_{x_0} zwar sehr viele Elemente, aber nur wenige Paretooptima, mag dies nicht gerade einen gewaltigen Vorteil bringen.

Selbst wenn die Länge des Metropolis-Suchlaufes (NMA) vielfach ein Herausfiltern nichteffizienter Alternativen aus L_0 ermöglichte, müßte L_0 deswegen noch lange keine gute Darstellung der Effizienzmenge in R_D sein, da die Paretooptima in Ind_{x_0} gegenüber denen in Dom_{x_0} in den Experimenten $X_{k\underline{t}}$, $k \in \{l \in \mathbb{N} | l \leq \bar{t}\}$, vernachlässigt werden. Nichtsdestotrotz sollte man die **Sammelqualitäten** des Metropolis Algorithmus (IMA und NMA) irgendwie in die Suchstrategie SS einbauen. Dies umzusetzen, ist auf folgende Art vorstellbar:

Statt wie in der sequenziellen Suchstrategie SS mit einer einzigen Referenzalternative $x_0 \in R_D$ zu beginnen, starte man mit $1 < s_0 \in \mathbb{N}$ Referenzpunkten $x_0^{i_0} \in R_D$, $i_0 = 1, \dots, s_0$, die in einer Liste L_0 zusammengefaßt werden. Sie werden bei Gleichverteilung zufällig aus R_D gewählt und sollen sich gegenseitig nicht dominieren. Für jedes $i_0 \in \{1, \dots, s_0\}$ bearbeite eine Metropolis-Prozedur $MA_0^{i_0}$ (NMA) das entsprechende Problem

$$\min_{(x_0^{i_0}, y) \in \{x_0^{i_0}\} \times (R_D \setminus \{x_0^{i_0}\})} Z[(x_0^{i_0}, y)]. \quad (3.65)$$

$\underline{t} \in \mathbb{N}$ sei wieder so groß, daß in dieser Iteration der Markov Ketten, die die IMAs der $MA_0^{i_0}$, $i_0 = 1, \dots, s_0$, beschreiben, fast die Gültigkeit der stationären Verteilung vorausgesetzt werden darf. Jedes $y \in R_D \setminus \{x_0^{i_0}\}$ aus einem Paar $(x_0^{i_0}, y)$ mit $y \prec x_0^{i_0}$, das $MA_0^{i_0}$ von Iteration $\underline{t} \in \mathbb{N}$ an aufgespürt hat, wird nach folgenden Regeln in eine Liste $L_1 \subseteq R_D$ aufgenommen, die von allen $MA_0^{i_0}$ (NMA), $i_0 = 1, \dots, s_0$, gespeist wird:

- Wird y von einem Mitglied der Liste L_1 dominiert, so wird y abgewiesen.
- Ansonsten wird y in L_1 aufgenommen.
- Falls y einen Eintrag \bar{y} von L_1 dominiert, wird dieses \bar{y} aus L_1 entfernt.

Man kann eine Anzahl $numcont \in \mathbb{N}$ von L_1 -Einträgen vorgeben, die jede Suchprozedur $MA_0^{i_0}$, $i_0 = 1, \dots, s_0$, mindestens für L_1 beizusteuern hat. Wann auch immer ein Neueintrag $y \in R_D \setminus \{x_0^{i_0}\}$, $i_0 \in \{1, \dots, s_0\}$, eine Alternative $\bar{y} \in L_1$, die von einer Suchprozedur $MA_0^{\bar{i}_0}$, $\bar{i}_0 \in \{1, \dots, s_0\}$, beigetragen wurde, "hinauswirft", muß $MA_0^{\bar{i}_0}$ weiter nach einem Ersatzbeitrag für L_1 suchen. Dies kann eventuell zu oft vorkommen, als daß $MA_0^{\bar{i}_0}$ den Stand von $numcont$ beigesteuerten Einträgen in L_1 bei vernünftigem Arbeitsaufwand erreichen oder halten könnte. Dann mögen die bereits durch $MA_0^{\bar{i}_0}$ erfolgten $r_{i_0}^{\bar{i}_0} \in \mathbb{N}_0$, $r_{i_0}^{\bar{i}_0} < numcont$, Eintragungen ausreichen. Die Suchläufe $MA_0^{i_0}$, $i_0 = 1, \dots, s_0$, werden beendet, sobald jede Prozedur $numcont$ Punkte $y \in R_D \setminus \{x_0^{i_0}\}$ (welche die entsprechende Referenzalternative $x_0^{i_0}$ dominieren) in L_1 etabliert hat oder die maximal erlaubte Rechenzeit überschritten ist. Es gibt zwei Gründe, warum keine Alternative von einem $MA_0^{\bar{i}_0}$, $\bar{i}_0 \in \{1, \dots, s_0\}$, in L_1 gespeichert ist:

- Wie bereits beschrieben, könnten Neueinträge in L_1 die von $MA_0^{\bar{i}_0}$ stammenden Mitglieder von L_1 wieder und wieder verdrängt haben.

- $MA_0^{\bar{x}_0}$ konnte vielleicht kein Paar $(x_0^{\bar{x}_0}, y) \in \{x_0^{\bar{x}_0}\} \times (R_D \setminus \{x_0^{\bar{x}_0}\})$ mit Zielfunktionswert \min , d.h. keine Alternative, die den Referenzpunkt $x_0^{\bar{x}_0}$ dominiert, finden. Dann darf man von $x_0^{\bar{x}_0}$ annehmen, daß es paretooptimal sei.

$L_1 \subseteq R_D$ ist eine Menge von Alternativen aus R_D mit drei wichtigen Eigenschaften:

- Innerhalb L_1 sind alle Alternativen "gleich gut" im Sinne der Dominanz, d.h. $\bar{y} \asymp \hat{y} \quad \forall \bar{y}, \hat{y} \in L_1$.
- Jedes $\bar{y} \in L_1$ dominiert mindestens ein $\bar{x} \in L_0$ - natürlich die Referenzalternative der Metropolis-Prozedur (NMA), die \bar{y} für L_1 hervorgebracht hat, aber eventuell zufällig auch andere Elemente von L_0 .
- Kein Eintrag $\bar{y} \in L_1$ wird von auch nur einem $\bar{x} \in L_0$ dominiert. Käme nämlich eine derartige Dominanz vor, würde \bar{x} neben \bar{y} auch die Referenzalternative von \bar{y} dominieren, was Dominanz innerhalb von L_0 bedeutete und daher unmöglich ist.

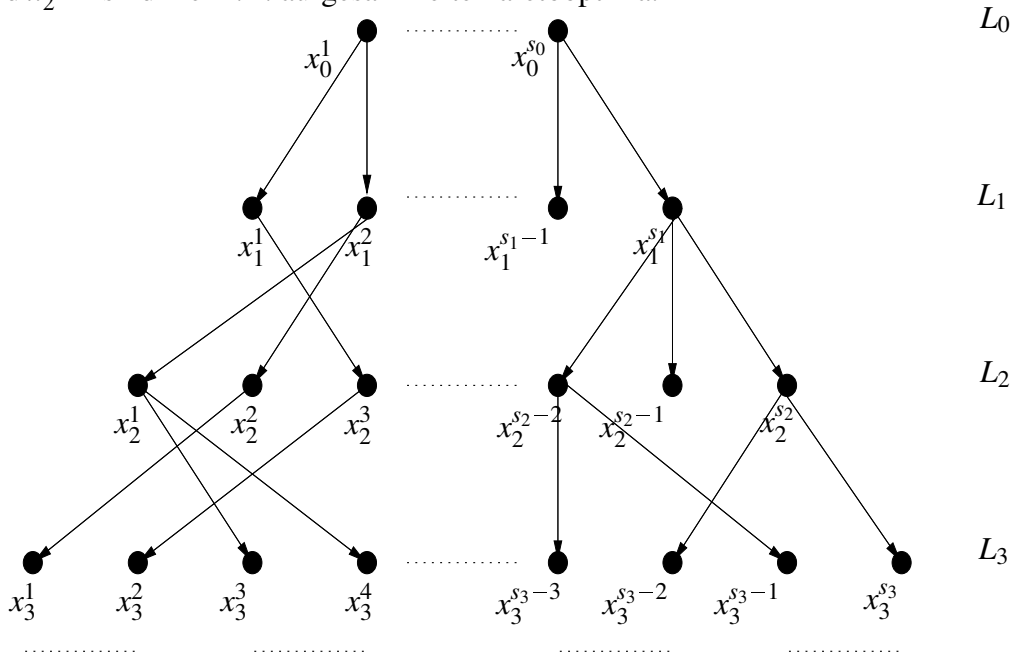
Die Liste L_1 ist also eine Teilmenge des Suchraumes R_D , die im Sinne der drei hervorgehobenen Eigenschaften "besser" als die Startmenge L_0 ist. Ihre Elemente $x_1^{i_1}, i_1 = 1, \dots, s_1, s_1 \in \mathbb{N}$, übernehmen jetzt die Rolle von Referenzalternativen für weitere Metropolis-Suchen $MA_1^{i_1}$ (NMA). Die Metropolis-Prozeduren $MA_1^{i_1}$ (NMA), $i_1 = 1, \dots, s_1$, bearbeiten die Probleme

$$\min_{(x_1^{i_1}, y) \in \{x_1^{i_1}\} \times (R_D \setminus \{x_1^{i_1}\})} Z[(x_1^{i_1}, y)]. \quad (3.66)$$

Analog zur eben beschriebenen Vorgehensweise der Prozeduren $MA_0^{i_0}, i_0 = 1, \dots, s_0$, füllen die $MA_1^{i_1}, i_1 = 1, \dots, s_1$, eine Teilmenge L_2 von R_D , innerhalb der es keine Dominanzen gibt und die auf die gleiche Weise "besser" ist als L_1 und L_0 , wie es L_1 im Vergleich zu L_0 ist. Jeder Punkt $y \in L_1$, für den die zuständige Metropolis-Suche keinerlei dominierende Alternativen ergab, zählt als entdecktes Paretooptimum (zumindest konnte der Verdacht auf Effizienz von den Ergebnissen des Metropolis Algorithmus nicht widerlegt werden).

L_2 besteht nun wieder aus Referenzalternativen $x_2^{i_2}, i_2 = 1, \dots, s_2, s_2 \in \mathbb{N}$, für die weitere Metropolis-Prozeduren $MA_2^{i_2}$ (NMA), $i_2 = 1, \dots, s_2$, eine Alternativenliste $L_3 \subseteq R_D$ erarbeiten, welche im bereits beschriebenen Sinn "besser" als L_2, L_1 und L_0 ist. Eventuell können auch einige Elemente von L_2 nach den Suchprozeduren als effiziente Alternativen eingestuft werden. Diese Produktion von zunehmend "besseren" Teilmengen $L_k, k \in \mathbb{N}_0$, von R_D kann man fortsetzen, bis entweder die Metropolis-Suchprozeduren für sämtliche Alternativen in einer Liste $L_s, s \in \mathbb{N}_0$, keine dominierenden Punkte finden können, also eine Liste voll von (anscheinenden) Paretooptima erzeugt wurde oder bereits genügend effiziente Alternativen eingesammelt wurden oder ein maximal erlaubter Gesamtrechenaufwand überschritten ist. Suchstrategie PS ist eine Art Parallelausführung der Suchstrategie SS (Abschnitt 3.2.2). Dabei endet jede Folge von diskreten Suchen (NMA) nach dominierenden Punkten für Referenzpunkte, die eine Folge dominierender Alternativen hervorbringt, beim ersten paretooptimalen Folgenglied.

Abbildung 3.4: Suchstrategie PS erzeugt von Stufe zu Stufe "bessere" Teilmengen von R_D . $x_1^{s_1-1}$ und $x_2^{s_2-1}$ sind hier z.B. aufgesammlte Paretooptima.



Die Sammlerqualitäten des Metropolis Algorithmus (IMA und NMA) werden ausgenutzt, da jede Metropolis-Prozedur (NMA) nicht nur eine, sondern gleich mehrere die Referenzalternative dominierende Alternativen (bestenfalls *numcont* an der Zahl) zu erarbeiteten Zeit bekommt.

3.2.7 Reduzierung des Suchraumes

In Suchstrategie PS hat jede Metropolis-Prozedur $MA_k^{i_k}$ (NMA), $k \in \mathbb{N}_0$, $i_k \in \{1, 2, \dots, s_k\}$, $s_k \in \mathbb{N}$, einen Suchraum $S_k^{i_k} \subseteq \{x_k^{i_k}\} \times (R_D \setminus \{x_k^{i_k}\})$, $x_k^{i_k} \in L_k$, zu erforschen. Von $x_0^{i_0} \in L_0$ zu einer effizienten Alternative $x_l^{i_l} \in L_l$, $l \in \mathbb{N}_0$, gelangen wir mit Hilfe einer Folge von Metropolis-Suchprozeduren $MA_0^{i_0}, MA_1^{i_1}, \dots, MA_{l-1}^{i_{l-1}}$ (NMA). Beim Weg von $x_0^{i_0}$ zu $x_l^{i_l}$ über einen Pfad dominierender Punkte muß eine Gesamtzahl von $\sum_{k=0}^{l-1} |S_k^{i_k}|$ Alternativenpaaren aus $R_D \times R_D$ von der Folge $(MA_0^{i_0}, MA_1^{i_1}, \dots, MA_{l-1}^{i_{l-1}})$ bewältigt werden. Unter Ausnutzung der vorteilhaften Struktur der Menge R_D kann man jedoch bewerkstelligen, daß $\sum_{k=0}^{l-1} |S_k^{i_k}|$ um einiges kleiner ist als $l|R_D|$. Durch geschicktes Anwenden des Wissens um dominierte Punkte kann jeder Suchraum $S_k^{i_k}$ einer Prozedur $MA_k^{i_k}$, $i_k \in \{1, 2, \dots, s_k\}$, $k = 0, \dots, l-1$, mit wachsendem k erheblich reduziert werden.

Wir betrachten den Fall, daß eine Alternative $(y_C, y_D) \in R_D$ eine andere Alternative $(x_C, x_D) \in R_D$ dominiert. Dabei erinnern wir uns an den Aufbau von $R_D = \cup_{x_D \in D} E^{x_D}$, daß diese Menge nämlich die Vereinigung der Mengen kontinuierlich-effizienter Punkte aller möglichen diskreten Variationen $x_D \in D$ ist.

Daß $x_D = y_D$ ist, können wir sofort ausschließen, da Alternativen innerhalb der Menge E^{yD} sich gegenseitig nicht dominieren, weil sie alle paretooptimal sind in $C \times \{x_D\}$. Auf der Suche nach einem Punkt $(z_C, z_D) \in R_D$, der (y_C, y_D) dominiert, braucht man die Alternativen in $E^{xD} \cup E^{yD}$ erst gar nicht beachten. Der Ausschluß von E^{yD} ist mit den eben gemachten Ausführungen begründet, der Ausschluß von E^{xD} folgt aus der Transitivität der Dominanz: $(z_C, z_D) \prec (y_C, y_D)$ dominiert auch (x_C, x_D) . Eine Alternative aus E^{xD} kann keine Alternative aus E^{yD} dominieren.

Wir kehren zu der Betrachtung des Weges von $x_0^{i_0}$ zum Paretooptimum $x_l^{i_l}$ über $x_0^{i_0} \succ x_1^{i_1} \succ \dots \succ x_{l-1}^{i_{l-1}} \succ x_l^{i_l}$ zurück, auf dem wir erfahren, daß der momentane Referenzpunkt $x_k^{i_k}, k \in \{1, \dots, l-1\}$, nicht nur $x_0^{i_0}, \dots, x_{k-1}^{i_{k-1}}$ dominiert, sondern auch die Punkte

$$z_1, \dots, z_r, r \in \mathbb{N}, z_j \in E^{(z_j)D}, j = 1, \dots, r. \quad (3.67)$$

Von der Suche nach einem die Referenzalternative $x_k^{i_k}$ dominierenden Punkt $x_{k+1}^{i_{k+1}}$ kann man somit die gesamte Menge $\cup_{j=1}^r E^{(z_j)D}$ ausnehmen. Der Suchraum der zuständigen Metropolis-Prozedur ist

$$S_k^{i_k} := \{x_k^{i_k}\} \times (R_D \setminus ((\cup_{j=0}^k E^{(x_j^{i_j})D}) \cup (\cup_{j=1}^r E^{(z_j)D}))) =: \{x_k^{i_k}\} \times R_k^{i_k}.$$

Ein gefundenes $x_{k+1}^{i_{k+1}}$ mit $x_{k+1}^{i_{k+1}} \prec x_k^{i_k}$ dominiert natürlich auch die Punkte z_1, \dots, z_r und $x_0^{i_0}, \dots, x_{k-1}^{i_{k-1}}$. Darüber hinaus erhalten wir während der Arbeit von $MA_k^{i_k}$ (NMA), nämlich, wenn ein Ergebnis dieser Prozedur in die Liste L_{k+1} aufgenommen werden soll, eventuell Kenntnis von weiteren Punkten $z_{r+1}, \dots, z_p, p \in \mathbb{N}, p > r$, die von $x_{k+1}^{i_{k+1}}$ dominiert werden. Mit wachsendem $k \in \{1, \dots, l-1\}$ sinkt $|S_k^{i_k}|$:

$$S_0^{i_0} := \{x_0^{i_0}\} \times (R_D \setminus E^{(x_0^{i_0})D}) =: \{x_0^{i_0}\} \times R_0^{i_0}, \quad (3.68)$$

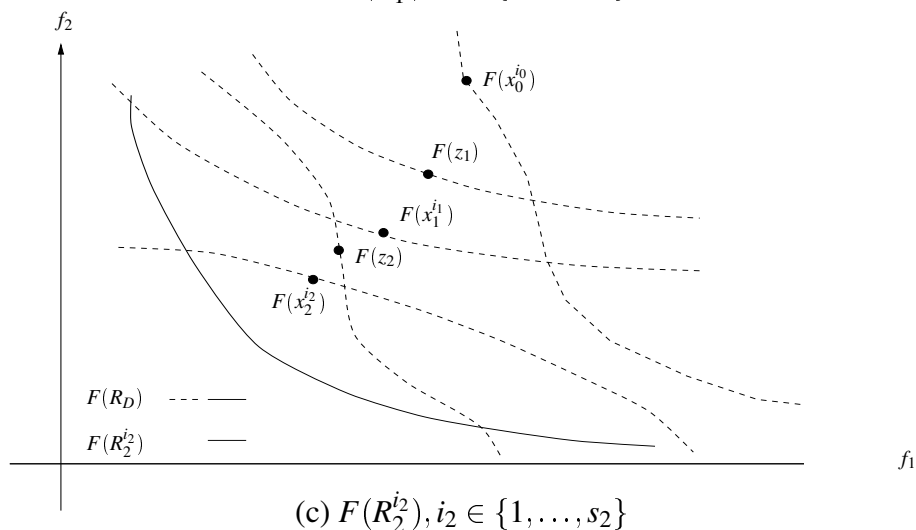
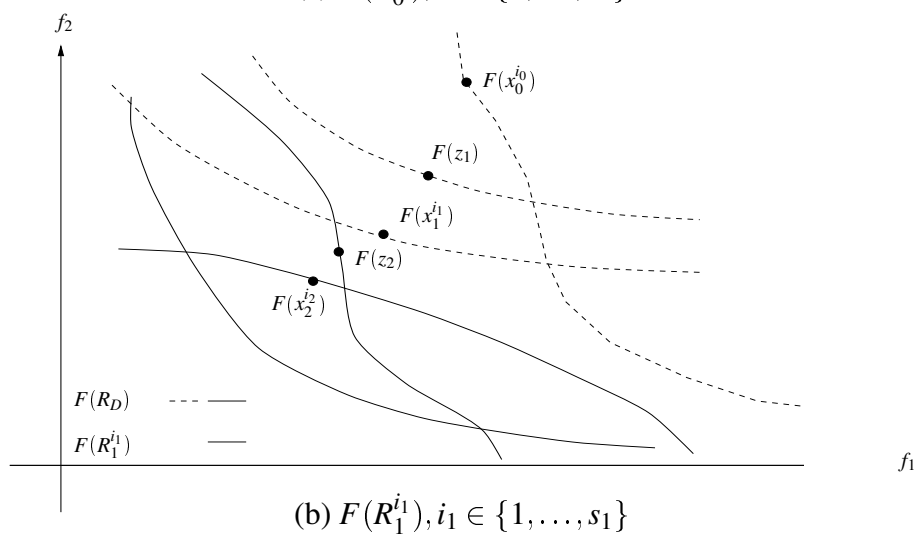
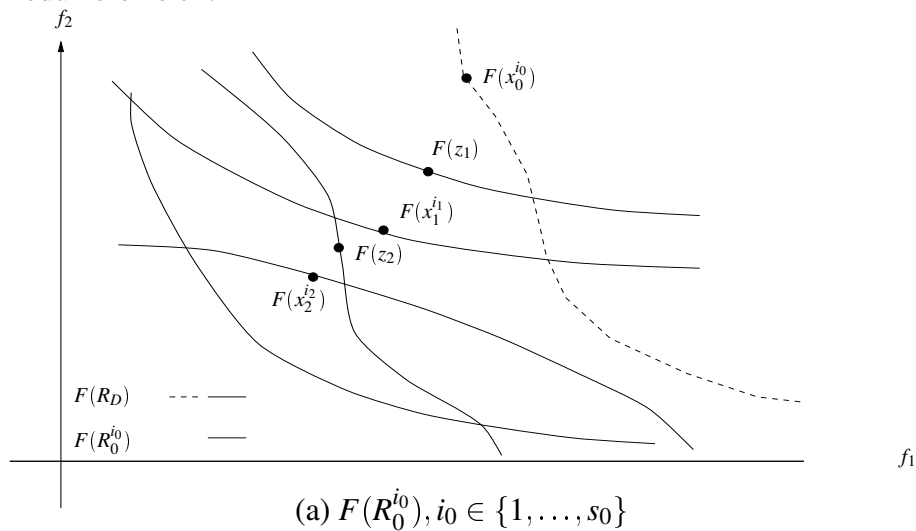
$$S_k^{i_k} := \{x_k^{i_k}\} \times (R_D \setminus ((\cup_{j=0}^k E^{(x_j^{i_j})D}) \cup (\cup_{j=1}^p E^{(z_j)D}))) =: \{x_k^{i_k}\} \times R_k^{i_k}, \quad (3.69)$$

$$R_k^{i_k} \subset R_{k-1}^{i_{k-1}}. \quad (3.70)$$

Auf dem Weg von $x_0^{i_0}$ zu $x_l^{i_l}$ versucht jede Alternative $x_k^{i_k}, k = 0, \dots, l-1$, in ihrer Eigenschaft als Referenzalternative, mehrere sie dominierende Alternativen $y \in S_k^{i_k}$ in L_{k+1} einzutragen. Mit jedem $z \in L_{k+1}$, das durch y aus L_{k+1} "hinausgeworfen wird", weil $y \prec z$ gilt, kann die Menge kontinuierlich-effizienter Punkte E^{zD} vom Suchraum $S_{k+1}^{i_{k+1}}$ (y heiße später $x_{k+1}^{i_{k+1}}$) einer Metropolis-Prozedur der Stufe $k+1$ ausgeschlossen werden.

Abbildung 3.5 bietet eine graphische Veranschaulichung der Reduzierung von Suchräumen.

Abbildung 3.5: Während des Suchlaufes $MA_0^{i_0}$ wird bekannt, daß $x_1^{i_1} \in R_D$ neben $x_0^{i_0} \in R_D$ auch $z_1 \in R_D$ dominiert und während des Suchlaufes $MA_1^{i_1}$, daß $x_2^{i_2} \in R_D$ neben $x_0^{i_0}, x_1^{i_1}, z_1$ auch $z_2 \in R_D$ dominiert. Die gestrichelten Mengen sind die Bildmengen der Alternativenmengen, die jeweils von der Suche nach einem dominierenden Punkt für den jeweiligen Referenzpunkt ausgeschlossen werden können. Sie nehmen von Bild (a) bis Bild (c) zu, die Suchräume der Metropolis-Prozeduren reduzieren sich.



3.2.8 Wahrscheinlichkeitsaussagen über das Erreichen von Paretooptima

3.2.8.1 Der einzelne Markov Prozeß

Die möglichen Zustände der Markov Kette $\{X_t\}_{t \in \mathbb{N}_0}$, die hinter $MA_k^{i_k}$ (NMA), $k \in \mathbb{N}_0, i_k \in \{1, \dots, s_k\}, s_k \in \mathbb{N}$, steht, sind alle Punktepaare $(x_k^{i_k}, y) \in S_k^{i_k} := \{x_k^{i_k}\} \times R_k^{i_k}$, nämlich der Suchraum für $MA_k^{i_k}$. Von $x_k^{i_k} \in R_D$ wissen wir, daß es die Punkte $x_0^{i_0}, \dots, x_{k-1}^{i_{k-1}}, z_1, \dots, z_r \in R_D, r \in \mathbb{N}$, dominiert. Wie im letzten Abschnitt erklärt, ist es nicht notwendig, in $(\cup_{l=0}^k E^{(x_l^{i_l})D}) \cup (\cup_{l=1}^r E^{(z_l)D})$ nach Punkten, die $x_k^{i_k}$ dominieren, zu suchen. Folglich gilt $R_k^{i_k} := R_D \setminus ((\cup_{l=0}^k E^{(x_l^{i_l})D}) \cup (\cup_{l=1}^r E^{(z_l)D}))$. $MA_k^{i_k}$ sei so konstruiert, daß die Markov Kette $\{X_t\}_{t \in \mathbb{N}_0}$, die für festen Kontrollparameter $c \in \mathbb{R}^+$ den IMA modelliert, von dem $MA_k^{i_k}$ eine numerische Version ist, die Anforderungen von Satz 3.2.22 erfüllt. Des Weiteren wird Notation 3.2.23 jetzt leicht verändert:

Notation 3.2.24. Betrachtet man einen Punkt $x_k^{i_k} \in L_k$, so erlaubt uns das Prinzip der Dominanz, die Menge $R_k^{i_k}$ in drei Teilmengen zu zerlegen:

- $Dom_k^{i_k} \subseteq R_k^{i_k}$ repräsentiert alle $y \in R_k^{i_k}$ mit $y \prec x_k^{i_k}$.
- $Ind_k^{i_k} \subseteq R_k^{i_k}$ enthält alle $y \in R_k^{i_k}$ mit $y \asymp x_k^{i_k}$.
- $NDom_k^{i_k} = R_k^{i_k} \setminus (Dom_k^{i_k} \cup Ind_k^{i_k})$ ist folglich die Menge aller $y \in R_k^{i_k}$ mit $y \succ x_k^{i_k}$.

△

Unter Verwendung der eben eingeführten Mengen erscheinen in der stationären Verteilung der Markov Kette $\{X_t\}_{t \in \mathbb{N}_0}$, die in Experiment $X_{\bar{t}}, \bar{t} \in \mathbb{N}$, schon fast gelte, wobei $\{X_t\}_{0 \leq t \leq \bar{t}}$ für ein $c \in \mathbb{R}^+$ die Handlungen von $MA_k^{i_k}$ beschreibe, folgende drei Werte:

$$q_{(x_k^{i_k}, y)}(c) = \begin{cases} \frac{\exp(\frac{-min}{c})}{|Dom_k^{i_k}| \exp(\frac{-min}{c}) + |Ind_k^{i_k}| \exp(\frac{-mid}{c}) + |NDom_k^{i_k}| \exp(\frac{-max}{c})}, & \text{falls } y \prec x_k^{i_k}, \\ \frac{\exp(\frac{-mid}{c})}{|Dom_k^{i_k}| \exp(\frac{-min}{c}) + |Ind_k^{i_k}| \exp(\frac{-mid}{c}) + |NDom_k^{i_k}| \exp(\frac{-max}{c})}, & \text{falls } y \asymp x_k^{i_k}, \\ \frac{\exp(\frac{-max}{c})}{|Dom_k^{i_k}| \exp(\frac{-min}{c}) + |Ind_k^{i_k}| \exp(\frac{-mid}{c}) + |NDom_k^{i_k}| \exp(\frac{-max}{c})}, & \text{falls } y \succ x_k^{i_k} \end{cases} \quad (3.71)$$

$\forall (x_k^{i_k}, y) \in S_k^{i_k}$.

Satz 3.2.25. Für jedes $k \in \mathbb{N}_0, i_k \in \{1, \dots, s_k\}, s_k \in \mathbb{N}$, erfülle die Markov Kette $\{X_t\}_{t \in \mathbb{N}_0}$, die für ein $c \in \mathbb{R}^+$ den IMA modelliert, von dem $MA_k^{i_k}$ eine numerische Version ist, die Anforderungen von Satz 3.2.22, und die stationäre Verteilung sei durch (3.71) gegeben. $\bar{t} \in \mathbb{N}$ sei groß genug, daß man in etwa die Gültigkeit der Gleichung $P\{X_{\bar{t}} = (x_k^{i_k}, y)\} = q_{(x_k^{i_k}, y)}(c)$ für alle $(x_k^{i_k}, y) \in S_k^{i_k}$

annehmen kann. Damit ergibt sich

$$\begin{aligned}
P\{X_{\bar{t}} = (x_k^{i_k}, y)\} &= q_{(x_k^{i_k}, y)}(c) \quad \text{mit } y \prec x_k^{i_k} \\
&> \\
P\{X_{\bar{t}} = (x_k^{i_k}, y)\} &= q_{(x_k^{i_k}, y)}(c) \quad \text{mit } y \asymp x_k^{i_k} \\
&> \\
P\{X_{\bar{t}} = (x_k^{i_k}, y)\} &= q_{(x_k^{i_k}, y)}(c) \quad \text{mit } y \succ x_k^{i_k} \\
&\forall (x_k^{i_k}, y) \in S_k^{i_k}.
\end{aligned} \tag{3.72}$$

Korollar 3.2.26. Die Markov Kette $\{X_t\}_{t \in \mathbb{N}_0}$, die für ein $c \in \mathbb{R}^+$ den IMA modelliert, von dem $MA_k^{i_k}$ eine numerische Version ist, werde in Iteration $\bar{t} \in \mathbb{N}$ beobachtet. \bar{t} sei groß genug für die Annahme, daß jede Zeile von $P_{trans}^{\bar{t}}(c)$ eine gute Approximation von $q(c)$ liefert. Dann können wir feststellen:

$$\begin{aligned}
P\{X_{\bar{t}} \in Dom_k^{i_k}\} &= \sum_{y \in Dom_k^{i_k}} q_{(x_k^{i_k}, y)}(c) = \\
&= \frac{1}{1 + \frac{|Ind_k^{i_k}|}{|Dom_k^{i_k}|} \exp\left(\frac{min-mid}{c}\right) + \frac{|NDom_k^{i_k}|}{|Dom_k^{i_k}|} \exp\left(\frac{min-max}{c}\right)}.
\end{aligned} \tag{3.73}$$

Mit $\frac{min-mid}{c} \ll 0$ und $\frac{min-max}{c} \ll 0$ kann man die Wahrscheinlichkeit (3.73), nach genügend vielen Iterationen ein $y \in R_k^{i_k}$ zu erhalten, das $x_k^{i_k}$ dominiert, fast auf den Wert 1 bringen. In diesem Fall hat der Algorithmus $MA_k^{i_k}$ jedoch erhebliche Schwierigkeiten, Verschlechterungen im Zielfunktionswert zu akzeptieren. Er könnte mehrere Iterationen lang in einem lokalen Minimum hängen bleiben, so daß mehr Iterationen nötig sind, bis die stationäre Verteilung annähernd gilt, als mit Werten für $\frac{min-mid}{c}$ und $\frac{min-max}{c}$, die näher bei null liegen.

3.2.8.2 Der stochastische Superprozeß

Suchstrategie SS, Version SS1:

In Suchstrategie SS laufen mehrere Metropolis-Suchprozeduren (NMA) in Folge. Für ein zufällig gewähltes $x_0 \in R_D$ sucht ein Metropolis Algorithmus MA_0 (NMA) nach einem $y_0 \in R_D$, das x_0 dominiert. Dabei muß er den Suchraum $S_0 := \{x_0\} \times R_0$, $R_0 := R_D \setminus E^{(x_0)D}$ durchsuchen. MA_0 arbeite lange genug ($\bar{t} \in \mathbb{N}$ Iterationen), damit die stationäre Verteilung $q(c)$ der Markov Kette $\{X_t\}_{t \in \mathbb{N}_0}$, die für ein $c \in \mathbb{R}^+$ den IMA, von dem MA_0 eine numerische Version ist, modelliert, durch die Zeilen der Matrix $P_{trans}^{\bar{t}_0}(c)$ gut approximiert wird. Dann ergibt sich der willkommene Effekt, daß jedes einzelne Paar $(x_0, \bar{y}) \in S_0$ mit $\bar{y} \prec x_0$ im Zufallsexperiment $X_{\bar{t}_0}$ von MA_0 mit größerer Wahrscheinlichkeit entdeckt wird als jedes einzelne Paar $(x_0, \hat{y}) \in S_0$ mit $\hat{y} \succ x_0$ oder $\hat{y} \asymp x_0$. Das Ergebnispaar von Iteration \bar{t}_0 , (x_0, y_0) , repräsentiert den Übergang von Referenzalternative $x_0 \in R_D$ zu Alternative $y_0 \in R_D$. Dieser Übergang kann, da er vom stochastischen Metropolis Algorithmus zustande gebracht wird, als Ergebnis eines Zufallsexperimentes aufgefaßt werden, das von der diskreten Zufallsvariable Y_0 mit Realisierungen in R_0 dargestellt werde. y_0 wird die Referenzalternative x_1 der nun folgenden Metropolis-Suchprozedur MA_1

(NMA).

Eine Folge von Metropolis-Prozeduren $MA_0, MA_1, \dots, MA_{k-1}, k \in \mathbb{N}$, (alle NMA) mit Start-Referenzalternative $x_0 \in R_D$ habe die Folge x_1, \dots, x_k von Alternativen $x_l \in R_{l-1} \subseteq R_D, l = 1, \dots, k$, produziert oder anders ausgedrückt einen „Weg“ $x_0 \mapsto x_1 \mapsto \dots \mapsto x_k$ durch R_D . Daraufhin suche MA_k (NMA) nach einem y_k aus R_k , das x_k dominiert. Es sei k_1 die größte Zahl in $\{1, \dots, k\}$, bei der $x_{k_1-1} \prec x_{k_1}$ oder $x_{k_1-1} \asymp x_{k_1}$ gelte, dann ergibt sich $R_k := R_D \setminus (\cup_{i=k_1}^k E^{(x_i)D})$. Die Größe von \bar{t}_k mag in Iterationen wieder die näherungsweise Gültigkeit der stationären Verteilung $q(c)$ der Markov Kette $\{X_t\}_{t \in \mathbb{N}_0}$ erlauben, die für ein $c \in \mathbb{R}^+$ den IMA modelliert, von dem MA_k eine numerische Version ist. Das Ergebnispaar $(x_k, y_k) \in S_k := \{x_k\} \times R_k$ des Experiments $X_{\bar{t}_k}$ von MA_k steht wieder für den Übergang von x_k zu x_{k+1} . Die diskrete Zufallsvariable Y_k mit Ergebnissen in R_k symbolisiere diesen Übergang.

$\{y_{l-1}\}_{l=1, \dots, k+1} = \{x_l\}_{l=1, \dots, k+1}$ ist eine Realisierung des **stochastischen Superprozesses** $\{Y_{l-1}\}_{l=1, \dots, k+1}$. Der Ausdruck Superprozeß weist auf die Tatsache hin, daß jedes einzelne seiner Zufallsexperimente $Y_l, l \in \{0, \dots, k\}$, in sich wieder ein stochastischer Prozeß ist, nämlich eine Markov Kette $\{X_t\}_{0 \leq t \leq \bar{t}_l}, \bar{t}_l \in \mathbb{N}$, die für ein $c \in \mathbb{R}^+$ die Metropolis-Prozedur MA_l (NMA) modelliert.

Die Menge der möglichen Realisierungen aus R_l von $Y_l, l \in \mathbb{N}_0$, hängt für alle $l \in \mathbb{N}$ vom Ausgang der Experimentenfolge $\{Y_j\}_{j=0, \dots, l-1}$ ab, da R_l aufgrund von Informationen, die in den vorhergehenden Experimenten gesammelt werden, reduziert wird. Folglich ist

$$\begin{aligned} & P\{Y_l = y_l | Y_{l-1} = y_{l-1}, Y_{l-2} = \bar{y}_{l-2}, \dots, Y_0 = \bar{y}_0\} \\ & \neq P\{Y_l = y_l | Y_{l-1} = y_{l-1}, Y_{l-2} = \hat{y}_{l-2}, \dots, Y_0 = \hat{y}_0\} \end{aligned} \quad (3.74)$$

mit $\bar{y}_j \neq \hat{y}_j$ für ein $j \in \{0, \dots, l-2\}$

möglich. Offensichtlich ist der stochastische Superprozeß keine Markov Kette.

Für weitere Aussagen machen wir uns eine Umformulierung von Notation 3.2.23 zunutze:

Notation 3.2.27. Betrachtet man eine Alternative $x_k \in R_D$, die das Ergebnis des momentan letzten Experimentes $Y_{k-1}, k \in \mathbb{N}$, des stochastischen Superprozesses $\{Y_l\}_{l=0, \dots, k-1}$ ist, so erlaubt uns das Prinzip der Dominanz, die Menge R_k in drei Teilmengen zu zerlegen:

- $Dom_k \subset R_k$ repräsentiert alle $y \in R_k$ mit $y \prec x_k$.
- $Ind_k \subseteq R_k$ enthält alle $y \in R_k$ mit $y \asymp x_k$.
- $NDom_k = R_k \setminus (Dom_k \cup Ind_k)$ ist folglich die Menge aller Punkte $y \in R_k$ mit $y \succ x_k$.

△

Satz 3.2.28. Jeder direkte Weg durch R_D von einer nicht-effizienten Start-Referenzalternative $x_0 \in R_D$ zu einem Paretooptimum $\bar{x} \in Dom_0$ ist wahrscheinlicher als jeder Weg $x_0 \mapsto x$ mit $x \in R_0$ und $x \not\prec x_0$, der x_0 fälschlicherweise als Paretooptimum einstuft,

und wahrscheinlicher als jeder Weg $x_0 \mapsto \hat{x}, \hat{x} \in Dom_0$, mit irrtümlicher Einstufung von \hat{x} als Paretooptimum durch Y_1 ,

sowie wahrscheinlicher als jeder Umweg $x_0 \mapsto x_1 \mapsto \dots \mapsto x_k = \hat{x}, k > 1$, durch R_D über nichteffiziente Alternativen x_1, \dots, x_{k-1} zu einer ersten Alternative $\hat{x} \in R_D$, die von Y_k als paretooptimal eingestuft wird.

Beweis: Bei dieser Betrachtung interessieren die Ausgänge des Zufallsprozesse $\{Y_l\}_{l \in \mathbb{N}_0}$ nur bis zum ersten Experiment $Y_k, k \in \mathbb{N}_0$, das einen Punkt x_k als paretooptimal einstuft.

$$\begin{aligned}
P\{x_0 \mapsto \bar{x}\} &= \\
P\{Y_0 = \bar{x}\} &= \\
&\frac{\exp\left(\frac{-\min}{c}\right)}{|Dom_0| \exp\left(\frac{-\min}{c}\right) + |Ind_0| \exp\left(\frac{-\text{mid}}{c}\right) + |NDom_0| \exp\left(\frac{-\max}{c}\right)} \\
&> \\
&\frac{\exp\left(\frac{-Z[(x_0, \bar{x})]}{c}\right)}{|Dom_0| \exp\left(\frac{-\min}{c}\right) + |Ind_0| \exp\left(\frac{-\text{mid}}{c}\right) + |NDom_0| \exp\left(\frac{-\max}{c}\right)} \\
&= P\{Y_0 = x\} = \\
P\{x_0 \mapsto x\} &
\end{aligned} \tag{3.75}$$

für alle $x \in R_D$ mit $x \not\prec x_0$.

$$\begin{aligned}
P\{x_0 \mapsto \bar{x}\} &= \\
P\{Y_0 = \bar{x}\} &= \\
&\frac{\exp\left(\frac{-\min}{c}\right)}{|Dom_0| \exp\left(\frac{-\min}{c}\right) + |Ind_0| \exp\left(\frac{-\text{mid}}{c}\right) + |NDom_0| \exp\left(\frac{-\max}{c}\right)} \\
&> \\
&\prod_{l=0}^{k-1} \frac{\exp\left(\frac{-Z[(x_l, x_{l+1})]}{c}\right)}{|Dom_l| \exp\left(\frac{-\min}{c}\right) + |Ind_l| \exp\left(\frac{-\text{mid}}{c}\right) + |NDom_l| \exp\left(\frac{-\max}{c}\right)} * P = \\
P\{Y_{k-1} = \hat{x} | Y_{k-2} = x_{k-1}, \dots, Y_0 = x_1\} * P &= \\
P\{x_0 \mapsto x_1 \mapsto \dots \mapsto x_k = \hat{x}\} * P, P \in \mathbb{R}, &
\end{aligned} \tag{3.76}$$

mit $P < 1$, falls \hat{x} von Y_0 ($k = 0$) als Ergebnis produziert und von Y_1 irrtümlicherweise als Paretooptimum eingestuft wird, $P \leq 1$ im Falle $k > 1$, je nachdem, ob \hat{x} nun wirklich paretooptimal ($P = 1$) ist oder nicht ($P < 1$). P drückt die Wahrscheinlichkeit aus, mit der Y_k kein $x \in R_k$ findet, das \hat{x} dominiert. ■

Bemerkung 3.2.29. Satz 3.75 sagt nichts darüber aus, wie sicher es ist, durch Hintereinanderführung von Zufallsexperimenten $Y_l, l \in \mathbb{N}_0$, überhaupt ein Paretooptimum zu finden. Im Falle $x_{k+1} \in (Ind_k \cup NDom_k)$ für ein $k \in \mathbb{N}_0$ mit $Dom_k \neq \emptyset$ hat das Experiment Y_k die Alternative x_k als Paretooptimum eingestuft, da es keinen Punkt hervorbrachte, der x_k dominiert, und somit nicht die Nichteffizienz von x_k beweisen konnte. Dem Experiment \bar{t}_k der Markov Kette $\{X_t\}_{t \in \mathbb{N}_0}$, die für ein $c \in \mathbb{R}^+$ den IMA modelliert, von dem MA_k eine numerische Version ist, liege in etwa die stationäre Verteilung $q(c)$ zugrunde. $\{X_t\}_{0 \leq t \leq \bar{t}_k}$ beschreibt den Suchlauf von MA_k , und der Ausgang des Experimentes $X_{\bar{t}_k}$ liefert den Ausgang des Experimentes Y_k . Die Wahrscheinlichkeit, daß Y_k bei der Einstufung von x_k als effizient einen Fehler begangen hat, ist

$$\begin{aligned}
P_{err}^k &:= 1 - \frac{|Dom_k| \exp\left(\frac{-min}{c}\right)}{|Dom_k| \exp\left(\frac{-min}{c}\right) + |Ind_k| \exp\left(\frac{-mid}{c}\right) + |NDom_k| \exp\left(\frac{-max}{c}\right)} \\
&= 1 - \frac{1}{1 + \frac{|Ind_k|}{|Dom_k|} \exp\left(\frac{min-mid}{c}\right) + \frac{|NDom_k|}{|Dom_k|} \exp\left(\frac{min-max}{c}\right)}.
\end{aligned} \tag{3.77}$$

△

Satz 3.2.30. *Sobald ein echtes Paretooptimum $\bar{x} \in R_D$ auf einem bestimmten Weg durch R_D in Experiment $Y_{l_1}, l_1 \in \mathbb{N}_0$, erreicht wurde und Ind_{l_1+1} keine leere Menge ist und Paretooptima enthält, wird ein weiteres Paretooptimum $\tilde{x} \in Ind_{l_1+1}$ mit größerer Wahrscheinlichkeit unmittelbar danach gefunden, d.h. $Y_{l_1} = \bar{x}$ und $Y_{l_1+1} = \tilde{x}, l_1 \in \mathbb{N}_0$,*

als daß Y_{l_1+1} ein nichteffizientes $\hat{x} \in R_D$ hervorbringt und Y_{l_1+2} keine \hat{x} dominierende Alternative findet

und als daß eine unter den nach \bar{x} angelaufenen Alternativen als nächste für ein Paretooptimum gehaltene Alternative \hat{x} erst auf einem "Umweg" über mehrere nichteffiziente Alternativen entdeckt wird, d.h. $x_{l_1+1} = \bar{x} \mapsto x_{l_1+2} \mapsto \dots \mapsto x_{l_2} = \hat{x}, l_2 > l_1 + 2$.

Beweis: Das erste (echte) Paretooptimum $\bar{x} \in R_D$ sei auf dem Weg $x_0 \mapsto x_1 \mapsto \dots \mapsto x_{l_1} \mapsto x_{l_1+1} = \bar{x}$ erreicht worden. $Ind_{l_1+1} \subseteq Ind_{\bar{x}} \setminus (E^{\bar{x}D} \setminus \{\bar{x}\})$, da die Paretooptimalität von \bar{x} erst durch Experiment Y_{l_1+1} erkannt wird und man daher Y_{l_1+1} nach dominierenden Punkten für \bar{x} suchen läßt; aus dieser Suche wird unter anderem $E^{\bar{x}D}$ ausgeklammert.

$\tilde{x} \in Ind_{l_1+1}$ sei eine weitere nichtdominierte Alternative und $\hat{x} \in R_D$ ein Punkt, der als effizient eingestuft wird. Betrachten wir den direkten Weg $x_{l_1+1} = \bar{x} \mapsto x_{l_1+2} = \tilde{x}$ und die Wege $x_{l_1+1} = \bar{x} \mapsto x_{l_1+2} \mapsto \dots \mapsto x_{l_2} = \hat{x}, l_1 + 2 \leq l_2 \in \mathbb{N}$:

$$\begin{aligned}
&P\{x_0 \mapsto x_1 \mapsto \dots \mapsto x_{l_1+1} = \bar{x} \mapsto x_{l_1+2} = \tilde{x}\} \\
&= P\{Y_{l_1+1} = \tilde{x} | Y_{l_1} = \bar{x}, \dots, Y_0 = x_1\} \\
&= P\{Y_{l_1} = \bar{x}, \dots, Y_0 = x_1\} \\
&* \frac{\exp\left(\frac{-mid}{c}\right)}{|Ind_{l_1+1}| \exp\left(\frac{-mid}{c}\right) + |NDom_{l_1+1}| \exp\left(\frac{-max}{c}\right)} \\
&> \\
&P\{Y_{l_1} = \bar{x}, \dots, Y_0 = x_1\} \\
&* \prod_{l=l_1+1}^{l_2-1} \frac{\exp\left(\frac{-Z[(x_l, x_{l+1})]}{c}\right)}{|Dom_l| \exp\left(\frac{-min}{c}\right) + |Ind_l| \exp\left(\frac{-mid}{c}\right) + |NDom_l| \exp\left(\frac{-max}{c}\right)} * P \\
&= P\{Y_{l_2-1} = \hat{x} | Y_{l_2-2} = x_{l_2-1}, \dots, Y_{l_1} = \bar{x}, Y_{l_1-1} = x_{l_1}, \dots, Y_0 = x_1\} * P \\
&= P\{x_0 \mapsto x_1 \mapsto \dots \mapsto x_{l_1+1} = \bar{x} \mapsto x_{l_1+2} \mapsto \dots \mapsto x_{l_2} = \hat{x}\} * P, \\
&|Dom_{l_1+1}| = 0.
\end{aligned} \tag{3.78}$$

$Z[(x_{l_1+1}, x)] \geq mid$ für alle $x \in R_{l_1+1}$. $P < 1$ im Falle $l_2 = l_1 + 1$, da es die Wahrscheinlichkeit beschreibt, mit welcher Y_{l_1+1} keine Alternative $x \in Dom_{l_1+2}$ findet, obwohl diese Menge nicht leer ist. Bei $l_2 > l_1 + 2$ ist $P = 1$, falls \hat{x} wirklich effizient ist und $P < 1$, falls es irrtümlicherweise für paretooptimal gehalten wird. ■

Bemerkung 3.2.31. Die Sätze 3.2.28 und 3.2.30 können nur deshalb leicht bewiesen werden, weil die stationären Verteilungen der verschiedenen vorkommenden Markov Ketten die Alternativen aus den jeweiligen Mengen DOM_k bzw. IND_k (falls $DOM_k = \emptyset$), $k \in \mathbb{N}_0$, bevorzugen.

△

Der Leser möge sich daran erinnern, daß das Ergebnis eines Zufallsexperimentes $Y_k, k \in \mathbb{N}_0$, folgendermaßen definiert wurde: Die Markov Kette $\{X_t\}_{t \in \mathbb{N}_0}$, die für festen Kontrollparameter $c \in \mathbb{R}^+$ den IMA modelliert, von dem MA_k eine numerische Version ist, wird in Iteration $\bar{t}_k \in \mathbb{N}$ beobachtet. Diese Iterationszahl ist so groß angesetzt, daß man von der Markov Kette annehmen darf, daß in Experiment $X_{\bar{t}_k}$ die stationäre Verteilung fast gilt. Betrachtet man das Ergebnispaar (x_k, y) von $X_{\bar{t}_k}$, so bildet y das Ergebnis y_k von Y_k . Diese spezielle Definition der einzelnen Experimente $\{Y_k\}_{k \in \mathbb{N}_0}$ des Superprozesses ermöglicht es, die Wahrscheinlichkeiten für die Realisierungen des Superprozesses in einfacher Weise zu errechnen. Es sind jedoch noch zwei leicht modifizierte Konstruktionen für jedes Zufallsexperiment $Y_k, k \in \mathbb{N}$, vorstellbar, die in numerischer Hinsicht praktikabler sind.

SS2: Modifikation 1 der Definition von $Y_k, k \in \mathbb{N}_0$, in Suchstrategie SS1 Diese Alternative für die Definition von $Y_k, k \in \mathbb{N}_0$, will der Markov Kette $\{X_t\}_{t \in \mathbb{N}_0}$, die hinter Y_k steckt, genügend Iterationen $\underline{t}_k \in \mathbb{N}$ erlauben, damit sie von der dann fast gültigen stationären Verteilung profitieren kann, sie aber danach noch weiter, aber nicht mehr als notwendig (höchstens $\bar{t}_k \geq \underline{t}_k$ Iterationen lang) arbeiten lassen. Die Weiterarbeit nach \underline{t}_k soll Ergebnisse $y_k \prec x_k$ und $y_k \succ x_k$ mehr fördern und Ausgänge $y_k \succ x_k$ mehr erschweren als die Definition von Y_k in SS1. Die Suchprozedur MA_k (NMA), modelliert durch $\{X_t\}_{0 \leq t \leq \bar{t}_k}$, wird von Experiment $X_{\underline{t}_k}$ an so lange beobachtet, bis das erste Alternativenpaar $(x_k, y) \in S_k$ mit $y \prec x_k$ erreicht oder das letzte (zeitlich) zugelassene Experiment $X_{\bar{t}_k}$ vollführt ist. Wurde ein gewünschtes Paar (x_k, y) mit $y \prec x_k$ vor Iteration \bar{t}_k entdeckt, wird sein zweiter Partner y Ergebnis y_k des Experiments Y_k , ansonsten wird der zweite Partner des ersten gefundenen Paares aus S_k ohne vorzuziehendem Vorgänger- oder Nachfolgepaar das Ergebnis y_k von Y_k . Um den Wert der Wahrscheinlichkeit für $Y_k = \bar{y}$ bei einem bestimmten Paar $(x_k, \bar{y}) \in S_k$ in handliche Ausdrucksform zu bringen, bedienen wir uns folgender Mengen und Bezeichnungen:

- $OUT_{(x_k, \bar{y})} \subseteq R_k$ enthalte alle Alternativen $y \in R_k$, bei denen das entsprechende Alternativenpaar $(x_k, y) \in S_k$ als möglicher Lieferant des Ergebnisses von Y_k durch (x_k, \bar{y}) ersetzt wird, falls (x_k, \bar{y}) in einem späteren Experiment der Markov Kette erreicht wird als (x_k, y) , d.h. eventuell gilt dann $y_k = \bar{y}$, aber sicher nicht $y_k = y$. Ein später gefundenes Paar (x_k, \bar{y}) wird allen Paaren (x_k, y) mit $y \in OUT_{(x_k, \bar{y})}$ vorgezogen.
- $IN_{(x_k, \bar{y})} \subseteq R_k$ enthält alle Alternativen $y \in R_k$, die zusammen mit x_k ein Paar $(x_k, y) \in S_k$ bilden, das (x_k, \bar{y}) in seiner Rolle als Lieferant des Ergebnisses von Y_k nicht verdrängen kann, falls es in einem späteren Experiment der Markov Kette aufgespürt wird als (x_k, \bar{y}) . Alle später gefundenen (x_k, y) mit $y \in IN_{(x_k, \bar{y})}$ werden (x_k, \bar{y}) "nicht vorgezogen".
- $r_k := \bar{t}_k - \underline{t}_k$.

Die Wahrscheinlichkeit dafür, daß $\bar{y} \in R_k$ das Ergebnis y_k von Y_k wird, ist

$$\begin{aligned}
& P\{Y_k = \bar{y}\} \\
&= q_{(x_k, \bar{y})}(c) \sum_{z_{r_k} \in IN_{(x_k, \bar{y})}} \cdots \sum_{z_1 \in IN_{(x_k, \bar{y})}} P_{(x_k, \bar{y})}(x_k, z_1)(c) P_{(x_k, z_1)}(x_k, z_2)(c) \cdots P_{(x_k, z_{r_k-1})}(x_k, z_{r_k})(c) \\
&+ \sum_{z_1 \in OUT_{(x_k, \bar{y})}} q_{(x_k, z_1)}(c) P_{(x_k, z_1)}(x_k, \bar{y})(c) \\
&\quad * \left(\sum_{z_{r_k} \in IN_{(x_k, \bar{y})}} \cdots \sum_{z_2 \in IN_{(x_k, \bar{y})}} P_{(x_k, \bar{y})}(x_k, z_2)(c) P_{(x_k, z_2)}(x_k, z_3)(c) \cdots P_{(x_k, z_{r_k-1})}(x_k, z_{r_k})(c) \right) \\
&+ \dots
\end{aligned} \tag{3.79}$$

$$\begin{aligned}
&+ \sum_{z_{r_k-1} \in OUT_{(x_k, \bar{y})}} \cdots \sum_{z_2 \in OUT_{(x_k, \bar{y})}} \sum_{z_1 \in OUT_{(x_k, \bar{y})}} q_{(x_k, z_1)}(c) P_{(x_k, z_1)}(x_k, z_2)(c) \cdots P_{(x_k, z_{r_k-1})}(x_k, \bar{y})(c) \\
&\quad * \left(\sum_{z_{r_k} \in IN_{(x_k, \bar{y})}} P_{(x_k, \bar{y})}(x_k, z_{r_k})(c) \right) \\
&+ \sum_{z_{r_k} \in OUT_{(x_k, \bar{y})}} \cdots \sum_{z_2 \in OUT_{(x_k, \bar{y})}} \sum_{z_1 \in OUT_{(x_k, \bar{y})}} q_{(x_k, z_1)}(c) P_{(x_k, z_1)}(x_k, z_2)(c) \cdots P_{(x_k, z_{r_k})}(x_k, \bar{y})(c).
\end{aligned}$$

Drei interessante Fälle sind zu unterscheiden:

1. $\bar{y} \in Dom_k$:

$$\begin{aligned}
OUT_{(x_k, \bar{y})} &= NDom_k \cup Ind_k, \\
IN_{(x_k, \bar{y})} &= R_k,
\end{aligned} \tag{3.80}$$

2. $\bar{y} \in Ind_k$:

$$\begin{aligned}
OUT_{(x_k, \bar{y})} &= NDom_k \cup Ind_k, \\
IN_{(x_k, \bar{y})} &= NDom_k,
\end{aligned} \tag{3.81}$$

3. $\bar{y} \in NDom_k$:

$$\begin{aligned}
OUT_{(x_k, \bar{y})} &= NDom_k, \\
IN_{(x_k, \bar{y})} &= \emptyset.
\end{aligned} \tag{3.82}$$

Offensichtlich gilt

$$P\{Y_k = y_k\} > \frac{\exp\left(\frac{-\min}{c}\right)}{|Dom_k| \exp\left(\frac{-\min}{c}\right) + |Ind_k| \exp\left(\frac{-mid}{c}\right) + |NDom_k| \exp\left(\frac{-max}{c}\right)}, \quad (3.83)$$

falls $y_k \prec x_k$,

$$P\{Y_k = y_k\} < \frac{\exp\left(\frac{-max}{c}\right)}{|Dom_k| \exp\left(\frac{-\min}{c}\right) + |Ind_k| \exp\left(\frac{-mid}{c}\right) + |NDom_k| \exp\left(\frac{-max}{c}\right)}, \quad (3.84)$$

falls $y_k \succ x_k$.

Die in (3.79) ausgedrückte Wahrscheinlichkeit ist die Summe der Wahrscheinlichkeiten, daß (x_k, \bar{y}) in den Experimenten $X_j, \bar{t}_k \geq j \geq \underline{t}_k$ von MA_k (NMA) entdeckt wird und es kein "vorzuziehendes" Ergebnis in einem der vorhergehenden Experimente $X_i, i = \underline{t}_k, \dots, j-1$, ("vorzuziehende" Vorgängerpaare sind alle (x_k, y) mit $y \notin OUT_{(x_k, \bar{y})}$) gab und in keinem der nachfolgenden Experimente $X_i, i = j+1, \dots, \bar{t}_k$, ein "vorzuziehendes" Ergebnis produziert werden wird ("vorzuziehende" Nachfolgerpaare sind alle (x_k, y) mit $y \notin IN_{(x_k, \bar{y})}$). Wegen $OUT_{(x_k, \bar{y})} \subseteq OUT_{(x_k, \hat{y})}$ oder $OUT_{(x_k, \hat{y})} \subseteq OUT_{(x_k, \bar{y})}$ und $IN_{(x_k, \bar{y})} \subseteq IN_{(x_k, \hat{y})}$ oder $IN_{(x_k, \hat{y})} \subseteq IN_{(x_k, \bar{y})} \quad \forall \bar{y}, \hat{y} \in S_k$ steigt mit den Mächtigkeiten von $OUT_{(x_k, \bar{y})}$ und $IN_{(x_k, \bar{y})}$ nicht nur die Anzahl der mögliche Realisierungen $(x_k, z_{\underline{t}_k}), \dots, (x_k, z_{j-1}), (x_k, z_{j+1}), \dots, (x_k, z_{\bar{t}_k})$ von $X_{\underline{t}_k}, \dots, X_{j-1}, X_{j+1}, \dots, X_{\bar{t}_k}$ die (x_k, \bar{y}) nicht davon abhalten, das Ergebnis $y_k = \bar{y}$ von Y_k zu stellen, sondern auch die Wahrscheinlichkeit des Ereignisses „Nichtabhaltung“. Es bleibt jedoch die Frage zu klären, wie wahrscheinlich die Übergänge von Standpunkten $(x_k, z_{j-1}), z_{j-1} \in OUT_{(x_k, \bar{y})}$, zu (x_k, \bar{y}) und von (x_k, \bar{y}) zu den Standpunkten $(x_k, z_{j+1}), z_{j+1} \in IN_{(x_k, \bar{y})}$, sind. Da ein generiertes Paar (x_k, \bar{y}) immer sicher als neuer Standpunktkandidat akzeptiert wird ($Z[(x_k, z_{j-1})] \geq Z[(x_k, \bar{y})] \quad \forall z_{j-1} \in OUT_{(x_k, \bar{y})}$), hängen die Übergangswahrscheinlichkeiten $P_{(x_k, z_{j-1})(x_k, \bar{y})}(c)$ lediglich von den Generierungswahrscheinlichkeiten ab. Bildet z.B. für jedes Alternativenpaar $(x_k, y) \in S_k$ die gesamte Menge $S_k \setminus \{(x_k, y)\}$ die Nachbarschaft und differiert die Generierungswahrscheinlichkeit für jeden Nachbarn daraus nicht zu sehr von $\frac{1}{|S_k|-1}$, können wir folgern:

$$\sum_{z_{j-1} \in OUT_{(x_k, \bar{y})}} \dots \sum_{z_2 \in OUT_{(x_k, \bar{y})}} \sum_{z_1 \in OUT_{(x_k, \bar{y})}} q_{(x_k, z_1)}(c) P_{(x_k, z_1)(x_k, z_2)}(c) \dots P_{(x_k, z_{j-1})(x_k, \bar{y})}(c) \quad (3.85)$$

ist umso höher, je mehr Elemente in $OUT_{(x_k, \bar{y})}$ enthalten sind. Außerdem kann ein y aus dem Ergebnis (x_k, y) von X_j mit $y \in Dom_k$ sicher nicht mehr verdrängt werden, wohingegen es im Falle $y \in NDom_k$ immer wieder verdrängt wird, es sei denn $j = \bar{t}_k$. Daher bleibt

$$\begin{aligned} P\{Y_k = \bar{y}\} & \quad \text{mit } \bar{y} \in Dom_k \\ & > \\ P\{Y_k = \bar{y}\} & \quad \text{mit } \bar{y} \in Ind_k \\ & > \\ P\{Y_k = \bar{y}\} & \quad \text{mit } \bar{y} \in NDom_k \end{aligned} \quad (3.86)$$

erhalten, sogar mit mehr Tendenz gegen den schlechtesten Fall $\bar{y} \in NDom_k$ und höherer Wahrscheinlichkeit für $\bar{y} \in Dom_k$ als bei SS1. Somit behalten die Aussagen der Sätze 3.2.28 und 3.2.30 ihre Richtigkeit.

SS3: Modifikation 2 der Definition von $Y_k, k \in \mathbb{N}_0$, in Suchstrategie SS1 Eine weitere alternative Definition der Arbeitsweise von Experiment $Y_k, k \in \mathbb{N}_0$, versucht abermals, die Wahrscheinlichkeit der Ausgänge $y_k \in Dom_k$ und $y_k \in Ind_k$ zu erhöhen, während es die für $y_k \in NDom_k$ im Vergleich zu der Definition von Y_k in SS1 verringert. Darüber hinaus strebt sie an, jedem Paretooptimum in Dom_k eine größere Chance, y_k zu werden, einzuräumen als jedem einzelnen nichteffizienten Punkt aus Dom_k , sofern natürlich $Dom_k \neq \emptyset$ ist. Ist $Dom_k = \emptyset$, wird das Gleiche für jede effiziente Alternative in Ind_k im Vergleich zu jeder nichteffizienten Alternative aus Ind_k beabsichtigt.

Wieder wird die Markov Kette $\{X_t\}_{t \in \mathbb{N}_0}$, die MA_k (NMA) modelliert, von Experiment $X_{\underline{t}_k}$ an beobachtet, aber nicht vor Iteration \bar{t}_k gestoppt. Der zweite Partner \bar{y} eines Alternativenpaares (x_k, \bar{y}) wird zum Ergebnis von Y_k , falls das Paar von einem Experiment $X_j, j \in \{\underline{t}_k, \dots, \bar{t}_k\}$, hervorgebracht wird und weder ein vorzuziehendes Ergebnis bei einem vorhergehenden Experiment seit $X_{\underline{t}_k}$ herauskam ("vorzuziehende" Vorgängerpaare sind alle (x_k, y) mit $y \notin OUT_{(x_k, \bar{y})}$), noch ein vorzuziehendes Ergebnis von einem nachfolgenden Experiment bis hin zu $X_{\bar{t}_k}$ produziert wird ("vorzuziehende" Nachfolgepaare sind alle (x_k, y) mit $y \notin IN_{(x_k, \bar{y})}$). Die Formel für $P\{Y_k = \bar{y}\}$ lautet genauso wie in (3.79), die Mengen $OUT_{(x_k, \bar{y})}$ und $IN_{(x_k, \bar{y})}$ sind jetzt jedoch etwas anders definiert als zuvor. Diesmal sind vier interessante Fälle zu unterscheiden:

1. $\bar{y} \in Dom_k$ effizient:

$$\begin{aligned} OUT_{(x_k, \bar{y})} &= NDom_k \cup Ind_k \cup (Dom_k \cap NDom_{\bar{y}}), \\ IN_{(x_k, \bar{y})} &= R_k, \end{aligned} \quad (3.87)$$

2. $\bar{y} \in Dom_k$ nicht effizient:

$$\begin{aligned} OUT_{(x_k, \bar{y})} &= NDom_k \cup Ind_k \cup (Dom_k \cap NDom_{\bar{y}}), \\ IN_{(x_k, \bar{y})} &= OUT_{(x_k, \bar{y})} \cup (Dom_k \cap Ind_{\bar{y}}), \end{aligned} \quad (3.88)$$

3. $\bar{y} \in Ind_k$:

$$\begin{aligned} OUT_{(x_k, \bar{y})} &= NDom_k \cup (Ind_k \cap NDom_{\bar{y}}), \\ IN_{(x_k, \bar{y})} &= OUT_{(x_k, \bar{y})} \cup (Ind_k \cap Ind_{\bar{y}}), \end{aligned} \quad (3.89)$$

4. $\bar{y} \in NDom_k$:

$$\begin{aligned} OUT_{(x_k, \bar{y})} &= NDom_k, \\ IN_{(x_k, \bar{y})} &= \emptyset. \end{aligned} \quad (3.90)$$

Sobald ein Alternativenpaar (x_k, \bar{y}) mit $\bar{y} \in Dom_k$ und effizient das erste sich zwischen den Experimenten Nummer \underline{t}_k und \bar{t}_k ergebende ist ohne vorhergehendem Paar $(x_k, y), y \notin OUT_{(x_k, \bar{y})}$, und ohne nachfolgendem Paar $(x_k, y), y \notin IN_{(x_k, \bar{y})}$, wird sein zweiter Partner sicherlich das Ergebnis von Y_k werden. Wenn jedes Paar in S_k von jedem Paar in S_k in einem Metropolis-Schritt erreichbar ist und alle Generierungswahrscheinlichkeiten kaum von $\frac{1}{|S_k|-1}$ abweichen, erhalten wir offensichtlich für $Dom_k \neq \emptyset$

$$\begin{aligned} P\{Y_k = \bar{y}\}, \quad &\text{falls } \bar{y} \in Dom_k \text{ und effizient} \\ &> \\ P\{Y_k = \hat{y}\}, \quad &\text{falls } \hat{y} \in Dom_k \text{ und } \bar{y} \prec \hat{y} \text{ oder } \hat{y} \notin Dom_k \\ (OUT_{(x_k, \hat{y})} \subset OUT_{(x_k, \bar{y})}, IN_{(x_k, \bar{y})} = R_k). \end{aligned} \quad (3.91)$$

Im Falle $Dom_k = \emptyset$ ergibt sich

$$\begin{aligned}
 &P\{Y_k = \bar{y}\}, \quad \text{falls } \bar{y} \in Ind_k \text{ und effizient} \\
 &\qquad\qquad\qquad > \\
 &P\{Y_k = \hat{y}\}, \quad \text{falls } \hat{y} \in Ind_k \text{ und } \bar{y} \prec \hat{y} \text{ oder } \hat{y} \notin Ind_k \\
 &(\text{OUT}_{(x_k, \hat{y})} \subset \text{OUT}_{(x_k, \bar{y})}, \text{IN}_{(x_k, \bar{y})} = R_k).
 \end{aligned} \tag{3.92}$$

Allen Paaren mit $(y_k, \bar{y}) \in Dom_k \cup Ind_k$ werden nicht nur vorhergehende Alternativenpaare mit niedrigerem Zielfunktionswert Z vorgezogen, sondern auch solche mit gleichem Zielfunktionswert und zweitem Partner, der nicht von \bar{y} dominiert wird. Die am meisten vernachlässigte Teilmenge von S_k umfaßt alle Paare (y_k, \bar{y}) mit $\bar{y} \in NDom_k$, da diese als Ergebnislieferanten von Y_k stets unterdrückt werden. Inwiefern sich die Wahrscheinlichkeitsverteilung vor allem zugunsten von paretooptimalen Alternativen $y_k \in Dom_k$ und weg von allen $y_k \in NDom_k$ im Vergleich zu der Definition von Y_k in SS1 verschiebt, zeigt sich in folgenden Ungleichungen:

$$\begin{aligned}
 P\{Y_k = y_k\} &> \frac{\exp\left(\frac{-min}{c}\right)}{|Dom_k| \exp\left(\frac{-min}{c}\right) + |Ind_k| \exp\left(\frac{-mid}{c}\right) + |NDom_k| \exp\left(\frac{-max}{c}\right)}, \\
 &\text{falls } y_k \prec x_k \text{ und } y_k \text{ effizient,}
 \end{aligned} \tag{3.93}$$

$$\begin{aligned}
 P\{Y_k = y_k\} &< \frac{\exp\left(\frac{-max}{c}\right)}{|Dom_k| \exp\left(\frac{-min}{c}\right) + |Ind_k| \exp\left(\frac{-mid}{c}\right) + |NDom_k| \exp\left(\frac{-max}{c}\right)}, \\
 &\text{falls } y_k \succ x_k.
 \end{aligned} \tag{3.94}$$

Satz 3.2.32. *In allen Metropolis-Algorithmen $MA_k, k \in \mathbb{N}_0$, sei jedes Alternativenpaar von jedem Alternativenpaar aus in einem Schritt erreichbar mit einer Wahrscheinlichkeit von $\frac{1}{|S_k|-1}$. Dann gilt:*

Die Realisierung $Y_0 = \bar{x}$ mit $\bar{x} \in Dom_0$ ($Dom_0 \neq \emptyset$) und effizient ist wahrscheinlicher als jede Realisierung $Y_0 = x_1$ mit $x_1 \in Ind_0 \cup NDom_0$, die x_0 irrtümlicherweise zu einem Paretooptimum erklärt und als jede Realisierung $Y_0 = \tilde{x}$ mit $\tilde{x} \in (Dom_0 \cap NDom_{\tilde{x}}) \cup Ind_0 \cup NDom_0$ bei beliebigen weiteren Realisierungen der nachfolgenden Experimente $Y_k, k \in \mathbb{N}$. Dabei mag \tilde{x} fälschlicherweise für paretooptimal gehalten werden oder lediglich eine Zwischenstation auf dem Weg zu einer ersten Alternative sein, die als paretooptimal gilt.

Satz 3.2.33. *Es gelten die Voraussetzungen von Satz 3.2.32.*

Sobald ein echtes Paretooptimum $\bar{x} \in R_D$ auf einem bestimmten Weg durch R_D in Experiment $Y_{l_1}, l_1 \in \mathbb{N}_0$, erreicht wurde und Ind_{l_1+1} keine leere Menge ist und Paretooptima enthält, wird ein weiteres Paretooptimum $\tilde{x} \in Ind_{l_1+1}$ mit größerer Wahrscheinlichkeit unmittelbar danach gefunden, d.h. $Y_{l_1} = \bar{x}$ und $Y_{l_1+1} = \tilde{x}, l_1 \in \mathbb{N}_0$, als daß Y_{l_1+1} eine nichteffiziente Alternative $\hat{x} \in R_{l_1+1}$ hervorbringt mit $\hat{x} \in NDom_{l_1+1}$ oder $\hat{x} \in (IND_{l_1+1} \cap NDom_{\tilde{x}})$. Dabei mag \hat{x} fälschlicherweise für paretooptimal gehalten werden oder lediglich eine Zwischenstation auf dem Weg zu einer nächsten Alternative sein, die als paretooptimal gilt.

Die Aussagen beider Sätze ergeben sich aus den Gleichungen 3.91 und 3.92.

Suchstrategie PS

Suchstrategie PS wird von mehreren parallel arbeitenden stochastischen Superprozessen ausgeführt.

Eine Möglichkeit, mehr als ein Paretooptimum in einer Suche von einer Start-Referenzalternative aus einzusammeln besteht darin, Listen $L_k, k \in \mathbb{N}_0$, mit $s_k \geq 1$ Elementen zu erarbeiten. Bei jedem Eintrag $x_k^{i_k}, i_k \in \{1, \dots, s_k\}, k \in \mathbb{N}_0$, von L_k erinnert man sich der Alternativen $z_1, \dots, z_r, r \in \mathbb{N}$, von denen man im Laufe der Suchprozeduren erfahren hat, daß sie von $x_k^{i_k}$ dominiert werden, damit man die Menge $R_k^{i_k} := R_D \setminus (\cup_{j=1}^r E^{(z_j)D} \cup E^{(x_k^{i_k})D})$ erhält. $S_k^{i_k} := \{x_k^{i_k}\} \times R_k^{i_k}$ ist der Raum, in dem die Metropolis-Prozedur $MA_k^{i_k}$ (NMA) nach Paaren $(x_k^{i_k}, y)$ mit $y \prec x_k^{i_k}$ sucht. Die Mengenbezeichnungen $Dom_k^{i_k}, Ind_k^{i_k}$ und $NDom_k^{i_k}$ werden entsprechend Notation 3.2.24 gebraucht.

Die Realisierungen des Zufallsexperimentes Y_k des stochastischen Superprozesses $\{Y_k\}_{k \in \mathbb{N}_0}$ sind jetzt ganze Teilmengen von $\cup_{i=1}^{s_k} R_k^i$ (anstatt einzelner Alternativen). Y_k besteht aus s_k parallel laufenden Metropolis-Prozeduren $MA_k^1, MA_k^2, \dots, MA_k^{s_k}$ (NMA). Bevor ein Experiment Y_k des Superprozesses startet, müssen für alle $i_k \in \{1, \dots, s_k\}$ die Iterationszahlen $\underline{t}_k^{i_k} < \bar{t}_k^{i_k}$, jede aus \mathbb{N} , für die Markov Kette gewählt werden, die den IMA modelliert, von dem $MA_k^{i_k}$ eine numerische Version ist. $\underline{t}_k^{i_k}$ sollte jeweils groß genug sein, daß dem Experiment Nummer $\underline{t}_k^{i_k}$ der entsprechenden Markov Kette fast die stationäre Verteilung zugrunde liegt. Von Iteration $\underline{t}_k^{i_k}$ an wird $MA_k^{i_k}$ (NMA) wieder und wieder darum "gebeten", die zweiten Partner y der Standpunkte $(x_k^{i_k}, y) \in S_k^{i_k}$ der Liste L_{k+1} beizusteuern, sobald $y \prec x_k^{i_k}$ gilt und kein Eintrag von L_{k+1} diese Punkte y dominiert. Hier gibt es nur eine sinnvolle Definition für Y_k und daher keine Modifikationen.

Die Wahrscheinlichkeit dafür, daß eine Alternative $\bar{y} \in \cup_{i=1}^{s_k} Dom_k^i$ ein Mitglied der Liste L_{k+1} wird, hängt von drei Faktoren ab:

- von der Anzahl von Elementen $y \in \cup_{i=1}^{s_k} Dom_k^i$, die \bar{y} wieder aus L_{k+1} "hinauswerfen" können - dies ist der Fall, wenn $y \prec \bar{y}$ - zusammen mit der Wahrscheinlichkeit, mit der diese y von einer Suchprozedur $MA_k^{i_k}$ (NMA), $i_k = 1, \dots, s_k$, als Kandidaten erzeugt werden; letztere Wahrscheinlichkeit hängt auch davon ab, wie oft jedes $MA_k^j, j \in \{1, \dots, s_k\}$, mit solch einem $y \in Dom_k^j$ dazu angehalten wird, einen Kandidaten zur Aufnahme in die Liste zu stellen. Sobald eine effiziente Alternative aus $\cup_{i=1}^{s_k} Dom_k^i$ in L_{k+1} gelangt ist, kann sie natürlich nicht mehr "hinausgeworfen" werden.
- von der Anzahl von Elementen $y \neq \bar{y}$ in $\cup_{i=1}^{s_k} Dom_k^i$, die bereits in L_{k+1} eingetragen sind und die Aufnahme von \bar{y} verhindern können - dies tritt bei $y \prec \bar{y}$ ein - zusammen mit der Wahrscheinlichkeit, mit der diese y bereits von einer Metropolis-Prozedur (NMA) der Stufe k erzeugt wurden, die L_{k+1} versorgt; wieder hängt letztere Wahrscheinlichkeit auch davon ab, wie oft jedes $MA_k^j, j \in \{1, \dots, s_k\}$, mit so einem $y \in Dom_k^j$ aufgefordert wurde, einen Kandidaten zur Aufnahme in die Liste zu präsentieren. Jedes Paretooptimum aus $Dom_k^j, j \in \{1, \dots, s_k\}$, das noch nicht Element von L_{k+1} ist, kann nicht am Beitritt zu L_{k+1} gehindert werden.
- von der Zahl der Einträge $x_k^{i_k}, i_k \in \{1, \dots, s_k\}$, der Liste L_k , bei denen \bar{y} in $Dom_k^{i_k}$ enthalten ist; insbesondere, wenn jedes $MA_k^{i_k}$ (NMA), $i_k \in \{1, \dots, s_k\}$, gleichoft dazu angehalten wird, einen Kandidaten für die Liste L_{k+1} zu liefern, ist es vorteilhaft für \bar{y} , in so vielen

Mengen $Dom_k^{i_k}$ wie möglich zu liegen. Wann auch immer in diesem Fall ein nichteffizienter Punkt \bar{y} als Element vieler Mengen $Dom_k^{i_k}$ privilegiert ist, ist jedes Paretooptimum, das diesen Punkt dominiert, sicher auch Element all dieser Mengen und eventuell sogar von noch anderen.

Daher und insbesondere, wenn jede Prozedur $MA_k^{i_k}, i_k = 1, \dots, s_k$, gleichoft dazu kommt, einen Kandidaten für L_{k+1} zu stellen, werden von Suchstrategie PS effiziente Alternativen aus $\cup_{i=1}^{s_k} Dom_k^i$ gefördert in dem Sinne, daß im Vergleich zu jedem Punkt $\bar{y} \in \cup_{i=1}^{s_k} Dom_k^i$ jedes Paretooptimum \hat{y} , das \bar{y} dominiert, eine größere Chance bekommt, Kandidat für L_{k+1} zu werden, darin aufgenommen und behalten zu werden.

3.2.9 Algorithmus DKMOP SS2

$$D_{R_k} := \{x_D \in D : (x_c, c_D) \in R_k, x_c \in C\}, k \in \mathbb{N}_0$$

MA₀	Schritt 0	Zufallswahl der Start-Referenzalternative $x_0 \in R_D$; dabei Zufallswahl von $(x_0)_D \in D$, Erzeugen von $E^{(x_0)_D}$ und Zufallswahl von x_0 aus $E^{(x_0)_D}$; Festlegung von $c \in \mathbb{R}^+$;
	Schritt 1	$R_0 := R_D \setminus E^{(x_0)_D}$; Festlegung der Nachbarschaftsstruktur auf dem Suchraum $\{x_0\} \times R_0$; Festlegung von $\underline{t}_0, \bar{t}_0 \in \mathbb{N}, \underline{t}_0 \leq \bar{t}_0$;
	Schritt 2	Zufallswahl des Startpunktepaars (x_0, y_0^0) für MA_0 (NMA) durch Zufallswahl einer Alternative $y_0^0 \in R_0$; dabei Zufallswahl von $(y_0^0)_D \in D_{R_0}$, Erzeugen von $E^{(y_0^0)_D}$, falls noch nicht vorhanden, und Zufallswahl von y_0^0 aus $E^{(y_0^0)_D}$; setze $k = 0$ und $i = 1$;
	Schritt 3	Zufallswahl eines Nachbarn (x_0, y_0^{prop}) von (x_0, y_0^{i-1}) , die den festgelegten Generierungswahrscheinlichkeiten folgt, durch Zufallswahl einer Alternative $y_0^{prop} \in R_0$; dabei Zufallswahl von $(y_0^{prop})_D \in D_{R_0}$, Erzeugen von $E^{(y_0^{prop})_D}$, falls noch nicht vorhanden, und Zufallswahl von y_0^{prop} aus $E^{(y_0^{prop})_D}$; Akzeptanz von (x_0, y_0^{prop}) als neuen Standpunkt (x_0, y_0^i) mit Wahrscheinlichkeit $\exp\left(\frac{-(Z[(x_0, y_0^{prop})] - Z[(x_0, y_0^{i-1}]])^+}{c}\right)$; im Ablehnungsfall setze $(x_0, y_0^i) = (x_0, y_0^{i-1})$; falls $i = \underline{t}_0$, setze $y_0 = y_0^i$; falls $Z[(x_0, y_0^i)] \leq Z[(x_0, y_0)]$ bei $i \geq \underline{t}_0$, setze $y_0 = y_0^i$; falls $y_0^i \prec x_0$ bei $i \geq \underline{t}_0$, gehe zu Schritt 4 ; ansonsten, falls $i = \bar{t}_0$, zu Schritt 5 ; ansonsten setze $i = i + 1$ und gehe zurück zu Schritt 3 ;
	Schritt 4	setze $k = k + 1$; Starten einer weiteren Metropolis-Suche MA_k (NMA) mit Referenzalternative $x_k = y_{k-1}$; Verkleinerung des Suchraumes $\{x_k\} \times R_k$: $R_k := R_{k-1} \setminus E^{(x_k)_D}$;
	Schritt 5	x_k ist als paretooptimal eingestuft; Eintragung als x_0^{po} in Lösungsmenge; setze $l = 1$; setze $k = k + 1$; Starten einer weiteren Metropolis-Suche MA_k (NMA) mit Referenzalternative $x_k = y_{k-1}$; die bisherige Verkleinerung des Suchraumes $\{x_k\} \times R_k$ geht verloren: $R_k := R_D \setminus E^{(x_k)_D}$;

MA_k	<p>Schritt 1 Festlegung der Nachbarschaftsstruktur auf dem Suchraum $\{x_k\} \times R_k$; Festlegung von $\underline{t}_k, \bar{t}_k \in \mathbb{N}, \underline{t}_k \leq \bar{t}_k$;</p> <p>Schritt 2 Zufallswahl des Startpunktpaares (x_k, y_k^0) für MA_k (NMA) durch Zufallswahl einer Alternative $y_k^0 \in R_k$; dabei Zufallswahl von $(y_k^0)_D \in D_{R_k}$, Erzeugen von $E^{(y_k^0)}_D$, falls noch nicht vorhanden, und Zufallswahl von y_k^0 aus $E^{(y_k^0)}_D$; setze $i = 1$;</p> <p>Schritt 3 Zufallswahl eines Nachbarn (x_k, y_k^{prop}) von (x_k, y_k^{i-1}), die den festgelegten Generierungswahrscheinlichkeiten folgt, durch Zufallswahl einer Alternative $y_k^{prop} \in R_k$; dabei Zufallswahl von $(y_k^{prop})_D \in D_{R_k}$, Erzeugen von $E^{(y_k^{prop})}_D$, falls noch nicht vorhanden, und Zufallswahl von y_k^{prop} aus $E^{(y_k^{prop})}_D$; Akzeptanz von (x_k, y_k^{prop}) als neuen Standpunkt (x_k, y_k^i) mit Wahrscheinlichkeit $\exp\left(\frac{-(Z[(x_k, y_k^{prop})] - Z[(x_k, y_k^{i-1}]])^+}{c}\right)$; im Ablehnungsfall setze $(x_k, y_k^i) = (x_k, y_k^{i-1})$; falls $i = \underline{t}_k$, setze $y_k = y_k^i$; falls $Z[(x_k, y_k^i)] \leq Z[(x_k, y_k)]$ bei $i \geq \underline{t}_k$, setze $y_k = y_k^i$; falls $y_k^i \prec x_k$ bei $i \geq \underline{t}_k$, gehe zu Schritt 4; ansonsten, falls $i = \bar{t}_k$, zu Schritt 5; ansonsten setze $i = i + 1$ und gehe zurück zu Schritt 3;</p> <p>Schritt 4 falls keine weitere Rechenzeit mehr zur Verfügung, stoppe die Suche in R_D; ansonsten setze $k = k + 1$; Starten einer weiteren Metropolis-Suche MA_k (NMA) mit Referenzalternative $x_k = y_{k-1}$; Verkleinerung des Suchraumes $\{x_k\} \times R_k$: $R_k := R_{k-1} \setminus E^{(x_k)}_D$;</p> <p>Schritt 5 x_k ist als paretooptimal eingestuft; Eintragung als x_l^{po} in Lösungsmenge; setze $l = l + 1$; falls die Anzahl der bereits eingesammelten Paretooptima ausreichend ist oder keine weitere Rechenzeit zur Verfügung steht, stoppe die Suche in R_D; ansonsten setze $k = k + 1$; Starten einer weiteren Metropolis-Suche MA_k (NMA) mit Referenzalternative $x_k = y_{k-1}$; die bisherige Verkleinerung des Suchraumes $\{x_k\} \times R_k$ geht verloren: $R_k := R_D \setminus E^{(x_k)}_D$;</p>
--------	---

3.2.10 Algorithmus DKMOP PS

Wie bei DKMOP SS2 läuft auch hier die Zufallswahl einer Alternative aus R_D dreistufig ab, was der Einfachheit halber nicht mehr explizit erwähnt wird. In der folgenden Darstellung ist j für alle $MA_k^i, k \in \mathbb{N}_0$, eigentlich ein eigenes j_i . Aus Gründen der Darstellbarkeit wird auf die zusätzliche Indizierung verzichtet.

Stufe 0	Liste L_0	Zufallswahl (bei Gleichverteilung) von $s_0 \in \mathbb{N}$ paarweise verschiedenen Alternativen $\in R_D$, die für alle $i_0 \neq j_0, i_0, j_0 \in \{1, \dots, s_0\}, x_0^{i_0} \succ x_0^{j_0}$ erfüllen; setze $k = 0$; Festlegung der Zahl $numcont \in \mathbb{N}$ dominierender Alternativen, die jede Metropolis-Prozedur (NMA) mit Referenzalternative aus L_0 der Liste L_1 beizusteuern hat; setze $i = 1$; Festlegung von $c \in \mathbb{R}^+$;
MA_0^i	Schritt 0	Verkleinerung des Suchraumes $\{x_0^i\} \times R_0^i: Red_0^i := E^{(x_0^i)^D}; R_0^i := R_D \setminus Red_0^i$; Festlegung der Nachbarschaftsstruktur auf dem Suchraum $\{x_0^i\} \times R_0^i$; Festlegung von $\underline{t}_0^i, \bar{t}_0^i \in \mathbb{N}, \underline{t}_0^i \leq \bar{t}_0^i$; setze $count_0^i = 0$;
	Schritt 1	Zufallswahl einer Alternative $y^j \in R_0^i, j = 0$, für das Startpunktpaar $(x_0^i, y^j) \in \{x_0^i\} \times R_0^i$ von MA_0^i (NMA); setze $j = j + 1$;
	Schritt 2	Zufallswahl eines Nachbarn (x_0^i, y^{prop}) von (x_0^i, y^{j-1}) , die den festgelegten Generierungswahrscheinlichkeiten folgt; Akzeptanz von (x_0^i, y^{prop}) als neuen Standpunkt (x_0^i, y^j) mit Wahrscheinlichkeit $\exp\left(\frac{-(Z[(x_0^i, y^{prop})] - Z[(x_0^i, y^{j-1}]])^+}{c}\right)$; ansonsten setze $(x_0^i, y^j) = (x_0^i, y^{j-1})$; falls $j = \underline{t}_0^i$, setze $z_0^i = y^j$; falls $Z[(x_0^i, y^j)] \leq Z[(x_0^i, z_0^i)]$ bei $j \geq \underline{t}_0^i$, setze $z_0^i = y^j$; falls $y^j \prec x_0^i$ bei $j \geq \underline{t}_0^i$, gehe zu Schritt 3 ; ansonsten, falls $j = \bar{t}_0^i$ mit $Z[(x_0^i, z_0^i)] > min$, zu Schritt 4 ; ansonsten setze $j = j + 1$ und gehe zurück zu Schritt 2 ;
	Schritt 3	$Red_{y^j} := E^{(y^j)^D} \cup Red_0^i$; falls y^j nicht von einem Element von L_1 dominiert wird, wird y^j Eintrag $x_1^{i_1}, i_1 \in \mathbb{N}$, von L_1 ; für jedes Element $x_1^{\hat{i}_1} \in L_1, \hat{i}_1 \in \mathbb{N}$, das von y^j dominiert wird, ergibt sich $Red_{y^j} = Red_{y^j} \cup Red_0^{\hat{i}_1}$, und $x_1^{\hat{i}_1}$ wird aus L_1 entfernt; falls $x_1^{\hat{i}_1}$ ein Beitrag von MA_r^i (NMA), $r \in \{1, 2, \dots, s_0\}$, war, setze $count_0^r = count_0^r - 1$; setze $count_0^i = count_0^i + 1$ und $Red_1^i = Red_{y^j}$; setze $j = j + 1$; gehe zu Schritt 5 ; wird y^j nicht in L_1 aufgenommen, setze für jedes $x_1^{\hat{i}_1} \in L_1$, das y^j dominiert, $Red_1^{\hat{i}_1} = Red_1^{\hat{i}_1} \cup Red_{y^j}$;
	Schritt 4	x_0^i ist als paretooptimal eingestuft und wird als x_0^{po} in der Lösungsmenge gespeichert; setze $l = 1$; gehe zu Schritt 5 ;
	Schritt 5	falls für jede Prozedur $MA_0^{i_0}$ (NMA), $i_0 \in \{1, \dots, s_0\}$, die maximale Rechenzeit für das Auffüllen von L_1 überschritten ist, setze $k = 1$ und gehe zu Stufe k , ansonsten suche nach dem nächsten $i \in \{1, 2, \dots, s_0\}$ mit $count_0^i < numcont$, bei dem MA_0^i (NMA) noch Rechenzeit zur Verfügung hat; falls es kein solches mehr gibt, setze $k = 1$ und gehe zu Stufe k ; ansonsten Fortsetzung von MA_0^i ab Schritt 2;

Stufe k	Liste L_k	Falls $L_k = \emptyset$, stoppe die Suche in R_D - für keines der Elemente von L_{k-1} konnte in zulässiger Rechenzeit eine dominierende Alternative gefunden werden; alle Elemente werden als paretooptimal eingestuft; ansonsten setze $s_k = L_k $; Festlegung der Zahl $numcont \in \mathbb{N}$ dominierender Alternativen, die jede Metropolis-Prozedur (NMA) mit Referenzalternative aus L_k der Liste L_{k+1} beizusteuern hat; setze $i = 1$;
MA_k^i	Schritt 0	Verkleinerung des Suchraumes $\{x_k^i\} \times R_k^i$: $R_k^i := R_D \setminus Red_k^i$; Festlegung der Nachbarschaftsstruktur auf dem Suchraum $\{x_k^i\} \times R_k^i$; Festlegung von $\underline{t}_k^i, \bar{t}_k^i \in \mathbb{N}, \underline{t}_k^i \leq \bar{t}_k^i$; setze $count_k^i = 0$;
	Schritt 1	Zufallswahl einer Alternative $y^j \in R_k^i, j = 0$, für das Startpunktpaar $(x_k^i, y^j) \in \{x_k^i\} \times R_k^i$ von MA_k^i (NMA); setze $j = j + 1$;
	Schritt 2	Zufallswahl eines Nachbarn (x_k^i, y^{prop}) von (x_k^i, y^{j-1}) , die den festgelegten Generierungswahrscheinlichkeiten folgt; Akzeptanz von (x_k^i, y^{prop}) als neuen Standpunkt (x_k^i, y^j) mit Wahrscheinlichkeit $\exp\left(\frac{-(Z[(x_k^i, y^{prop})] - Z[(x_k^i, y^{j-1}]])^+}{c}\right)$; ansonsten setze $(x_k^i, y^j) = (x_k^i, y^{j-1})$; falls $j = \underline{t}_k^i$, setze $z_k^i = y^j$; falls $Z[(x_k^i, y^j)] \leq Z[(x_k^i, z_k^i)]$ bei $j \geq \underline{t}_k^i$, setze $z_k^i = y^j$; falls $y^j \prec x_k^i$ bei $j \geq \underline{t}_k^i$, gehe zu Schritt 3 ; ansonsten, falls $j = \bar{t}_k^i$ mit $Z[(x_k^i, z_k^i)] > min$, zu Schritt 4 ; ansonsten setze $j = j + 1$ und gehe zurück zu Schritt 2 ;
	Schritt 3	$Red_{y^j} := E^{(y^j)D} \cup Red_k^i$; falls y^j nicht von einem Element von L_{k+1} dominiert wird, wird y^j Eintrag $x_{k+1}^{i_{k+1}}, i_{k+1} \in \mathbb{N}$, von L_{k+1} ; für jedes Element $x_{k+1}^{\hat{i}_{k+1}} \in L_{k+1}, \hat{i}_{k+1} \in \mathbb{N}$, das von y^j dominiert wird, ergibt sich $Red_{y^j} = Red_{y^j} \cup Red_{k+1}^{\hat{i}_{k+1}}$, und $x_{k+1}^{\hat{i}_{k+1}}$ wird aus L_{k+1} entfernt; falls $x_{k+1}^{\hat{i}_{k+1}}$ ein Beitrag von MA_k^r (NMA), $r \in \{1, 2, \dots, s_k\}$, war, setze $count_k^r = count_k^r - 1$; setze $count_k^i = count_k^i + 1$ und $Red_{k+1}^{i_{k+1}} = Red_{y^j}$; setze $j = j + 1$; gehe zu Schritt 5 ; wird y^j nicht in L_{k+1} aufgenommen, setze für jedes $x_{k+1}^{\hat{i}_{k+1}} \in L_{k+1}$, das y^j dominiert, $Red_{k+1}^{\hat{i}_{k+1}} = Red_{k+1}^{\hat{i}_{k+1}} \cup Red_{y^j}$;
	Schritt 4	x_k^i ist als paretooptimal eingestuft und wird als x_l^{po} in der Lösungsmenge gespeichert; setze $l = l + 1$; gehe zu Schritt 5 ;
	Schritt 5	falls für jede Prozedur MA_k^i (NMA), $i_k \in \{1, \dots, s_k\}$, die maximale Rechenzeit für das Auffüllen von L_{k+1} überschritten ist, gehe zu Stufe $k + 1$, ansonsten suche nach dem nächsten $i \in \{1, 2, \dots, s_k\}$ mit $count_k^i < numcont$, bei dem MA_k^i (NMA) noch Rechenzeit zur Verfügung hat; falls es kein solches mehr gibt, setze $k = k + 1$ und gehe zu Stufe k ; ansonsten Fortsetzung von MA_k^i ab Schritt 2;

Kapitel 4

Anwendungen und Testergebnisse

4.1 Optimale Auslegung einer Vorwärmstrecke in einem fossil befeuerten Kraftwerk

Die Ausführungen über technische und thermodynamische Grundlagen und Teile der Beschreibung des Optimierungsproblems orientieren sich an [34].

4.1.1 Kurze Einführung in die technischen und thermodynamischen Grundlagen

4.1.1.1 Der Wasser-/Dampfkreislauf

Kraftwerke erzeugen elektrischen Strom, indem sie thermische in elektrische Energie umwandeln. In fossil befeuerten Kraftwerken wird die thermische Energie aus der Verbrennung von fossilen Primärenergieträgern wie z.B. Kohle, Gas oder Öl gewonnen. Ein Arbeitsfluid, im Allgemeinen Wasser bzw. Wasserdampf, durchläuft einen Kreisprozeß, nimmt dabei Energie in Form von Wärme und mechanischer Arbeit auf und gibt sie wieder ab.

Die wesentlichen Elemente eines Dampfkraftwerkes sind

- der Dampferzeuger,
- die Turbinen,
- der Generator,
- der Kondensator,
- die Speisewasserpumpe.

Im Dampferzeuger wird das sogenannte Speisewasser erhitzt und verdampft dann. Der so entstandene heiße Dampf, der unter hohem Druck steht, heißt Frischdampf. Dieser Frischdampf wird Turbinen zugeführt, die einen Generator antreiben, der wiederum elektrischen Strom erzeugt. Die Größe der verschiedenen eingesetzten Turbinen hängt von dem Druck des Frischdampfes ab, der sie erreicht - man unterscheidet zwischen einer Hoch-, Mittel- und Niederdruckzone. In den Turbinen wird der Druck, unter dem der Frischdampf steht, entspannt, d.h. Druck und Wärme werden reduziert. Etwas später kondensiert der abgekühlte Dampf mit Hilfe von Kühlwasser (\implies Speisewasser). Die Speisewasserpumpe bringt das Speisewasser zurück auf das Druckniveau des Frischdampfes und wieder erreicht das Speisewasser den Dampferzeuger.

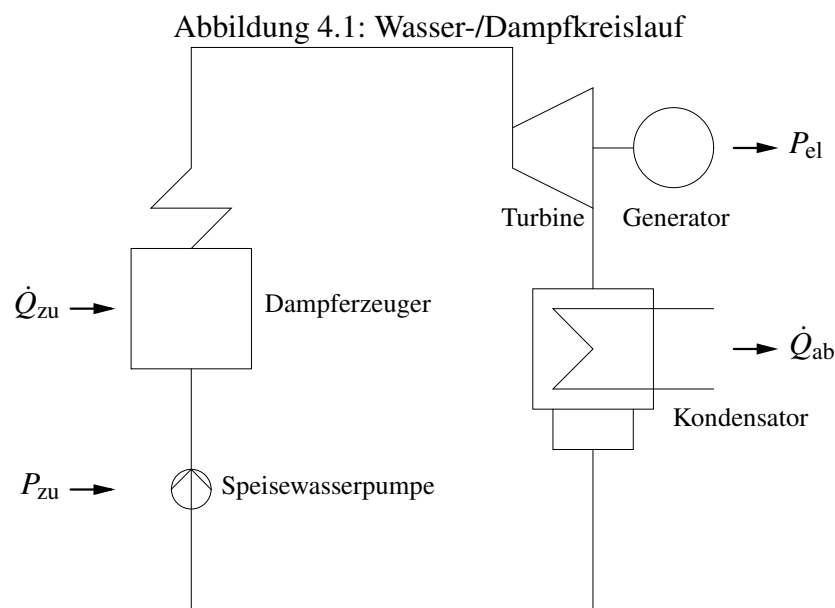


Abbildung 4.1 zeigt den beschriebenen Prozeß. P_{zu} steht für die Energie, die in der Speisewasserpumpe, \dot{Q}_{zu} für die Wärmemenge, die im Dampferzeuger zugeführt wird. P_{el} repräsentiert die elektrische Leistung, die der Generator erzeugt, und \dot{Q}_{ab} die Wärmemenge, die dem Dampf im Kondensator vom Kühlwasser abgenommen wird.

4.1.1.2 Das Prinzip der Speisewasservorwärmung

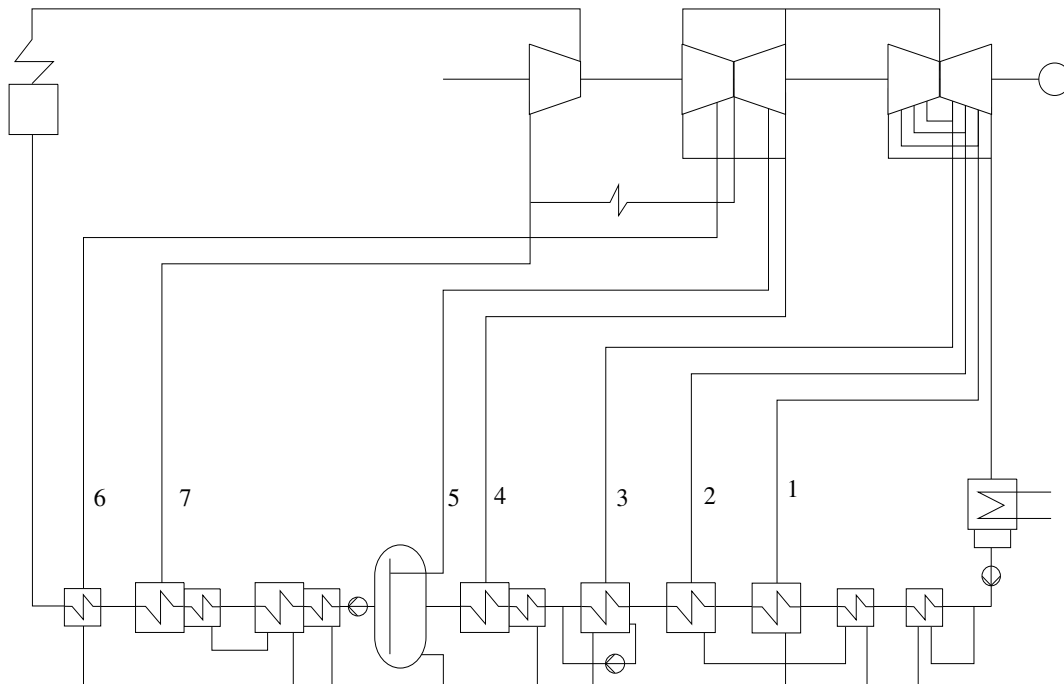
Das Vorwärmen des Speisewassers stellt eine Möglichkeit dar, den Wirkungsgrad des Dampfkraftwerks, der das Verhältnis aus erzeugter elektrischer Energie zur zugeführten thermischen Energie bezeichnet, zu steigern:

$$\eta := \frac{P_{el}}{\dot{Q}_{Brennstoff}}. \quad (4.1)$$

Teilentspannter Dampf aus den Turbinen wird in Wärmetauscher geleitet, um die Speisewassertemperatur zu erhöhen. Diese Wärmetauscher werden Speisewasservorwärmer genannt. Mehrere hintereinandergeschaltete Vorwärmer bilden die sogenannte Vorwärmstrecke.

Abbildung 4.2 zeigt ein modernes Dampfkraftwerk mit Speisewasservorwärmung. Oben rechts im Bild sind drei Turbinen für Hoch-, Mittel- und Niederdruck dargestellt. Anzapfleitungen verbinden die Turbinen mit den Wärmetauschern der Vorwärmstrecke, die unten im Bild zu sehen

Abbildung 4.2: Dampfkraftwerk mit Vorwärmstrecke



ist. Steigend mit dem Anzapfdruck wächst die Numerierung der Vorwärmer. Der Anzapfdruck fällt längs der Turbinenströmung streng monoton ab. Daher sind Anzapfungen und Vorwärmer eindeutig einander zugeordnet.

Vorwärmer werden gemäß dem Aggregatzustand des wärmeabgebenden Mediums folgendermaßen klassifiziert:

- **Enthitzer:** Der Anzapfdampf gibt Wärme ab, bleibt aber in gasförmigem Zustand - das wärmeabgebende Medium wird dem Wärmetauscher gasförmig zugeführt und verläßt ihn auch gasförmig (Dampf \mapsto Dampf).
- **Kondensationsvorwärmer:** Indem er Wärme abgibt, kondensiert der Anzapfdampf - das wärmeabgebende Medium erreicht den Wärmetauscher als Dampf, tritt jedoch als Wasser aus (Dampf \mapsto Wasser).
- **Kondensatkühler:** Hier kühlt sich das wärmeabgebende Medium Wasser weiter ab und ändert seinen Aggregatzustand nicht (Wasser \mapsto Wasser).

Ein Speisewasservorwärmer besteht mindestens aus einem Kondensationsvorwärmer. Er kann noch durch einen Enthitzer und/oder einen Kondensatkühler erweitert sein. All diese Elemente sind entweder in einem Gehäuse integriert oder separat installiert. Im Falle der separaten Ausführung können die Komponenten längs der Vorwärmstrecke verschoben werden - Enthitzer werden relativ zu den Kondensationsvorwärmern vorgezogen, Kondensatkühler nachgeschaltet. Der Speisewasserbehälter fällt unter den Vorwärmern etwas aus dem Rahmen (siehe Anzapfung 5 in Abbildung 4.2): Hier wird der Anzapfdampf direkt in den Behälter eingeleitet und kondensiert an der Oberfläche. Der Speisewasserbehälter dient als Entgaser und übernimmt für das Speisewasser eine Art Pufferrolle bei Funktionsstörungen.

4.1.1.3 Auslegungsparameter der Vorwärmstrecke

Der Begriff Anlagenauslegung umfaßt die Anordnung und Spezifizierung von Komponenten und der Verbindungen zwischen diesen. Möglichkeiten der Auslegung einer Vorwärmstrecke liegen vor allem in den Entscheidungen über Vorhanden- und Nichtvorhandensein von Speisewasservorwärmern und bei vorhandenen Apparaten in der Spezifizierung und Positionierung von Turbinenanzapfungen und Vorwärmern. Bei den Anzapfungen bestimmt bereits der Anzapfdruck ihre Position - der Druck nimmt längs der Turbinenströmung ab. Die Eigenschaften eines Wärmetauschers hängen vor allem von seiner **Grädigkeit** ab. Sie bezeichnet den Temperaturunterschied zwischen wärmeabgebendem und wärmeaufnehmendem Medium und beeinflußt die Größe des Speisewasservorwärmers, da dessen Größe mit sinkender Grädigkeit zunehmen muß.

Der Kraftwerkhersteller definiert die Grädigkeiten ϑ der verschiedenen Wärmetauscher wie folgt:

- Enthitzer: $\vartheta_E := T_{D,ab} - T_{SpW,zu}$,
- Kondensationsvorwärmer: $\vartheta_{KV} := T_{Sat} - T_{SpW,ab}$,
- Kondensatkühler: $\vartheta_{KK} := T_{K,ab} - T_{SpW,zu}$,

wobei $T_{D,ab}$ und $T_{K,ab}$ die Temperatur des Dampfes bzw. Kondensates bezeichnen, das den Wärmetauscher verläßt, $T_{SpW,zu}$ und $T_{SpW,ab}$ für die Temperatur des Speisewassers stehen, das dem Wärmetauscher zugeführt bzw. aus ihm herausgeleitet wird, und T_{Sat} die Temperatur repräsentiert, bei der der Dampf kondensiert (Sättigungstemperatur).

4.1.2 Das Optimierungsproblem

Was kann man an der Auslegung der Vorwärmstrecke so lange variieren, bis die optimale Auslegung (was darunter zu verstehen ist, muß noch definiert werden) gefunden ist? Es sind die Existenz, Position und Spezifizierung der Komponenten und der Verbindungen zwischen den existierenden Bestandteilen.

4.1.2.1 Diskrete Variablen

Diskrete Optimierungsvariablen ergeben sich aus strukturellen Variationsmöglichkeiten. Zu diesen zählen z.B. Entscheidungen über das Vorhandensein und die Position von vorhandenen Vorwärmern oder über die integrierte oder separate Anbringung von Enthitzer, Kondensationsvorwärmer und Kondensatkühler.

Falls ein diskreter Parameter $d_k, k \in \{0, \dots, m_2 - 1\}$, Element einer Menge $\{lb_{k+m_1+1}, \dots, ub_{k+m_1+1}\}$

- mit $lb_{k+m_1+1} = 0$ und $ub_{k+m_1+1} = 1$ ist, dann drückt d_k Nichtvorhandensein / eine "Nein"-Entscheidung ($d_k = 0$) oder Vorhandensein / eine "Ja"-Entscheidung ($d_k = 1$) aus,
- mit $lb_{k+m_1+1} < ub_{k+m_1+1}, lb_{k+m_1+1}, ub_{k+m_1+1} \in \mathbb{Z}$ ist, dann bezeichnet d_k die Position einer Komponente, oder genauer gesagt, die Anzahl der Verschiebungsschritte relativ zu einer Standardposition.

$m_2 \in \mathbb{N}$ sei dabei die Anzahl aller diskreten Parameter, und $m_1 \in \mathbb{N}_0$ wird noch für eine spätere Notation gebraucht.

Der Begriff "Anzapfung" steht hier nicht nur für die tatsächliche Turbinenanzapfung, sondern umfaßt sowohl alle mit ihr verbundenen Speisewasservorwärmer als auch zusätzliche Komponenten wie Kondensatpumpen oder Kondensateinleitungen.

Sei $\alpha \in \mathbb{N}$ die **maximale Anzahl von Anzapfungen**. Dann gibt es für alle $i \in \{1, \dots, \alpha\}$ die folgenden Parameter $\tilde{d}_{i,j} := d_{(i-1)9+j}, j = 1, \dots, 9$, für die Strukturbeschreibung der Anzapfung i :

$\tilde{d}_{i,1} \in \{0, 1\}$	Anzapfung vorhanden
$\tilde{d}_{i,2} \in \{0, 1\}$	Enthitzer vorhanden
$\tilde{d}_{i,3} \in \{0, 1\}$	Enthitzer separat installiert
$\tilde{d}_{i,4} \in \{0, 1, 2, 3\}$	relative Enthitzerposition
$\tilde{d}_{i,5} \in \{0, 1\}$	Kondensatpumpe vorhanden
$\tilde{d}_{i,6} \in \{0, 1\}$	Kondensatkühler vorhanden
$\tilde{d}_{i,7} \in \{0, 1\}$	Kondensatkühler separat installiert
$\tilde{d}_{i,8} \in \{0, 1, 2\}$	relative Kondensatkühlerposition
$\tilde{d}_{i,9} \in \{1, 2, 3\}$	Spezifikation der Kondensateinleitung.

Ein weiterer Parameter steht für die Nummer der Anzapfung, die mit dem Speisewasserbehälter verbunden ist:

$$d_0 \in \{lb_{m_1+1}, \dots, ub_{m_1+1}\}, 1 \leq lb_{m_1+1} \leq ub_{m_1+1} \leq \alpha, lb_{m_1+1}, ub_{m_1+1} \in \mathbb{N}.$$

Alle diskreten Parameter werden nun in dem Vektor

$$\begin{aligned} d &= (d_0, \tilde{d}_{1,1}, \tilde{d}_{1,2}, \dots, \tilde{d}_{1,9}, \tilde{d}_{2,1}, \tilde{d}_{2,2}, \dots, \tilde{d}_{2,9}, \dots, \tilde{d}_{\alpha,9}) \\ &= (d_0, d_1, d_2, \dots, d_9, d_{10}, \dots, d_{18}, \dots, d_{9\alpha}) \end{aligned}$$

zusammengefaßt. Er codiert somit die strukturelle Auslegung der Vorwärmstrecke. Sofern diese Parameter variabel sind, kann man sie innerhalb von bestimmten Box-Constraints variieren. Sollte eine Komponente nicht existieren, sind natürlich weitere sie betreffende strukturelle Entscheidungen entweder bereits festgelegt oder stehen erst gar nicht zur Debatte.

4.1.2.2 Kontinuierliche Variablen

Kontinuierliche Variablen ergeben sich aus den Möglichkeiten, gewisse technische Eigenschaften eines bestehenden Elementes der Vorwärmstrecke genauer festzulegen. Bei jeder vorhan-

denen Anzapfung sind z.B. Anzapfdruck und Grädigkeit der angeschlossenen Speisewasservorwärmer innerhalb bestimmter Nebenbedingungen zu bestimmen. Bei der Spezifizierung von Grädigkeiten muß man verschiedene Fälle unterscheiden:

- Für jeden Kondensationsvorwärmer ohne integriertem Enthitzer und/oder Kondensatkühler muß die Grädigkeit festgelegt werden.
- Jeder Kondensatkühler, ob zusammen mit einem Kondensationsvorwärmer in einem Gehäuse oder separat installiert, besitzt einen Grädigkeitsparameter.
- Bei Enthitzern braucht nur dann die Grädigkeit bestimmt werden, wenn sie separat vom jeweiligen Kondensationsvorwärmer angebracht sind.
- Im Falle eines Kondensationsvorwärmers mit integriertem Enthitzer erlaubt das verwendete Kraftwerkssimulationsprogramm lediglich die Spezifizierung des Gesamtapparates (Enthitzer + Kondensationsvorwärmer). $\vartheta_{KV} = T_{Sat} - T_{SpW,ab}$ kann auch negativ werden, da $T_{SpW,ab}$ hier die Temperatur des Speisewassers nach Durchlaufen des Enthitzers ist und diese höher als T_{Sat} sein kann.

Ob nun Anzapfdruck und Grädigkeiten wählbare Parameter (Variablen) sind, hängt natürlich davon ab, ob die Komponenten, die sie beschreiben, überhaupt existieren. Bei einer maximal ausgestatteten Anzapfung $i \in \{1, \dots, \alpha\}$ (Kondensationsvorwärmer, separater Enthitzer und separater Kondensatkühler) gibt es folgende reellwertige Parameter $\tilde{c}_{i,j} := c_{(i-1)4+j}$, $i = 1, \dots, \alpha$, $j = 1, \dots, 4$ mit $c_k \in [lb_k, ub_k]$, $lb_k < ub_k$, $lb_k, ub_k \in \mathbb{R}$, $k \in \{1, \dots, 4\alpha\}$:

$\tilde{c}_{i,1} \in [lb_{(i-1)4+1}, ub_{(i-1)4+1}]$	Anzapfdruck
$\tilde{c}_{i,2} \in [lb_{(i-1)4+2}, ub_{(i-1)4+2}]$	Grädigkeit des Kondensationsvorwärmers
$\tilde{c}_{i,3} \in [lb_{(i-1)4+3}, ub_{(i-1)4+3}]$	Grädigkeit des Enthitzers
$\tilde{c}_{i,4} \in [lb_{(i-1)4+4}, ub_{(i-1)4+4}]$	Grädigkeit des Kondensatkühlers.

Alle reellwertigen Parameter werden im Vektor

$$\begin{aligned} c &= (\tilde{c}_{1,1}, \dots, \tilde{c}_{1,4}, \dots, \tilde{c}_{\alpha,1}, \dots, \tilde{c}_{\alpha,4}) \\ &= (c_1, \dots, c_4, \dots, c_{4\alpha}) \end{aligned}$$

zusammengefaßt. Er symbolisiert die technischen Eigenschaften der Komponenten der Vorwärmstrecke und ihrer Verbindungsteile, sofern diese vorhanden sind.

Der **diskret-kontinuierliche Vektor** (c, d) beschreibt die Auslegungsmöglichkeiten der **Vorwärmstrecke**. Da innerhalb der Parameter hierarchische Abhängigkeiten bestehen, codiert lediglich eine Teilmenge der innerhalb der Box-Constraints lb, ub (siehe (4.3) für die genaue Notation) möglichen Belegungen von (c, d) realisierbare Auslegungen: Mit $\tilde{d}_{i,1} = 0$ ist auch automatisch $\tilde{d}_{i,2} = \tilde{d}_{i,5} = \tilde{d}_{i,6} = 0$ für $i \in \{1, \dots, \alpha\}$ festgelegt.

Gibt es eine Komponente nicht, so können auch keine Entscheidungen zu ihrer weiteren Spezifizierung getroffen werden, wie über ihre Position, über separate oder integrierte Anbringung oder ihre technischen Eigenschaften. Sind also einige binäre Parameter auf "Nichtexistenz" gesetzt, bleibt nur ein Teil der Komponenten von (c, d) variabel oder überhaupt verfügbar. Parameter, auf die man gerade gar keinen Zugriff hat, kann man auf irgendeinen Wert, z.B. null, setzen - sie beeinflussen die Simulation des aktuellen Kraftwerkes sowieso in keiner Weise.

4.1.2.3 Darstellung der (variablen) Parameter in einem Entscheidungsbaum

Durch die Darstellung aller Parameter in einem **Entscheidungsbaum** (siehe Abbildung 4.3) kann man die bereits erwähnten hierarchischen Abhängigkeiten bei dem betrachteten Auslegungsproblem bildhaft verdeutlichen.

Jedem Parameter entspricht ein Knoten im Baum. Die Verbindung eines Knotens mit einem Knoten weiter oben im Baum symbolisiert seine Abhängigkeit vom oberen Knoten, d.h. es besteht nur Zugriff auf den zugehörigen Parameter, wenn der Parameter des oberen Knotens einen bestimmten Wert annimmt. Jeder mögliche Wert für einen Parameter wird durch je eine Kante dargestellt, die vom entsprechenden Knoten zu einem weiter unten liegenden Knoten verläuft. Da von reellen Parametern keine weiteren Parameter/Entscheidungen abhängen, werden diese jeweils auf Blätter des Entscheidungsbaumes abgebildet.

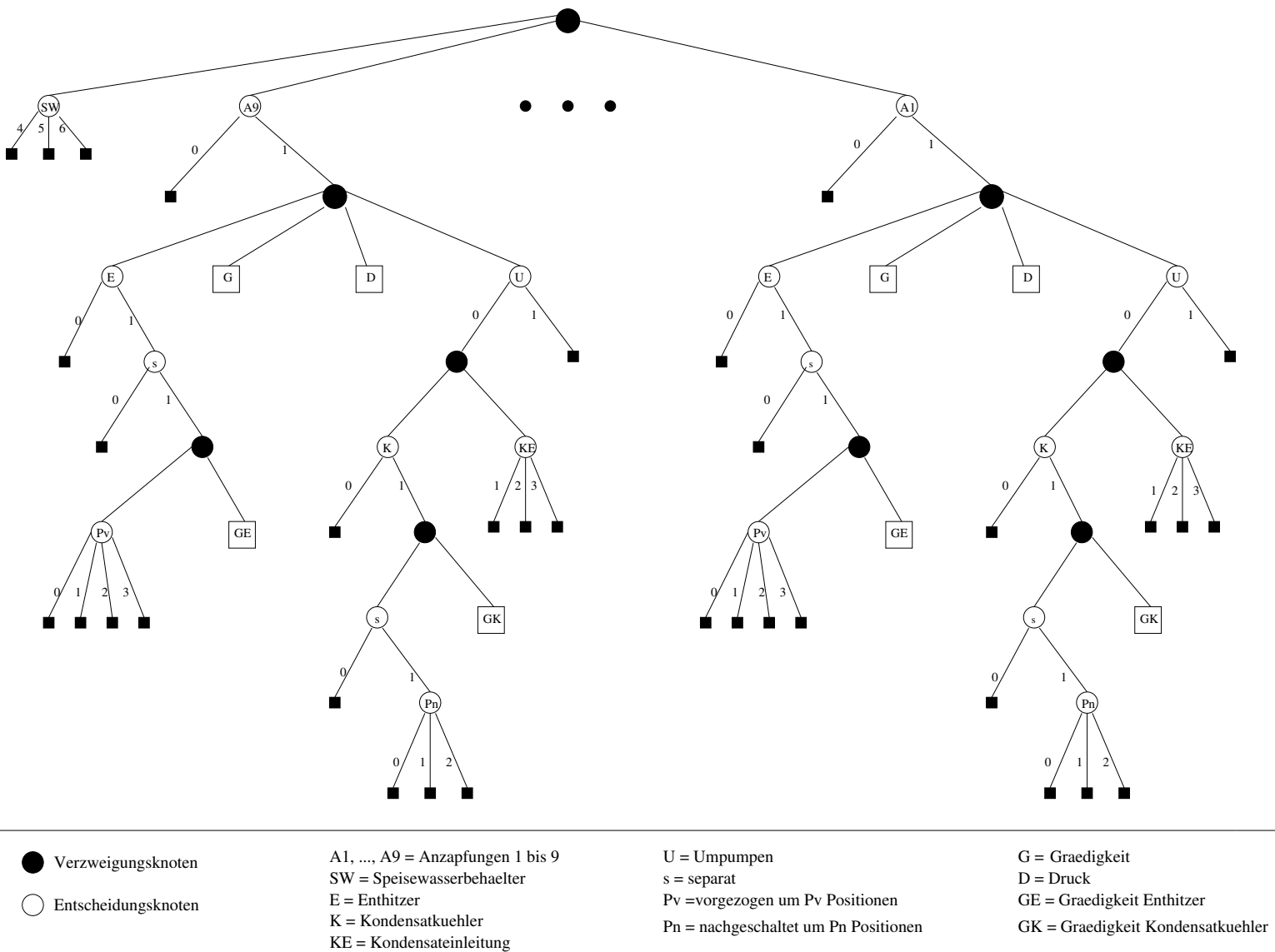
Außer Knoten, die Parameter repräsentieren, werden im Baum noch weitere Knoten eingebaut, die symbolisieren, daß mehrere voneinander unabhängige Entscheidungen zu treffen sind - sie sind Verzweigungen im Baum.

Knoten, die Parameter abbilden, nennen wir **Entscheidungsknoten**, die anderen **Verzweigungsknoten**.

Sind einige Entscheidungen bezüglich einer bestimmten Auslegung bereits getroffen, so kann man herausfinden, welche Parameter noch variierbar sind, indem man den Entscheidungsbaum von der Wurzel oben bis zu allen Blättern unten nach folgenden Regeln durchläuft:

- Von einem Verzweigungsknoten aus sind nacheinander alle Kanten zu verfolgen.
- Ausgehend von einem Entscheidungsknoten hat man nur der Kante zu folgen, die den momentanen Wert des repräsentierten Parameters verkörpert.
- Sobald bei einem Blatt angekommen, muß man zum nächsthöheren Verzweigungsknoten zurückspringen.

Abbildung 4.3: Darstellung der Parameter der Vorwärmstrecke in einem Entscheidungsbaum



4.1.2.4 Die Ziele

Typischerweise ist es das Ziel der Auslegungsoptimierung eines Kraftwerkes, die **spezifischen Stromerzeugungskosten**, d.h. die Kosten, die dem Kraftwerkbetreiber für die Herstellung einer kWh Strom entstehen, zu minimieren. Die spezifischen Stromerzeugungskosten spalten sich in die Blöcke **Investitionskosten** und **Betriebskosten** auf, wobei beim Letzteren lediglich die **Brennstoffkosten** berücksichtigt werden sollen:

$$K_{Strom} := \underbrace{\frac{K_{Invest} \cdot a}{P_{el} \cdot h}}_{\text{Investitionsanteil}} + \underbrace{\frac{K_{Brennstoff}}{\eta}}_{\text{Brennstoffanteil}}, \quad (4.2)$$

K_{Invest}	=	Investitionskosten ,
a	=	Annuitätenfaktor,
P_{el}	=	elektrische Leistung,
h	=	jährliche Betriebsstunden,
$K_{Brennstoff}$	=	Brennstoffkosten pro kWh Primärenergie,
η	=	Gesamtwirkungsgrad ,

wobei a , h und $K_{Brennstoff}$ vorgegebene Konstanten sind, wohingegen K_{Invest} , P_{el} und η in einem Kraftwerkssimulationsprogramm berechnet werden.

a und $K_{Brennstoff}$ können der Auslegungsoptimierung Schwierigkeiten bereiten:

- a wird durch den herrschenden Marktzins beeinflusst. Daher schwankt diese Größe zeitlich wie geographisch sehr stark.
- Auch $K_{Brennstoff}$ verändert sich mit Zeit und Ort.

Die Werte beider Parameter sind demnach schwer vorhersagbar, was eine gute Prognose der Größe K_{Strom} für Kunden des Kraftwerkherstellers erschwert. Auf eine zeitlich und geographisch global optimale Anlagenauslegung zu hoffen ist daher unrealistisch. Höchstens regional maßgeschneiderte Lösungen sind möglich.

K_{Invest} und η sind die Größen in Gleichung (4.2), die man durch Variieren der Auslegungsparameter, die in den Abschnitten 4.1.2.1 und 4.1.2.2 vorgestellt wurden, optimal einstellen kann und die obendrein problemlos berechenbar sind. η steht für die Qualität der Leistung des Kraftwerkes und K_{Invest} für den Preis, den die Anlagenauslegung, die diesen Leistungsstandard ermöglicht, kostet. Der Investitionsanteil in Gleichung (4.2) sinkt bei fallendem K_{Invest} , und der Brennstoffanteil sinkt bei steigendem η . Somit erwachsen aus dem Ziel, die spezifischen Stromerzeugungskosten zu minimieren, **zwei Ziele**, nämlich gleichzeitig die **Investitionskosten zu minimieren** und den **Wirkungsgrad des Kraftwerkes zu maximieren**. Diese beiden Ziele werden nicht von den problematischen Größen a und $K_{Brennstoff}$ beeinflusst. Es ist wohl keine Überraschung, daß sich die beiden genannten Ziele widersprechen. Der Zielkonflikt wird bereits deutlich, wenn man beispielsweise die Reaktion von Investitionskosten, des Brennstoffanteils und somit auch der spezifischen Stromerzeugungskosten auf das Herumspielen mit der Grädigkeit eines Kondensationsvorwärmers mit integriertem Enthitzer betrachtet:

Abbildung 4.4: Zielkonflikt

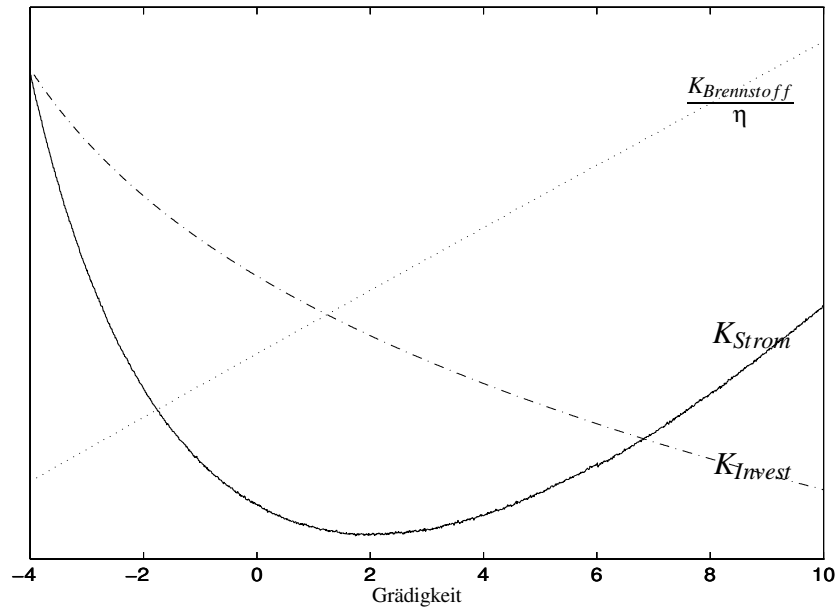


Abbildung 4.4 zeigt den Zielkonflikt. Die Investitionskosten hängen von der Heizfläche des Vorwärmers ab, die mit wachsender Grädigkeit sinkt. Je höher jedoch die Grädigkeit ist, desto höher ist auch der Wert von $\frac{K_{Brennstoff}}{\eta}$, da η dann sinkt.

Interpretiert man den in Abbildung 4.4 dargestellten speziellen Ausschnitt des Zielraumes vor dem Hintergrund der Optimalität in der Vektoroptimierung, so erkennt man, daß es sich dabei um die Abbildung von Zielfunktionswerten (lokal) paretooptimaler Alternativen für die Belegung eines Grädigkeitsparameters handelt: Verbesserungen in einem Ziel müssen stets mit Verschlechterungen im anderen Ziel bezahlt werden - es gibt in dem Ausschnitt des Alternativenraumes, auf den wir von dem abgebildeten Ausschnitt des Zielraumes schließen, zu keiner Alternative eine diese dominierende Konkurrenzalternative.

Die skalare Zielfunktion K_{Strom} hat ihr Minimum bei etwa 2 ($K = \text{Kelvin}$). Bei Grädigkeitswerten $< 2K$ beeinflussen vor allem die Investitionskosten die Höhe der spezifischen Stromerzeugungskosten, bei Grädigkeitswerten $> 2K$ ist η als Regelgröße des Brennstoffanteils der bedeutendere Einflußfaktor. Abgesehen von der Tatsache, daß K_{Strom} aufgrund des problematischen Einflusses von a und $K_{Brennstoff}$ nicht korrekt zu errechnen und daher zu optimieren ist, stellt die Optimierung der skalaren Zielfunktion K_{Strom} ohnehin eine Art Notlösung dar: Ein diskret-kontinuierliches Mehrzielproblem wird mit einem skalaren Ersatzproblem umgangen. Der Mehrzielansatz macht alle Trade-Offs zwischen den verschiedenen Zielen sichtbar und liefert eine repräsentative Teilmenge aller sinnvollen Parameterbelegungen (= paretooptimale Handlungsalternativen). Dem Entscheidungsträger wird auf diese Weise der bestmögliche Einblick in das Anlagenauslegungsproblem gewährt. Löst man hingegen das skalare Minimierungsproblem, wird der Entscheidungsträger lediglich auf einige wenige sinnvolle Handlungsalternativen aufmerksam gemacht, da jedem Ziel bereits eine bestimmte Priorität zugewiesen ist, die nicht unbedingt passend sein muß. Vielleicht würde der Entscheidungsträger seine Wünsche bezüglich der Kriterienerfüllung noch etwas verlagern wollen: Eventuell wäre er z.B. bereit, für ein höheres η etwas mehr Geld auszugeben,

würde also ein höheres K_{Invest} und η bevorzugen, als die optimalen Lösungen des skalaren Ersatzproblems erlauben, kann aber entsprechende Handlungsalternativen mit dem Einzielan-satz nicht finden, weil sie nicht zu den globalen Minimierern der skalaren Zielfunktion gehören.

Ein weiteres interessantes Zweiziel-Szenario ist, das Verhalten von K_{Strom} und η simul-tan zu betrachten. Die Art, in der K_{Invest} und η in die Berechnung des skalaren Zieles K_{Strom} eingehen, könnte den Einfluß von K_{Invest} auf K_{Strom} derartig überbetonen, daß eine Erhöhung von η , die ja auch K_{Invest} erhöht, dafür aber den Brennstoffanteil von K_{Strom} senkt, K_{Strom} insgesamt doch steigen lassen würde. Umgekehrt betrachtet könnte daher die Minimierung von K_{Invest} zu Lösungsalternativen führen, die lediglich ein unakzeptabel niedriges η zulassen. Bei dem Zweizielansatz K_{Strom} gegen η wird dem Entscheidungsträger hingegen ermöglicht, aus sinnvollen Lösungsalternativen zu wählen, die gute Werte für η bei gleichzeitig möglichst niedrigen Werten für K_{Strom} liefern. Allerdings gehen mit der Zielfunktion K_{Strom} auch wieder die problematischen Faktoren a und $K_{Brennstoff}$ in Kalkulation und Optimierung ein.

4.1.2.5 Die Nebenbedingungen

Als der diskret-kontinuierliche Vektor (c, d) als Codierung der variablen Parameter bei der Aus-legungsoptimierung eines Kraftwerkes eingeführt wurde, erschienen als erste Nebenbedingun-gen an den zulässigen Wertebereich dieser Parameter Box-Constraints. Bei maximal $\alpha \in \mathbb{N}$ möglichen Anzapfungen ist die Dimension von (c, d) gemäß der Abschnitte 4.1.2.1 und 4.1.2.2 $m = 1 + 9\alpha + 4\alpha$, wobei $m_1 = 4\alpha$ die Dimension von c und $m_2 = 1 + 9\alpha$ die Dimension von d ist. Wie bereits am Ende von Abschnitt 4.1.2.2 erwähnt, schwankt aufgrund hierarchischer Abhängigkeiten unter den Parametern die Dimension des Teilvektors aus (c, d) , der die aktu-ellen variablen Parameter umfaßt, mit speziellen Wertebelegungen bestimmter Parameter. Exi-stiert eine Anzapfung nicht, entfallen natürlich auch die Variationsmöglichkeiten der Parameter, die diese Anzapfung näher bestimmen. Trotzdem werden für alle Komponenten von (c, d) Box-Constraints vorgegeben, ob die entsprechenden Parameter nun zur Verfügung stehen oder nicht:

$$\begin{aligned}
 lb := \begin{pmatrix} lb_1 \\ lb_2 \\ \dots \\ lb_{m_1} \\ lb_{m_1+1} \\ lb_{m_1+2} \\ \dots \\ lb_m \end{pmatrix} &\leq \begin{pmatrix} c_1 \\ c_2 \\ \dots \\ c_{m_1} \\ d_0 \\ d_1 \\ \dots \\ d_{m_2-1} \end{pmatrix} \leq \begin{pmatrix} ub_1 \\ ub_2 \\ \dots \\ ub_{m_1} \\ ub_{m_1+1} \\ ub_{m_1+2} \\ \dots \\ ub_m \end{pmatrix} := ub, \\
 lb &\leq ub, \\
 lb_i, ub_i &\in \mathbb{R}, i = 1, \dots, m_1, \\
 lb_i, ub_i &\in \mathbb{Z}, i = m_1 + 1, \dots, m.
 \end{aligned} \tag{4.3}$$

Zu den Box-Constraints kommen bei den reellwertigen Parametern (kontinuierliche Variablen) c noch dynamische Nebenbedingungen hinzu.

Nebenbedingungen an den Anzapfdruck:

Die Nummer einer Anzapfung ist umso niedriger, je später die Anzapfung entlang des Turbinenflusses, entlang dessen ja der Druck des Dampfes sinkt, angebracht ist. Es sei $\mathcal{E} := \{e_1, \dots, e_l\} \subseteq \{1, \dots, \alpha\}, l \in \mathbb{N}$, die Menge aller vorhandenen Anzapfungen, d.h. $\tilde{d}_{e_i,1} = d_{g(e_i-1)+1} = 1$ für $i = 1, \dots, l$. Entsprechend der erwähnten Abnahme des Dampfdruckes muß für je zwei aufeinanderfolgende Anzapfungen $e_i, e_{i+1}, i \in \{1, \dots, l-1\}$, gelten:

$$g_i^{\hat{d}}(c, \hat{d}) := \tilde{c}_{e_i,1} - \tilde{c}_{e_{i+1},1} \leq 0, \hat{d} \in \begin{bmatrix} lb_{m_1+1}, ub_{m_1+1} \\ \dots \\ lb_m, ub_m \end{bmatrix}. \quad (4.4)$$

Das Hütchen auf \hat{d} soll auf die Tatsache hinweisen, daß obige Ungleichung für eine bestimmte Belegung \hat{d} von d mit $\hat{d}_{g(e_i-1)+1} = 1 \quad \forall i = 1, \dots, l$ definiert ist. Für ein $i \in \{1, \dots, \alpha\} \setminus \mathcal{E}$ existiert (4.4) einfach nicht, da $\tilde{c}_{i,1}$ nicht zur Verfügung steht. Die Nebenbedingungen (4.4) wären nicht nötig, wenn die Box-Constraints so gewählt würden, daß sie entsprechende disjunkte Mengen bildeten. Daraus resultierte jedoch eine derart drastische Einschränkung des zulässigen Bereiches für den Anzapfdruck, daß eventuell auch paretooptimale Parameterbelegungen ausgeschlossen würden.

Nebenbedingungen an die Grädigkeiten:

Theoretisch müssen Grädigkeiten stets positiv sein, in dieser Anwendung gibt man sogar eine Unterschranke von $\vartheta_{min} := 1.5K$ vor, da niedrigere Grädigkeiten bei den Vorwärmern zu unsinnig großen Heizflächen führen würden. Diesen Vorgaben kann fast immer durch die relevanten Box-Constraints zur Gültigkeit verholfen werden, Kondensationsvorwärmer mit integriertem Enthitzer bilden jedoch eine Ausnahme. Bei diesen darf die Grädigkeit des Gesamtapparates negativ werden, die Grädigkeit des Kondensationsvorwärmers muß jedoch positiv bleiben, kann aber nicht direkt gesteuert werden, da sie nicht zu den kontinuierlichen Variablen gehört.

Sei $\mathcal{F} := \{f_1, \dots, f_k\} \subseteq \{e_1, \dots, e_l\} \subseteq \{1, \dots, \alpha\}, k \in \mathbb{N}$, die Menge der vorhandenen Turbinenanzapfungen, an die ein Kondensationsvorwärmer mit integriertem Enthitzer angeschlossen ist, also $\tilde{d}_{f_i,1} = \tilde{d}_{f_i,2} = 1, \tilde{d}_{f_i,3} = 0 \quad \forall i \in \{1, \dots, k\}$. Dann muß für alle $i \in \{1, \dots, k\}$ die Nebenbedingung

$$\begin{aligned} g_j^{\hat{d}}(c, \hat{d}) &:= T_{SpW,ab}(CH; f_i) + \vartheta_{min} - T_{Sat}(f_i) \leq 0, \\ \hat{d} &\in \begin{bmatrix} lb_{m_1+1}, ub_{m_1+1} \\ \dots \\ lb_m, ub_m \end{bmatrix}, \\ \hat{d}_{g(f_i-1)+1} &= \hat{d}_{g(f_i-1)+2} = 1, \hat{d}_{g(f_i-1)+3} = 0, \\ j &= i + l - 1, \end{aligned} \quad (4.5)$$

gelten. Wieder soll das Hütchen auf \hat{d} darauf aufmerksam machen, daß obige Ungleichung für eine bestimmte Wertebelegung \hat{d} von d konstruiert ist. $T_{SpW,ab}(CH; f_i)$ und $T_{Sat}(f_i)$ werden für alle $i \in \{1, \dots, k\}$ vom Kraftwerkssimulationsprogramm berechnet, und l ist die Anzahl der

vorhandenen Anzapfungen. Die Anzahl $l - 1$ rührt von den bereits betrachteten Ungleichungsnebenbedingungen (4.4) her.

Die Gesamtzahl $p^{\hat{d}} := l + k - 1$ von Ungleichungsnebenbedingungen hängt von der jeweiligen Wertebelegung \hat{d} von d ab.

Sind Nebenbedingungen an die Grädigkeiten verletzt, so vergibt das Kraftwerkssimulationsprogramm künstliche Zielfunktionswerte, die den Optimierungsvorgang nicht stören. Gibt es jedoch verletzte Nebenbedingungen an den Anzapfdruck, so wird die Simulation des Kraftwerkes mit einer Fehlermeldung gestoppt.

4.1.2.6 Formulierung des Optimierungsproblems

Wie kann man das Problem, die Vorwärmstrecke eines Dampfkraftwerkes optimal auszulegen, als Mehrzieloptimierungsproblem der Form (3.2) formulieren, so daß es mit Hilfe des in Kapitel 3 entwickelten diskret-kontinuierlichen Vektoroptimierungsverfahrens gelöst werden kann? Typisch für praktische Anwendungsprobleme paßt diese Anwendung nicht hundertprozentig zu dem in Kapitel 3 erstellten Modell. Sie ergibt ein schönes Beispiel für die Schwierigkeiten, einen Algorithmus, der unter gewissen mathematischen Annahmen entstand, auf ein Problem der Praxis anzuwenden.

- Beim Vorwärmstreckenproblem besteht D nicht aus der gesamten Menge $[lb_D, ub_D]$ mit $lb_D := (lb_{m_1+1}, \dots, lb_m)^T, ub_D := (ub_{m_1+1}, \dots, ub_m)^T$ (siehe (4.3)), sondern lediglich aus einer Teilmenge davon, nämlich den Wertebelegungen innerhalb der Box-Constraints, die tatsächlich realisierbare Strukturvariationen darstellen. Stehen einige der m_2 diskreten Parameter (Komponenten von d) nicht zur Verfügung, da die Vorwärmstreckenelemente, die sie repräsentieren, nicht existieren, bekommen sie einfach einen bestimmten Wert innerhalb der Box-Constraints oder den Wert 0 zugewiesen - dieser Wert wird die Berechnungen nicht beeinflussen. Somit ergibt sich eine Menge $D \subseteq \mathbb{Z}^{m_2} \subset \mathbb{R}^{m_2}$, deren Elemente in Übereinstimmung mit der Notation in Abschnitt 3.1 x_D heißen sollen.

Beispiel 4.1.1. $d_0 = 4, d_1 = 1, d_2 = 0, d_5 = 1, d_{9k+1} = 0, k = 1, \dots, \alpha - 1$, ergibt den Vektor $x_D := (4, 1, 0, \dots, 0, 1, 0, \dots, 0) \in \mathbb{R}^{m_2}$. Es gibt also nur zwei Anzapfungen: Den Speisewasserbehälter (Anzapfung 4) ohne nähere Bestimmung und Anzapfung 1 mit lediglich einer Kondensatpumpe. Alle weiteren binären Parameter werden auf den Wert 0 gesetzt, nicht verfügbare Parameter erhalten auch den Wert 0.

△

- In Abhängigkeit von der aktuellen Belegung \bar{x}_D von d bleiben nur $m_{\bar{x}_D} \leq m_1, m_{\bar{x}_D} \in \mathbb{N}_0$, der m_1 Komponenten von c variabel. Ihre Werte sind aus der Menge $[lb_{C^{\bar{x}_D}}, ub_{C^{\bar{x}_D}}] \subseteq \mathbb{R}^{m_{\bar{x}_D}}$ zu wählen. Die Vektoren $lb_{C^{\bar{x}_D}}, ub_{C^{\bar{x}_D}}$ bestehen aus den entsprechenden Komponenten von lb, ub (siehe (4.3)). Die von den Box-Constraints umrissene Menge wird zusätzlich durch die in (4.4) und (4.5) definierten Nebenbedingungen $g_i^{\bar{x}_D}(c, \bar{x}_D) \leq 0 \quad \forall i = 1, \dots, p^{\bar{x}_D}, \hat{d} = \bar{x}_D$, eingeschränkt. Die so definierte Menge heißt $C^{\bar{x}_D}$. Obwohl die Elemente von $C := \cup_{x_D \in D} C^{x_D}$ nun mit x_C bezeichnet werden, stimmt die Definition von C nicht exakt mit der von C in (3.1) überein - die verschiedenen x_C sind verschieden lange Vektoren.

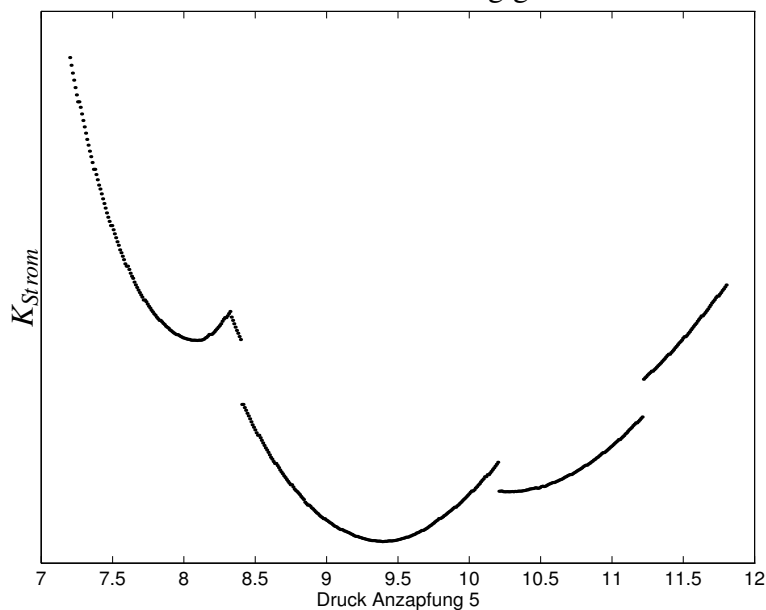
Beispiel 4.1.2. In Beispiel 4.1.1 sind lediglich zwei Komponenten von c variabel. Zu der speziellen Belegung \bar{x}_D von x_D gehört die Menge $C^{\bar{x}_D}$, die aus allen

$$x_C := (c_1, c_2) = (\tilde{c}_{1,1}, \tilde{c}_{1,2}), c_1 \in [lb_1, ub_1], c_2 \in [lb_2, ub_2],$$

besteht, die obendrein $g_1^{\bar{x}_D}(c, \bar{x}_D)$ aus (4.5) mit $i = 1, l = 1$ erfüllen müssen. \triangle

- Die problemspezifische Menge R hat mit $R := \cup_{x_D \in D} \{x_D\} \times C^{x_D}$ nicht ganz die Form $R := C \times D$ gemäß (3.1). Ihre Elemente zerfallen jedoch auch in kontinuierliche und diskrete Komponenten, so daß die Notation $x := (x_C, x_D)$ ihre Berechtigung behält, wobei $x_D \in D$ und $x_C \in C^{x_D} \subseteq C$ gilt. Obwohl die Mengen R aus Modell und Anwendung nicht völlig zusammenpassen, kann das Vorwärmstreckenproblem von dem stochastischen diskret-kontinuierlichen Mehrzieloptimierungsverfahren, das in Kapitel 3 entwickelt wurde, mit DKMOP SS und/oder DKMOP PS gelöst werden.
- Die beiden Zielfunktionen sind $f_1(x) := K_{Invest}(x)$ bzw. $f_1(x) := K_{Strom}(x)$ und $f_2(x) := -\eta(x)$. Sie analytisch niederschreiben wäre zu kompliziert. Das Programm, das das Kraftwerk simuliert, liefert zu jedem $x \in R$ die Zielfunktionswerte. Simulationen haben gezeigt, daß $f_i^{\bar{x}_D} : C^{\bar{x}_D} \mapsto \mathbb{R}, x_C \mapsto f_i(x_C, \bar{x}_D)$ nicht zweimal stetig differenzierbar sein muß für alle $i = 1, 2, \bar{x}_D \in D$. Abbildung 4.5 zeigt den Graph der skalarisierten Zielfunktion K_{Strom} bei Variation eines einzigen Anzapfungsparameters. An einigen Stellen ist diese Zielfunktion weder differenzierbar noch stetig. Die sichtbaren Sprünge entstehen, weil die Turbinenanzapfung nicht an jeder beliebigen Stelle der Turbine, sondern nur zwischen den Schaufelreihen angebracht werden kann. Falls für ein bestimmtes gewähltes Druckniveau die Anzapfung in Höhe einer Schaufelreihe liegen müßte, paßt das verwendete Simulationsprogramm die Lage der Schaufelreihen an - der Zielwert wird bei veränderter Kraftwerkskonstruktion simuliert und daraus resultieren Sprünge im Zielwertverlauf.

Abbildung 4.5: Skalarisierte Zielfunktion in Abhängigkeit eines freien Druckparameters



Offensichtlich ist das gegebene Auslegungsproblem nicht genau wie Problem (3.2) formulierbar. In den Problemen (3.2), (3.10) und (3.20) muß

$$\begin{aligned} R &:= C \times D & \text{durch} & & R &:= \cup_{x_D \in D} \{x_D\} \times C^{x_D}, \\ C &\in \mathbb{R}^{m_1} & \text{durch} & & C &:= \cup_{x_D} C^{x_D}, \\ f_i^{x_D} &: C \mapsto \mathbb{R} & \text{durch} & & f_i^{x_D} &: C^{x_D} \mapsto \mathbb{R} \quad \forall i = 1, \dots, n \end{aligned} \quad (4.6)$$

ersetzt werden. Die Definitionen von $E^{\bar{x}_D} := \{(x_C, \bar{x}_D) | x_C \in L^{\bar{x}_D}\} \subseteq R, \bar{x}_D \in D$, in (3.17) und daher von R_D bleiben unberührt.

Trotzdem ist der diskret-kontinuierliche Mehrzieloptimierer aus Kapitel 3 anwendbar. Es ist vor allem wichtig, daß sämtliche kontinuierlichen Subprobleme lösbar sind.

Die kontinuierlichen Subprobleme

Zu jedem festen $\bar{x}_D \in D$ gehört eine Menge "kontinuierlicher Alternativen" (d.h. die Menge der möglichen Kombinationen aus Belegungen kontinuierlicher Parameter). Es handelt sich dabei um eine Teilmenge $C^{\bar{x}_D}$ aus $\mathbb{R}^{m_{\bar{x}_D}}$, die neben den $m_{\bar{x}_D} \in \mathbb{N}$ relevanten Box-Constraints aus lb und ub (siehe (4.3)) zusätzlich durch die Ungleichungsnebenbedingungen $g_i^{\bar{x}_D}(c, \bar{x}_D) \leq 0 \quad \forall i = 1, \dots, p^{\bar{x}_D}$ (siehe (4.4) und (4.5) mit $\hat{d} = \bar{x}_D$) eingeschränkt ist. Falls diese $g_i^{\bar{x}_D}(c, \bar{x}_D)$ für alle $i = 1, \dots, p^{\bar{x}_D}$ stetig sind, so ist $C^{\bar{x}_D}$ kompakt und das Problem

$$\text{"min"}_{x_C \in C^{\bar{x}_D}} \begin{pmatrix} f_1^{\bar{x}_D}(x_C) \\ f_2^{\bar{x}_D}(x_C) \end{pmatrix} \quad (4.7)$$

besitzt sicherlich effiziente Lösungen, falls außerdem die beiden Zielfunktionen stetig sind. Die den Anzapfdruck betreffenden Ungleichungsnebenbedingungen (4.4) sind in der Tat stetig, die Grädigkeitsnebenbedingungen (4.5) spielen eigentlich keine Rolle; sind sie verletzt, so vergibt das Kraftwerkssimulationsprogramm irgendwelche Zielfunktionswerte, die sicher nicht besser als oder genauso gut wie effiziente Zielwerte sind. Mit Hilfe einiger Tricks kann ein kontinuierliches Mehrzielverfahren, das Gradienteninformation benötigt, mit den gegebenen Zielfunktionen trotz ihrer eventuellen Knicke und Sprünge im Zielwertverlauf arbeiten. Die technische Interpretation dieser nichtdifferenzierbaren und/oder unstetigen Stellen erlaubt es ohnehin auszuschließen, daß hier paretooptimale Auslegungsalternativen liegen.

Problem (4.7) ist also für jedes $\bar{x}_D \in D$ mit einem vorhandenen kontinuierlichen Vektoroptimierungsverfahren lösbar.

4.1.3 Schwierigkeiten bei der Anpassung des entwickelten Verfahrens an das gegebene Anwendungsproblem

Zur Lösung der Probleme (3.21) für die verschiedenen Referenzalternativen $\bar{x} = x_k \in R_D, k \in \mathbb{N}_0$, wird der Metropolis Algorithmus zu Hilfe genommen. Gemäß Satz 3.2.22 und wie in Abschnitt 3.2.5.4 dargelegt muß die Markov Kette $\{X_t\}_{t \in \mathbb{N}_0}$, die einen speziellen Metropolis Algorithmus (IMA) definiert, von dem eine numerische Version ein Problem (3.21) bearbeitet, einige Eigenschaften haben: Aperiodizität und Irreduzibilität und zum Beweis von Satz 3.2.22 die Symmetrie der Generierungswahrscheinlichkeiten G_{ij} bei allen Alternativenpaaren i und j aus S_k . Diese Eigenschaften sind in Abschnitt 3.2.5.3 näher beschrieben. Die Suchräume $S_k = \{x_k\} \times R_k \subseteq \{x_k\} \times (R_D \setminus \{x_k\})$ der Metropolis-Läufe können entsprechend der in Abschnitt 3.2.7 erklärten Möglichkeiten, den Suchraum mehr und mehr einzuschränken, genau

bestimmt werden. Da ja einige Auslegungsparameter hierarchisch voneinander abhängen, ist gerade die geforderte Symmetrie der Generierungswahrscheinlichkeiten nicht so einfach zu erreichen.

Was ist ein Nachbar (x_k, z) des momentanen Standpunktes $(x_k, y) \in S_k$? Da der erste Partner jedes Punktes aus S_k (jeder ein Paar zweier Alternativen aus R_D) in Form der Referenzalternative $\bar{x} = x_k$, für die der Metropolis Algorithmus (NMA) eine dominierende Alternative suchen soll, festgelegt ist, verbleibt nur noch, den zweiten Partner aus R_k zu wählen. Da R_D die Vereinigung $\cup_{x_D \in D} E^{x_D}$ aus den Effizienzmengen in $\{x_D\} \times C^{x_D}$ zu allen diskreten Parameterbelegungen $x_D \in D$ ist, sollte ein Zufallsexperiment zunächst eine bestimmte diskrete Variation z_D aus einer Teilmenge $N_D^{(x_k, y)}$ von $\{x_D | (x_C, x_D) \in R_k \text{ für ein } x_C \in C^{x_D}\}$ bestimmen. Die **diskrete Zufallsvariable** $X_{prop}^D(x_k, y)$ bezeichne dieses Zufallsexperiment. Auf diese Weise ist eine Menge E^{z_D} kontinuierlich effizienter Punkte anvisiert, aus der wiederum zufällig ein Element $z = (z_C, z_D)$ gewählt wird. Dieses weitere Zufallsexperiment werde von der **diskreten Zufallsvariable** $X_{prop}^{z_D}(x_k, y)$ symbolisiert. Es ist nun die Absicht, die Nachbarschaft $N_{(x_k, y)} \subseteq S_k$ von (x_k, y) aus allen Paaren $(x_k, z) \in S_k \setminus \{(x_k, y)\}$ bestehen zu lassen, da dies $N_D^{(x_k, y)} = \{x_D | (x_C, x_D) \in R_k \text{ für ein } x_C \in C^{x_D}\}$ erlaubt, $N_D^{(x_k, y)}$ dann also aus allen zulässigen diskreten Variationsmöglichkeiten innerhalb von R_k besteht.

Als Hintereinanderausführung zweier Zufallsexperimente haben die Generierungswahrscheinlichkeiten $G_{(x_k, y)(x_k, z)}$ (hier $i = (x_k, y), j = (x_k, z)$) für alle $(x_k, y), (x_k, z) \in S_k$ die Form

$$G_{(x_k, y)(x_k, z)} := P\{X_{prop}^D(x_k, y) = z_D\}P\{X_{prop}^{z_D}(x_k, y) = (z_C, z_D)\}. \quad (4.8)$$

Wie erhält man

$$\begin{aligned} G_{(x_k, y)(x_k, z)} &= P\{X_{prop}^D(x_k, y) = z_D\}P\{X_{prop}^{z_D}(x_k, y) = (z_C, z_D)\} = \\ &P\{X_{prop}^D(x_k, z) = y_D\}P\{X_{prop}^{y_D}(x_k, z) = (y_C, y_D)\} = G_{(x_k, z)(x_k, y)}? \end{aligned} \quad (4.9)$$

Die Wahrscheinlichkeitsverteilung von $X_{prop}^{z_D}(x_k, y), z_D \in N_D^{(x_k, y)}, y \in R_k$, hängt hauptsächlich von $|\bar{E}^{z_D}|$ (\bar{E}^{z_D} die numerische Approximation von E^{z_D}) ab, und es ist nicht sinnvoll, sie von der Wahrscheinlichkeitsverteilung von $X_{prop}^D(x_k, y)$ und von den Größen $|\bar{E}^{x_D}|$ anderer diskreter Variationen $x_D \in D \setminus \{y_D, z_D\}$ beeinflussen zu lassen; Letzteres wäre gar nicht möglich, da so ein \bar{E}^{x_D} im Suchlauf eventuell noch nicht errechnet wurde und die Größe $|\bar{E}^{x_D}|$ daher noch unbekannt ist. Folglich ist die in Gleichung (4.9) geforderte Symmetrie nur erreichbar, wenn man

$$P\{X_{prop}^D(x_k, y) = z_D\} = P\{X_{prop}^D(x_k, z) = y_D\} \quad (4.10)$$

und gleichzeitig

$$P\{X_{prop}^{z_D}(x_k, y) = (z_C, z_D)\} = P\{X_{prop}^{y_D}(x_k, z) = (y_C, y_D)\} \quad (4.11)$$

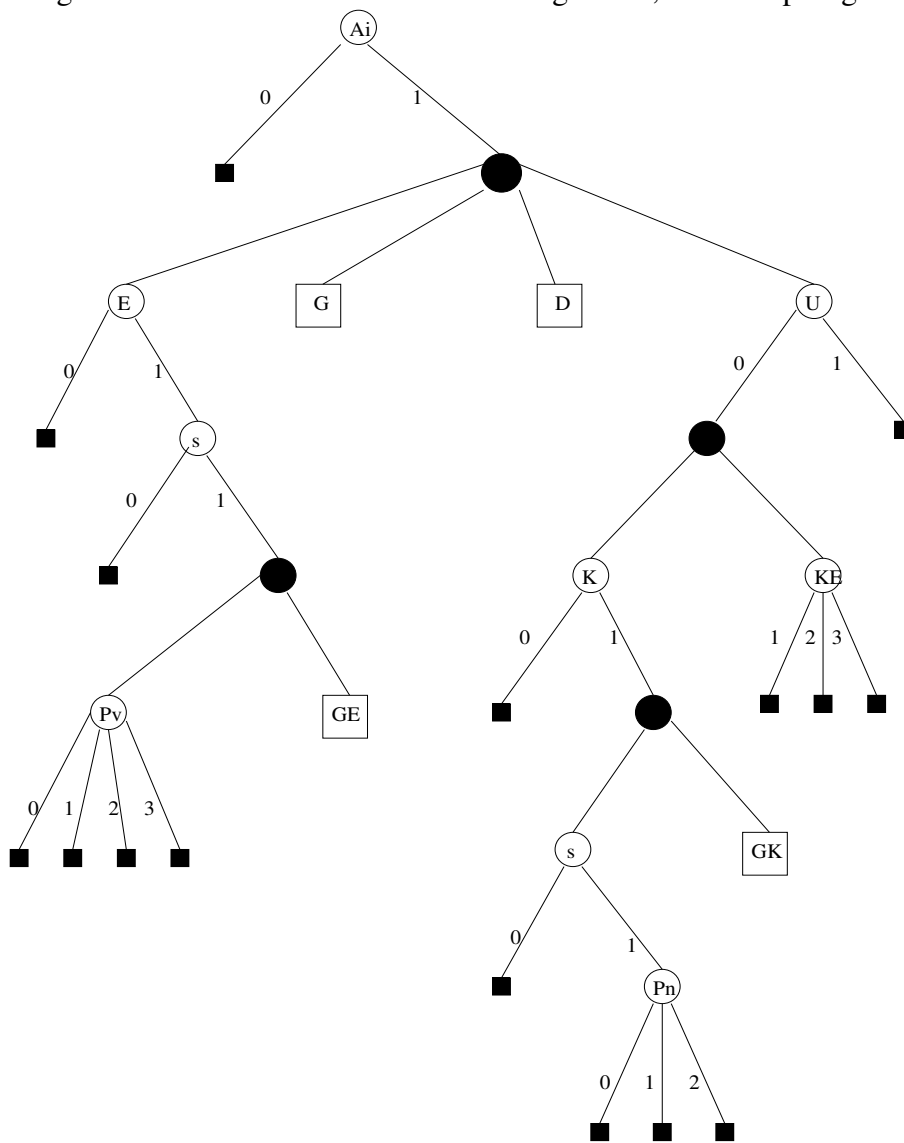
konstruiert.

Für die Gültigkeit von (4.10) sind die hierarchischen Abhängigkeiten unter einzelnen diskreten Parametern ein ernsthaftes Hindernis. Um dies zu verstehen, betrachte man den Teil \tilde{d}_i einer Belegung von d , der eine bestimmte Anzapfung $i \in \{1, \dots, \alpha\}$ codiert, wobei α wieder die maximale Anzahl vorhandener Anzapfungen sei:

$$\tilde{d}_i := (\tilde{d}_{i,1}, \tilde{d}_{i,2}, \dots, \tilde{d}_{i,9}). \tag{4.12}$$

Die Bedeutung der einzelnen Komponenten wurde bereits in Abschnitt 4.1.2.1 erklärt. Eine Anzapfung herausgelöst aus der gesamten Vorwärmstrecke zu betrachten ist möglich, da die Entscheidungen über ihre Existenz und nähere Spezifizierung unabhängig von Existenz und Design aller übrigen Anzapfungen treffbar sind.

Abbildung 4.6: Ausschnitt aus dem Entscheidungsbaum, der Anzapfung i darstellt



$\tilde{d}_{i,4}$ (relative Enthitzerposition) kann nur für einen vorhandenen ($\tilde{d}_{i,2} = 1$) und separat installierten ($\tilde{d}_{i,3} = 1$) Enthitzer gewählt werden, $\tilde{d}_{i,3}$ nur bei $\tilde{d}_{i,2} = 1$. $\tilde{d}_{i,8}$ (relative Position des Kondensatkühlers) ist nur variierbar, falls ein separat installierter ($\tilde{d}_{i,7} = 1$) Kondensatkühler existiert ($\tilde{d}_{i,6} = 1$), $\tilde{d}_{i,7}$ nur bei $\tilde{d}_{i,6} = 1$ und $\tilde{d}_{i,9}$ (Spezifizierung der Kondensateinleitung) und $\tilde{d}_{i,6}$ sind es nur, wenn keine Kondensatpumpe existiert ($\tilde{d}_{i,5} = 0$). Alle Parameter $\tilde{d}_{i,2}, \dots, \tilde{d}_{i,9}$ sind überhaupt nur variabel bzw. stehen zur Verfügung, wenn es die Anzapfung gibt ($\tilde{d}_{i,1} = 1$).

Die **diskrete Zufallsvariable** X_j^i repräsentiere die Zufallswahl des Wertes der Variablen $\tilde{d}_{i,j}$, $j \in \{1, \dots, 9\}$, falls diese zur Verfügung steht. Dann erhalten wir z.B.

$$P\{X_1^i = 0\} \quad (4.13)$$

für die Wahrscheinlichkeit, daß die Anzapfung i nicht vorhanden ist, und

$$P\{X_1^i = 1\}P\{X_2^i = 1\}P\{X_3^i = 1\}P\{X_4^i = 2\} \\ P\{X_5^i = 0\}P\{X_6^i = 1\}P\{X_7^i = 1\}P\{X_8^i = 2\}P\{X_9^i = 3\} \quad (4.14)$$

für die Wahrscheinlichkeit des Konstruktes "Anzapfung i existiert mit einem um zwei Positionen vorgezogenen, separat installierten Enthitzer, ohne daß das Kondensat umgepumpt wird, was wiederum die Existenz des um zwei Positionen nachgeschalteten, separat angebrachten Kondensatkühlers mit spezieller Kondensateinleitung ermöglicht".

Nehmen wir zunächst an, die Wahrscheinlichkeitsverteilung des Zufallsexperimentes X_1^i berücksichtigt nicht, wie viele Strukturvariationen bei $X_1^i = 1$ (bis zu 96) im Vergleich zu der Situation $X_1^i = 0$ (keine) möglich sind; dies ist z.B. der Fall, wenn man jedes X_j^i , $j = 1, \dots, 9$, bei Gleichverteilung unter allen zulässigen Werten für $\tilde{d}_{i,j}$ wählt: Wahrscheinlichkeit (4.13) ist dann gleich $\frac{1}{2}$, Wahrscheinlichkeit (4.14) hat den viel niedrigeren Wert $\frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{4} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{3} \frac{1}{3}$ - Gleichung (4.10) gilt ganz offensichtlich nicht. Außerdem bevorzugt hier das Zufallsexperiment $X_{prop}^D(x_k, y)$ die Nichtexistenz der Anzapfung jeder einzelnen Ausführung der Anzapfung, was nicht sehr logisch ist. Es ist nicht nur notwendig, Gleichung (4.10) zu erfüllen, sondern auch sinnvoll, jeder Strukturvariation von Anzapfung i die gleiche Chance zu geben, da nicht vorhersehbar ist, welche Strukturvariationen zu den paretooptimalen Alternativen des Problems (3.2) gehören. All diese Gedanken führen darauf hin, das Zufallsexperiment X_j^i , $j = 1, \dots, 9$, sofern es durchgeführt werden muß, folgendermaßen zu definieren:

Für jeden Knoten im Entscheidungsbaum, der einen variablen Parameter $\tilde{d}_{i,j}$, $i \in \{1, \dots, \alpha\}$, $j \in \{1, \dots, 9\}$, abbildet (Entscheidungsknoten), bestimme man die Anzahl $z_{ij}^k \in \mathbb{N}_0$ von diskreten Variationsmöglichkeiten, die von jeder einzelnen Entscheidung $\tilde{d}_{i,j} = w_{ij}^k$, $\tilde{d}_{i,j}$ mit einem der z_{ij} möglichen zulässigen Werte w_{ij}^k , $k \in \{1, \dots, z_{ij}\}$, zu belegen, abhängen. Dann bekommt das Ergebnis w_{ij}^k von X_j^i die Wahlwahrscheinlichkeit $\frac{z_{ij}^k}{\sum_{l=1}^k z_{ij}^l}$ zugewiesen. Wir machen uns dieses Vorgehen wieder am Beispiel der Anzapfung i mit den in Abbildung 4.6 dargestellten diskreten Parametern $\tilde{d}_{i,1}, \dots, \tilde{d}_{i,9}$ deutlich:

Parameter	z_{il}	$\sum_{l=1}^k z_{ij}^l$	w_{ij}^l	z_{ij}^l	$P\{X_j^i\}$
$\tilde{d}_{i,4}$	4	4	0	1	$P\{X_4^i = 0\} = \frac{1}{4}$
			1	1	$P\{X_4^i = 1\} = \frac{1}{4}$
			2	1	$P\{X_4^i = 2\} = \frac{1}{4}$
			3	1	$P\{X_4^i = 3\} = \frac{1}{4}$
$\tilde{d}_{i,3}$	2	5	0	1	$P\{X_3^i = 0\} = \frac{1}{5}$
			1	4	$P\{X_3^i = 1\} = \frac{4}{5}$
$\tilde{d}_{i,2}$	2	6	0	1	$P\{X_2^i = 0\} = \frac{1}{6}$
			1	5	$P\{X_2^i = 1\} = \frac{5}{6}$
$\tilde{d}_{i,8}$	3	3	0	1	$P\{X_8^i = 0\} = \frac{1}{3}$
			1	1	$P\{X_8^i = 1\} = \frac{1}{3}$
			2	1	$P\{X_8^i = 2\} = \frac{1}{3}$
$\tilde{d}_{i,7}$	2	4	0	1	$P\{X_3^i = 0\} = \frac{1}{4}$
			1	3	$P\{X_3^i = 1\} = \frac{3}{4}$
$\tilde{d}_{i,6}$	2	5	0	1	$P\{X_6^i = 0\} = \frac{1}{5}$
			1	4	$P\{X_6^i = 1\} = \frac{4}{5}$
$\tilde{d}_{i,9}$	3	3	1	1	$P\{X_9^i = 1\} = \frac{1}{3}$
			2	1	$P\{X_9^i = 2\} = \frac{1}{3}$
			3	1	$P\{X_9^i = 3\} = \frac{1}{3}$
$\tilde{d}_{i,5}$	2	16	0	15	$P\{X_5^i = 0\} = \frac{15}{16}$
			1	1	$P\{X_5^i = 1\} = \frac{1}{16}$
$\tilde{d}_{i,1}$	2	97	0	1	$P\{X_1^i = 0\} = \frac{1}{97}$
			1	96	$P\{X_1^i = 1\} = \frac{96}{97}$

Wahrscheinlichkeit (4.13) ist jetzt vom Wert $\frac{1}{97}$ und Wahrscheinlichkeit (4.14) vom Wert $\frac{96 \cdot 5 \cdot 4 \cdot 1 \cdot 15 \cdot 4 \cdot 3 \cdot 1 \cdot 1}{97 \cdot 6 \cdot 5 \cdot 4 \cdot 16 \cdot 5 \cdot 4 \cdot 3 \cdot 3} = \frac{1}{97}$. Da diskrete Variationen der Auslegung einer einzelnen Anzapfung unabhängig von denen anderer Anzapfungen vorgenommen werden können, folgt aus den Ergebnissen bei Anzapfung i für die Strukturvariationen der Auslegung der gesamten Vorwärmstrecke, daß Gleichung (4.10) gilt und das Zufallsexperiment $X_{prop}^D(x_k, y)$ jede mögliche Belegung diskreter Parameter jetzt mit gleicher Wahrscheinlichkeit auswählt.

Weitere Schwierigkeiten ergeben sich dabei, der Forderung (4.11) gerecht zu werden. Mit dem Ausgang z_D des Zufallsexperimentes $X_{prop}^D(x_k, y)$ wurde eine bestimmte Menge E^{z_D} kontinuierlich effizienter Punkte ausgezeichnet, aus der nun im Zufallsexperiment $X_{prop}^{z_D}(x_k, y)$ ein Element $z = (z_C, z_D)$ auszusuchen ist. Da man keine logischen Argumente finden kann, weshalb dabei ein Element aus E^{z_D} einem anderen Element aus E^{z_D} vorzuziehen wäre, bietet sich als Wahrscheinlichkeitsverteilung von $X_{prop}^{z_D}(x_k, y)$ die Gleichverteilung an. Aber für $y_D \neq z_D$ gilt meist $|\bar{E}^{y_D}| \neq |\bar{E}^{z_D}|$, und Gleichung (4.11) ist gefährdet. Ein Ausweg ist, z aus der Vereinigungsmenge $\bar{E}^{y_D} \cup \bar{E}^{z_D} \setminus \{y\}$ unter Gleichverteilung zu wählen. Dann gilt

$$P\{X_{prop}^{z_D}(x_k, y) = z\} = \begin{pmatrix} \frac{1}{|\bar{E}^{y_D}| + |\bar{E}^{z_D}| - 1}, & \text{falls } y_D \neq z_D \\ \frac{1}{|\bar{E}^{y_D}| - 1}, & \text{falls } y_D = z_D \end{pmatrix} = P\{X_{prop}^{y_D}(x_k, z) = y\}.$$

Mit diesem Wissen über die Wahrscheinlichkeitsverteilungen von $X_{prop}^D(x_k, y)$ und $X_{prop}^{z_D}(x_k, y)$ kann für jeden Nachbarn (x_k, z) jedes beliebigen Punktes (x_k, y) die Generierungswahrscheinlichkeit $G_{(x_k, y)(x_k, z)}$ bestimmt werden. Dies **definiert** die **Nachbarschaftsstruktur**, die dem Metropolis Algorithmus, der für $\bar{x} = x_k$ an Problem (3.21) arbeitet, zugrunde liegt. Da jedes Ergebnis von $X_{prop}^D(x_k, y)$ mit gleicher Wahrscheinlichkeit eintritt, bezeichnen wir deren Wert aus $]0; 1[$ mit $p_{prop}^D(x_k, y)$.

$$G_{(x_k, y)(x_k, z)} = \begin{cases} P\{X_{prop}^D(x_k, y) = z_D\} P\{X_{prop}^{z_D}(x_k, y) = z\} \\ = p_{prop}^D(x_k, y) \frac{1}{|\bar{E}^{y_D}| + |\bar{E}^{z_D}| - 1}, & \text{falls } y_D \neq z_D, \\ \sum_{\bar{z}_D: (\bar{z}_C, \bar{z}_D) \in R_k} P\{X_{prop}^D(x_k, y) = \bar{z}_D\} P\{X_{prop}^{\bar{z}_D}(x_k, y) = z\} \\ = p_{prop}^D(x_k, y) \left[\frac{1}{|\bar{E}^{y_D}| - 1} + \sum_{y_D \neq \bar{z}_D: (\bar{z}_C, \bar{z}_D) \in R_k} \frac{1}{|\bar{E}^{y_D}| + |\bar{E}^{\bar{z}_D}| - 1} \right], & \text{falls } y_D = z_D \end{cases} \quad (4.15)$$

entspricht zwar nicht ganz der Gleichung (4.8), aber $G_{(x_k, y)(x_k, z)} = G_{(x_k, z)(x_k, y)}$ ist dennoch garantiert dadurch, daß $p_{prop}^D(x_k, y) = p_{prop}^D(x_k, z)$ für alle $y, z \in R_k$ gilt. Die Nachbarn (x_k, z) von (x_k, y) mit $z \in E^{y_D}$ haben bessere Chancen, als neuer Lösungskandidat des Metropolis Algorithmus (NMA) vorgeschlagen zu werden als sämtliche Nachbarn (x_k, z) mit $z \in E^{x_D}$ anderer diskreter Variationen $x_D \neq y_D$. Erstere kann man tatsächlich als Nachbarn interpretieren, die "näher" am Standpunkt liegen als alle anderen Nachbarn. Man stelle sich vor, der Metropolis Algorithmus (NMA) sei in eine (irgendwie definierte) nähere Umgebung eines globalen Minimums von Problem (3.21) gekommen. Dann könnte es vorteilhaft sein und ist auf jeden Fall plausibel, speziell den Alternativen der Nachbarschaft, die gleiche diskrete Werte haben (Auslegungsvarianten mit gleichem Strukturdesign), besondere Aufmerksamkeit zu schenken, da in gleichem Strukturdesign eventuell ähnlich gute Eigenschaften, wie sie der Zielfunktionswert Z des momentanen Standpunktes ausdrückt, stecken.

4.1.4 Eigenschaften der Markov Kette, die den Metropolis Algorithmus modelliert

Da nun die Nachbarschaftsstruktur, die dem Metropolis Algorithmus, der Problem (3.21) für die Referenzalternative $\bar{x} = x_k$ bearbeitet, zugrunde liegt, mit symmetrischen Generierungswahrscheinlichkeiten definiert ist, sollte untersucht werden, ob Satz 3.2.22 gilt. Die Markov Kette

$\{X_t\}_{t \in \mathbb{N}_0}$ muß irreduzibel und aperiodisch sein, damit eine stationäre Verteilung der Form

$$q_{(x_k, y)}(c) := \frac{1}{N_0(c)} \exp\left(-\frac{Z[(x_k, y)]}{c}\right) \quad \forall (x_k, y) \in S_k, c \in \mathbb{R}^+,$$

$$N_0(c) := \sum_{(x_k, z) \in S_k} \exp\left(-\frac{Z[(x_k, z)]}{c}\right),$$
(4.16)

existieren kann. Die genauen Definitionen der beiden geforderten Eigenschaften kann der Leser in Abschnitt 3.2.5.3 nachlesen.

4.1.4.1 Irreduzibilität der Markov Kette

Es muß garantiert sein, daß die Übergangswahrscheinlichkeiten der Markov Kette

$$\forall (x_k, y), (x_k, z) \in S_k \quad \exists n \in \mathbb{N} : P_{(x_k, y)(x_k, z)}^n(c) > 0, c \in \mathbb{R}^+,$$
(4.17)

erfüllen. Mit

$$P_{(x_k, y)(x_k, z)}(c) := \begin{cases} G_{(x_k, y)(x_k, z)} A_{(x_k, y)(x_k, z)}(c), & \text{falls } (x_k, y) \neq (x_k, z) \\ 1 - \sum_{(x_k, \bar{z}) \neq (x_k, z) \in S_k} P_{(x_k, y)(x_k, \bar{z})}(c), & \text{falls } (x_k, y) = (x_k, z) \end{cases}, c \in \mathbb{R}^+,$$
(4.18)

und $G_{(x_k, y)(x_k, z)} > 0 \quad \forall (x_k, y) \neq (x_k, z) \in S_k$ und $A_{(x_k, y)(x_k, z)}(c) > 0 \quad \forall (x_k, z) \in S_k, c \in \mathbb{R}^+$, erhalten wir

$$P_{(x_k, y)(x_k, z)}(c) > 0 \quad \forall (x_k, z) \in S_k,$$
(4.19)

d.h. das n in Ungleichung (4.17) hat sogar den Wert 1 für alle $[(x_k, y), (x_k, z)] \in S_k \times S_k$. In der Tat kann der Metropolis Algorithmus (NMA) jeden Punkt in S_k von jedem Punkt in S_k aus in endlicher Zeit (sogar in nur einem Schritt) mit positiver Wahrscheinlichkeit erreichen.

4.1.4.2 Aperiodizität der Markov Kette

Die Periode $p \in \mathbb{N}$ jedes Punktes $(x_k, y) \in S_k$, die als der größte gemeinsame Teiler aller $n \in \mathbb{N}$ mit $P_{(x_k, y)(x_k, y)}^n(c) > 0, c \in \mathbb{R}^+$, definiert ist, muß den Wert 1 haben. Kann man beweisen, daß

$$P_{(x_k, y)(x_k, y)}^2(c) > 0 \quad \text{und} \quad P_{(x_k, y)(x_k, y)}^3(c) > 0 \quad \forall (x_k, y) \in S_k, c \in \mathbb{R}^+,$$
(4.20)

gelten, so ist die Aperiodizität gezeigt, da der größte gemeinsame Teiler von 2 und 3 gleich 1 ist.

Ohne Beschränkung der Allgemeinheit können wir $|S_k| \geq 3$ annehmen, da S_k mindestens aus einer Menge $E^{x_D}, x_D \in D$, bestehen muß; da Mehrzieloptimierungsprobleme naturgemäß viele paretooptimale Lösungen besitzen, ist die Annahme $|\bar{E}^{x_D}| \geq 3$ nicht unrealistisch. Wir benennen nun drei paarweise verschiedene Punkte (x_k, z^1) , (x_k, z^2) und (x_k, z^3) aus S_k . Da gemäß der in Abschnitt 4.1.3 definierten Nachbarschaftsstruktur $N_{(x_k, z^l)} := S_k \setminus \{(x_k, z^l)\}, l = 1, 2, 3$, gilt, ergibt sich offensichtlich aus Ungleichung (4.19)

$$P_{(x_k, z^1)(x_k, z^1)}^2(c) \geq P_{(x_k, z^1)(x_k, z^2)}(c) P_{(x_k, z^2)(x_k, z^1)}(c) > 0,$$

$$P_{(x_k, z^1)(x_k, z^1)}^3(c) \geq P_{(x_k, z^1)(x_k, z^2)}(c) P_{(x_k, z^2)(x_k, z^3)}(c) P_{(x_k, z^3)(x_k, z^1)}(c) > 0.$$
(4.21)

(x_k, z^1) steht für alle Punkte $(x_k, y) \in S_k$.

Die Generierungswahrscheinlichkeiten der Markov Kette $\{X_t\}_{t \in \mathbb{N}_0}$ sind symmetrisch, und Satz 3.2.22 garantiert die Existenz einer stationären Verteilung, die dafür sorgt, daß nach unendlich vielen (Theorie / IMA) bzw. einer genügend großen Zahl von Übergängen (Numerik / NMA) von Punkt zu Punkt der Metropolis Algorithmus bevorzugt globale Minima des bearbeiteten Problems findet.

4.1.5 Testergebnisse

Die in Abschnitt 4.1.2.4 angeführten Argumente liefern die Motivation für die Minimierung der Zweizielprobleme

$$\begin{aligned} & \text{"min"}_{x \in R}'' \begin{pmatrix} K_{Invest}(x) \\ (-1)\eta(x) \end{pmatrix}, \\ & \text{"min"}_{x \in R}'' \begin{pmatrix} K_{Strom}(x) \\ (-1)\eta(x) \end{pmatrix}, \end{aligned} \quad (4.22)$$

R definiert wie in Abschnitt 4.1.2.6.

In dem zu testenden Beispiel für ein Kraftwerk sind bei einer Anzapfung $i \in \{1, \dots, \alpha - 1\}$, $\alpha \in \mathbb{N}$ die maximale Anzahl von Anzapfungen, und ihrer benachbarten Anzapfung $i + 1$ folgende Parameter variierbar:

$\tilde{d}_{i,1} \in \{0, 1\}$	Anzapfung vorhanden,
$\tilde{d}_{i,2} \in \{0, 1\}$	Enthitzer vorhanden,
$\tilde{d}_{i,6} \in \{0, 1\}$	Kondensatkühler vorhanden,
$\tilde{c}_{i,1} \in [lb_{(i-1)9+1}, ub_{(i-1)9+1}]$	Anzapfdruck,
$\tilde{c}_{i,2} \in [lb_{(i-1)9+2}, ub_{(i-1)9+2}]$	Grädigkeit des Kondensationsvorwärmers (falls vorhanden) oder, falls ein Enthitzer existiert, Grädigkeit des Kondensationsvorwärmers mit integriertem Enthitzer (separater Enthitzer hier nicht möglich),
$\tilde{c}_{i,4} \in [lb_{(i-1)9+4}, ub_{(i-1)9+4}]$	Grädigkeit des Kondensatkühlers (falls vorhanden),
$\tilde{c}_{i+1,2} \in [lb_{i9+2}, ub_{i9+2}]$	Grädigkeit des (vorhandenen) Kondensationsvorwärmers (evtl. mit integriertem Enthitzer).

Wir können also an drei diskreten Variablen mit fünf möglichen diskreten Variationen und vier kontinuierlichen Variablen herumspielen.

Daraus ergeben sich fünf zu lösende kontinuierliche Subprobleme mit folgenden Variablen-Szenarien:

diskrete Wertebelegung	verfügbare kontinuierliche Variablen
$(\tilde{d}_{i,1}, \tilde{d}_{i,2}, \tilde{d}_{i,6}) = (0, 0, 0)$	$\tilde{c}_{i+1,2}$
$(\tilde{d}_{i,1}, \tilde{d}_{i,2}, \tilde{d}_{i,6}) = (1, 0, 0)$	$\tilde{c}_{i,1}, \tilde{c}_{i,2}, \tilde{c}_{i+1,2}$
$(\tilde{d}_{i,1}, \tilde{d}_{i,2}, \tilde{d}_{i,6}) = (1, 1, 0)$	$\tilde{c}_{i,1}, \tilde{c}_{i,2}, \tilde{c}_{i+1,2}$
$(\tilde{d}_{i,1}, \tilde{d}_{i,2}, \tilde{d}_{i,6}) = (1, 0, 1)$	$\tilde{c}_{i,1}, \tilde{c}_{i,2}, \tilde{c}_{i,4}, \tilde{c}_{i+1,2}$
$(\tilde{d}_{i,1}, \tilde{d}_{i,2}, \tilde{d}_{i,6}) = (1, 1, 1)$	$\tilde{c}_{i,1}, \tilde{c}_{i,2}, \tilde{c}_{i,4}, \tilde{c}_{i+1,2}$

Die kontinuierlichen Subprobleme werden mit Hilfe des in Abschnitt 2.3.2.3 vorgestellten kontinuierlichen Vektoroptimierungsverfahrens gelöst, das für den Umgang mit einem durch Nebenbedingungen eingeschränkten R modifiziert wurde.

Einige Symbole werden in den Beschriftungen der folgenden Grafiken immer wieder erscheinen:

$\underline{t} \in \mathbb{N}$	erste Iteration einer Metropolis-Suche (NMA), die ein (vorläufiges) Suchergebnis liefern könnte
$\bar{t} \in \mathbb{N}$	maximal erlaubte Iterationszahl für eine Metropolis-Suche (NMA), $\bar{t} \geq \underline{t}$
$nrp \in \mathbb{N}$	Anzahl der Referenzpunkte aus R_D , für die in der Suchstrategie SS (Abschnitt 3.2.2) ein Problem (3.21) gelöst wurde
\circ	Symbol für einen Punkt aus der numerischen Approximation von $F(R_D)$
*	Symbol für den Zielwertvektor einer vom getesteten Mehrzielverfahren gefundenen Alternative, die vom Mehrzielverfahren als effizient eingestuft wird; zeigt die Grafik ein Beispiel für eine Menge von Startpunkten, so bezeichnet dieses Symbol lediglich den Zielwertvektoren eines vom Verfahren generierten Startpunktes

Die Punktwolken in den folgenden Grafiken bilden jeweils $F(\bar{R}_D)$ ab, das Bild der numerischen Approximation von R_D oder eines Ausschnittes aus R_D , die vom in Abschnitt 2.3.2.3 vorgestellten stochastischen kontinuierlichen Mehrzielverfahren im jeweiligen Suchlauf erarbeitet wurde. \bar{R}_D stellt quasi die Information dar, die das diskret-kontinuierliche stochastische Mehrzielverfahren in einem Suchlauf über das wahre R_D sammeln konnte. Da die Mächtigkeit von \bar{R}_D in keinem Suchlauf besonders groß ist, machen die visualisierten Testergebnisse deutlich, daß das in dieser Arbeit entwickelte stochastische Mehrzielverfahren (Ausführungen DKMOP SS und DKMOP PS) auch bei Problemen mit relativ kleinem Suchraum nützlich ist, obwohl es vor allem auf sehr große Suchräume abzielt: Es liefert eine (Approximation für eine) repräsentative Teilmenge der Effizienzmenge in R_D und $F(R_D)$ selbst bei nichtkonvexem $F(R_D)$. Es liefen Tests der Verfahrensstrategien SS und PS mit verschiedenen Werten für \underline{t} und \bar{t} aus \mathbb{N} . Das sind die aus Abschnitt 3.2.8.2 bekannten Parameter $\underline{L}_k, \bar{t}_k$ (Suchstrategie SS) bzw. $\underline{L}_k^{i_k}, \bar{t}_k^{i_k}$ (Suchstrategie PS), $k \in \mathbb{N}_0$, $i_k \in \{1, \dots, s_k\}$, $s_k \in \mathbb{N}$. Variationen des Kontrollparameters $c \in \mathbb{R}^+$, der die Akzeptanzwahrscheinlichkeiten für Zielwertverschlechterungen beim Lösen der verschiedenen Probleme (3.21) steuert, ließen nicht erkennen, daß sich ein bestimmtes Akzeptanzniveau für Verschlechterungen besonders auszeichnete. Stets waren *min* mit -1 , *mid* mit 0 und *max* mit 1 belegt. Je nach Wahl von c ergaben sich Akzeptanzwahrscheinlichkeiten für eine Verschlechterung von *min* auf *mid* bzw. *mid* auf *max* bei der Bearbeitung eines Problems (3.21) von etwa 0.01 , 0.5 oder 0.95 .

4.1.5.1 Suchstrategie PS im Test

Bei Optimierungen der Zweizielfunktion $(K_{Invest}, -\eta)^T$ starteten die Suchläufe immer mit 20 Startpunkten, bei der Zweizielfunktion $(K_{Strom}, -\eta)^T$ mit 10 Startpunkten. Die Startpunktemenge wurde wie gewohnt L_0 genannt. Jeder Startpunkt wurde zufällig unter Gleichverteilung aus der numerisch erzeugten Approximation von $E^{(0,0,0)}$ gewählt. Ein Blick auf die entsprechenden Grafiken zeigt, daß dies im gegebenen Fallbeispiel sogar eine (absichtlich) schlechte Wahl von L_0 darstellt. Trotz der unvoreilhaftigen Startpunktewahl endeten die Suchläufe nach ein bis zwei Listengenerationen $L_k, k \in \mathbb{N}$, bereits erfolgreich.

Abbildung 4.7: Beispiel für eine Menge von Startpunkten bei einem Testlauf der Suchstrategie PS mit den Zielfunktionen K_{Invest} und $(-1)\eta$

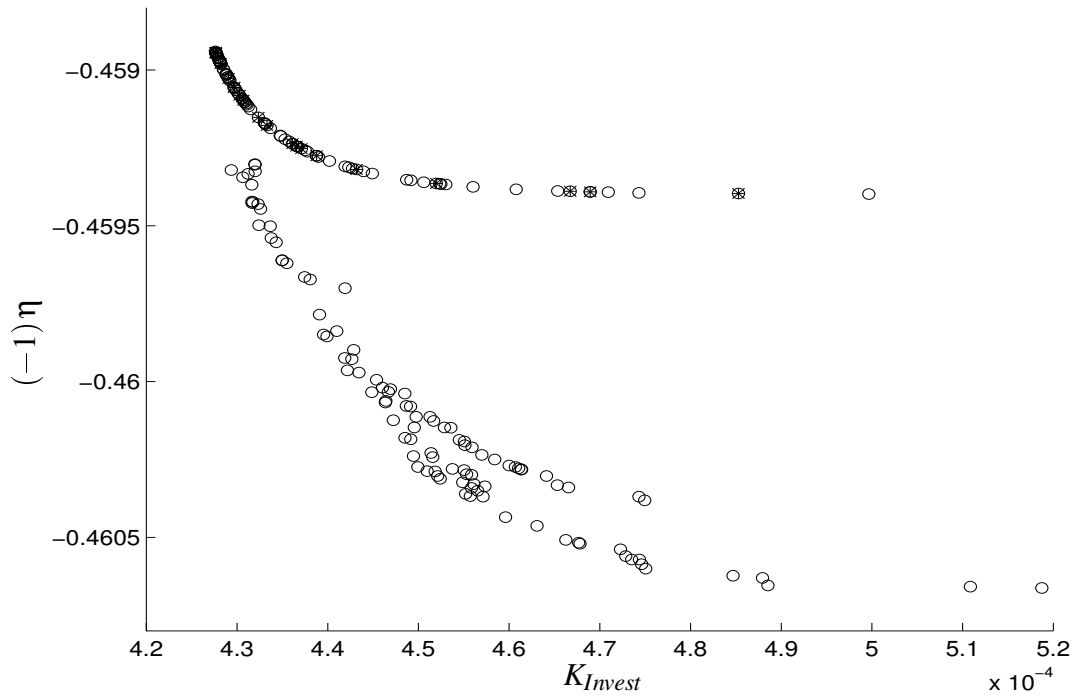


Abbildung 4.8: Suchstrategie PS mit $\underline{t} = 0, \bar{t} = 90, |\bar{R}_D| = 197$

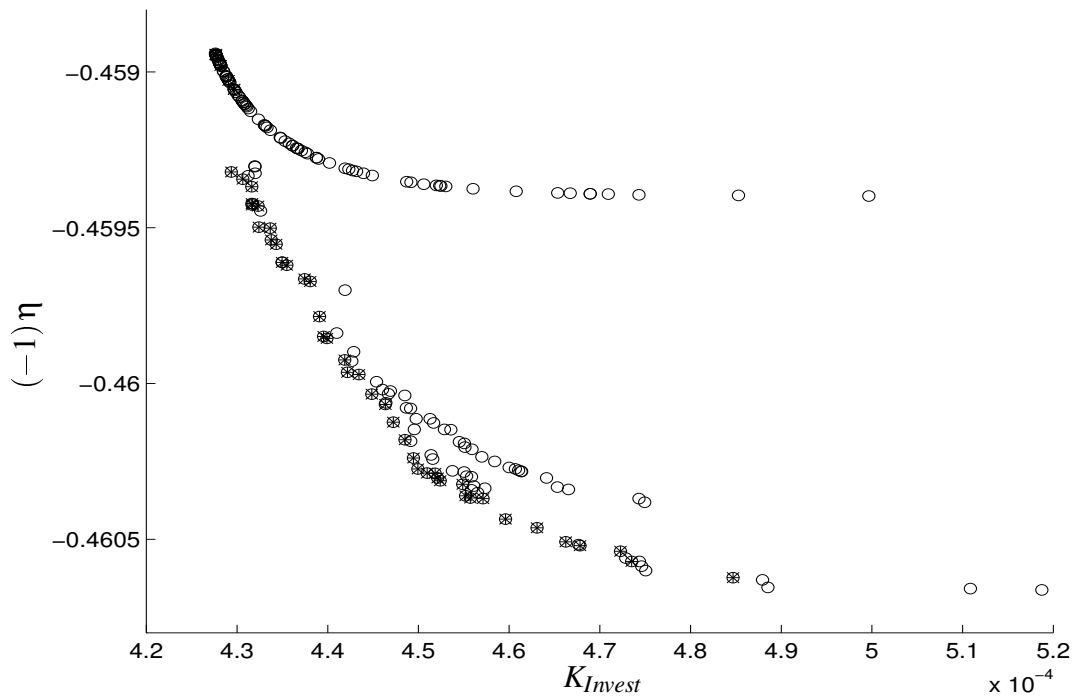


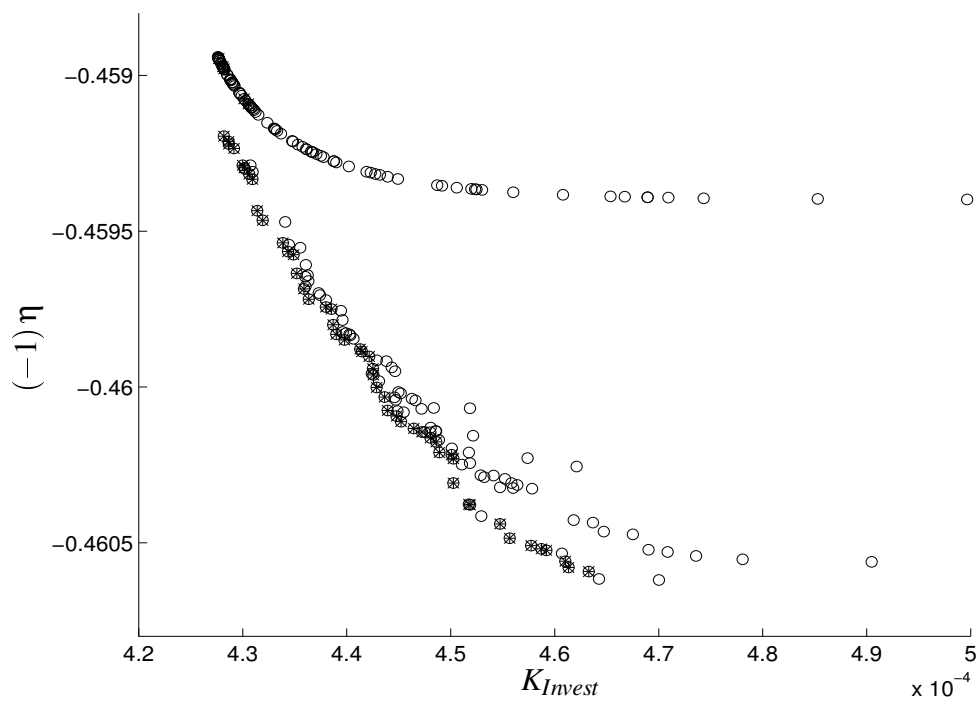
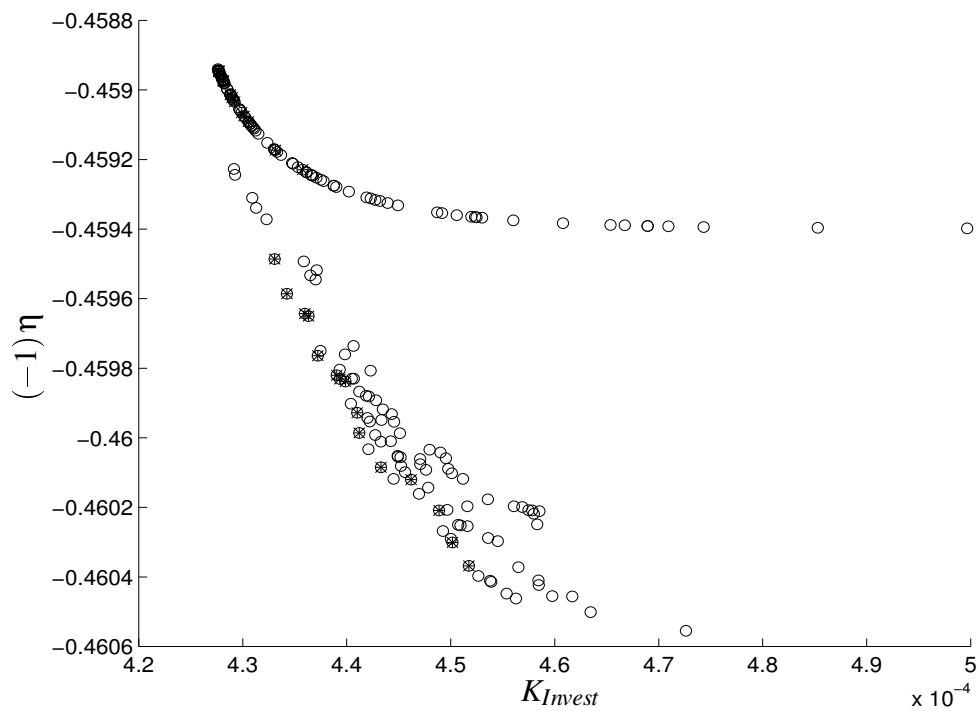
Abbildung 4.9: Suchstrategie PS mit $\underline{t} = 0, \bar{t} = 45, |\bar{R}_D| = 217$ Abbildung 4.10: Suchstrategie PS mit $\underline{t} = 0, \bar{t} = 10, |\bar{R}_D| = 188$ 

Abbildung 4.11: Suchstrategie PS mit $\underline{t} = 80, \bar{t} = 90, |\bar{R}_D| = 189$

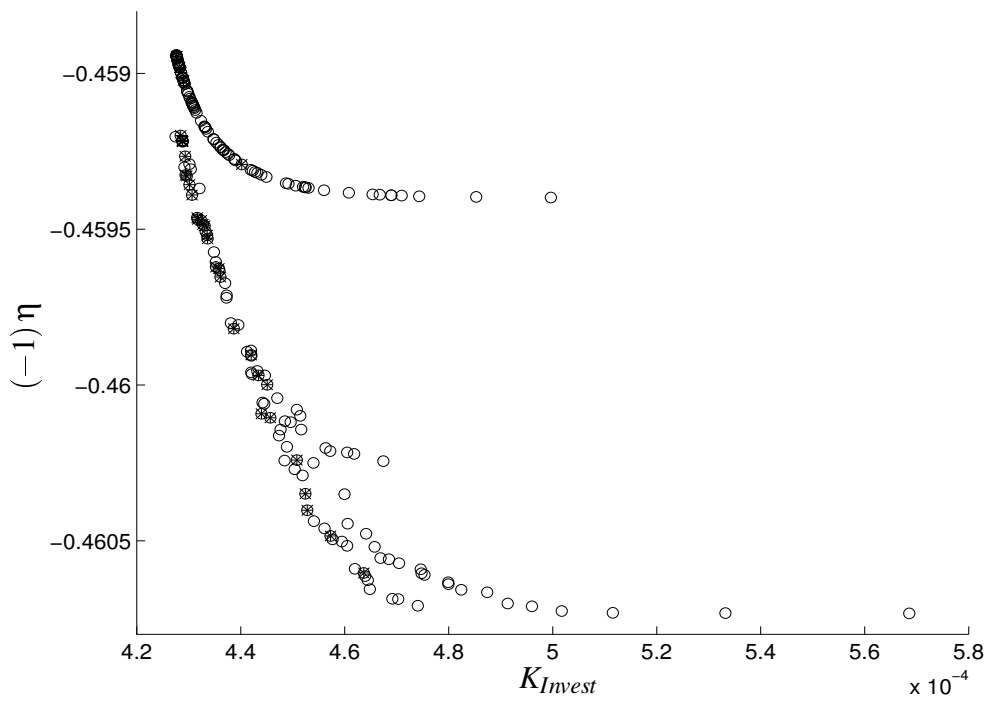


Abbildung 4.12: Suchstrategie PS mit $\underline{t} = 35, \bar{t} = 45, |\bar{R}_D| = 200$

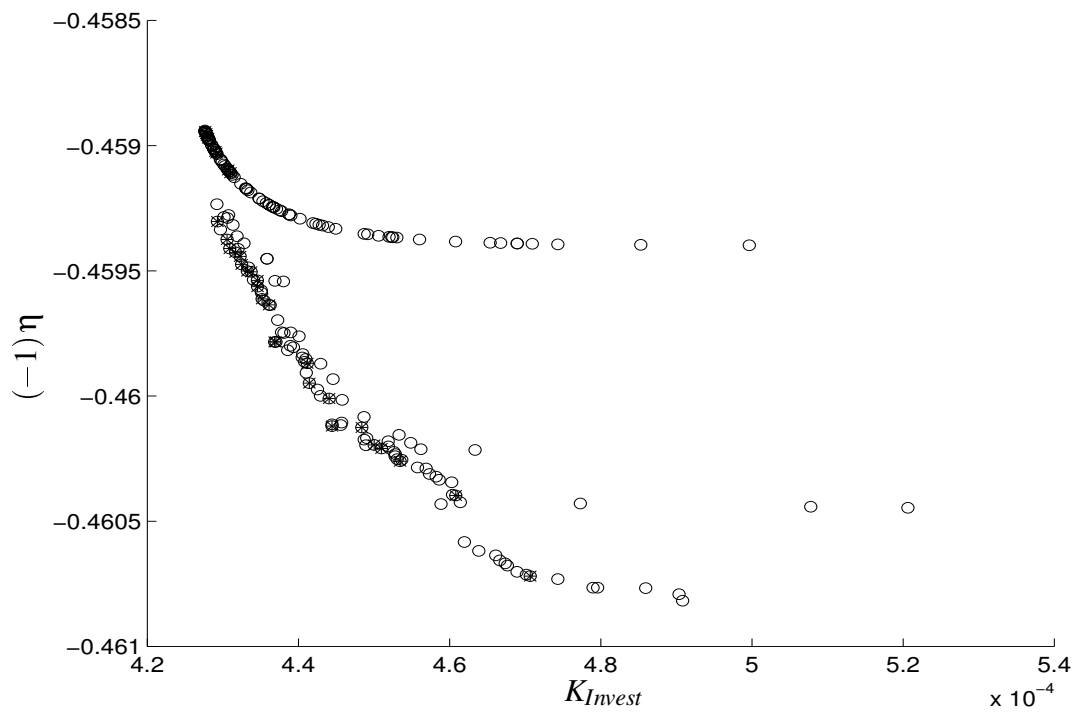


Abbildung 4.13: Beispiel für eine Menge von Startpunkten bei einem Testlauf der Suchstrategie PS mit den Zielfunktionen K_{Strom} und $(-1)\eta$

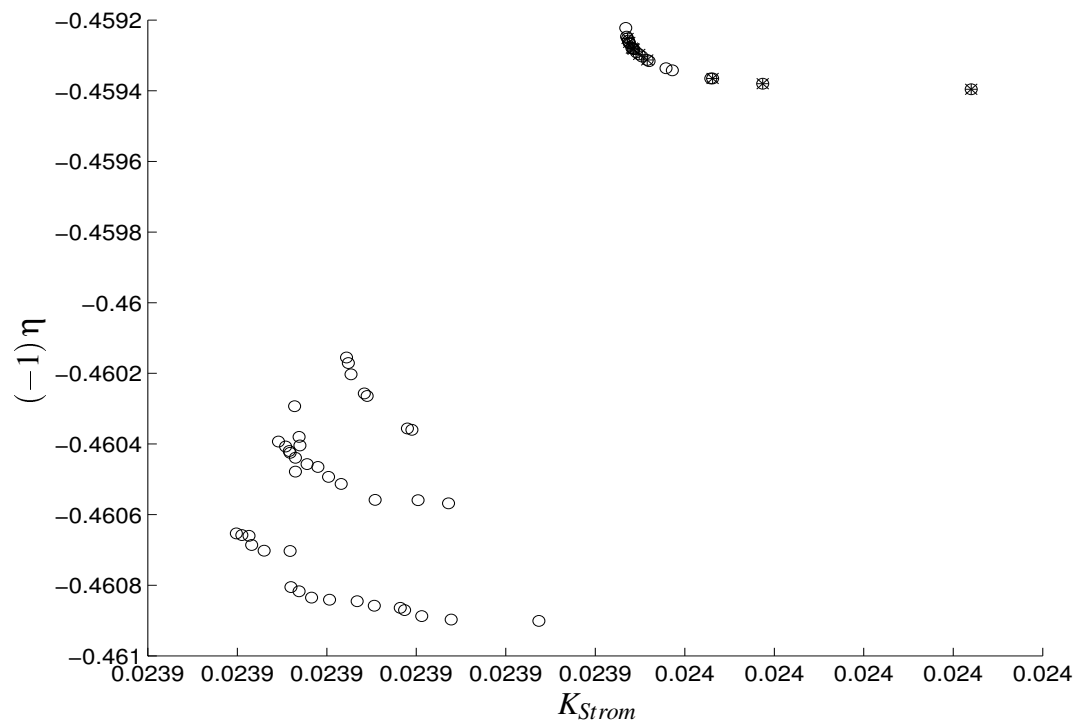


Abbildung 4.14: Suchstrategie PS mit $\underline{t} = 0, \bar{t} = 25, |\bar{R}_D| = 60$

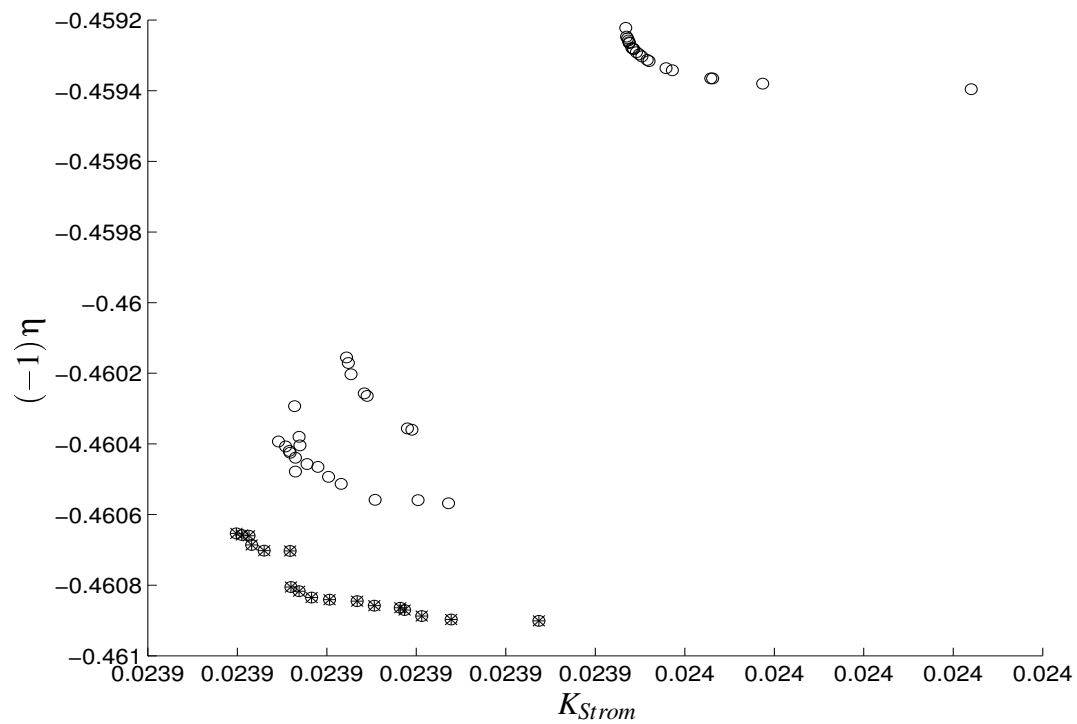


Abbildung 4.15: Suchstrategie PS mit $\underline{t} = 0, \bar{t} = 15, |\bar{R}_D| = 85$

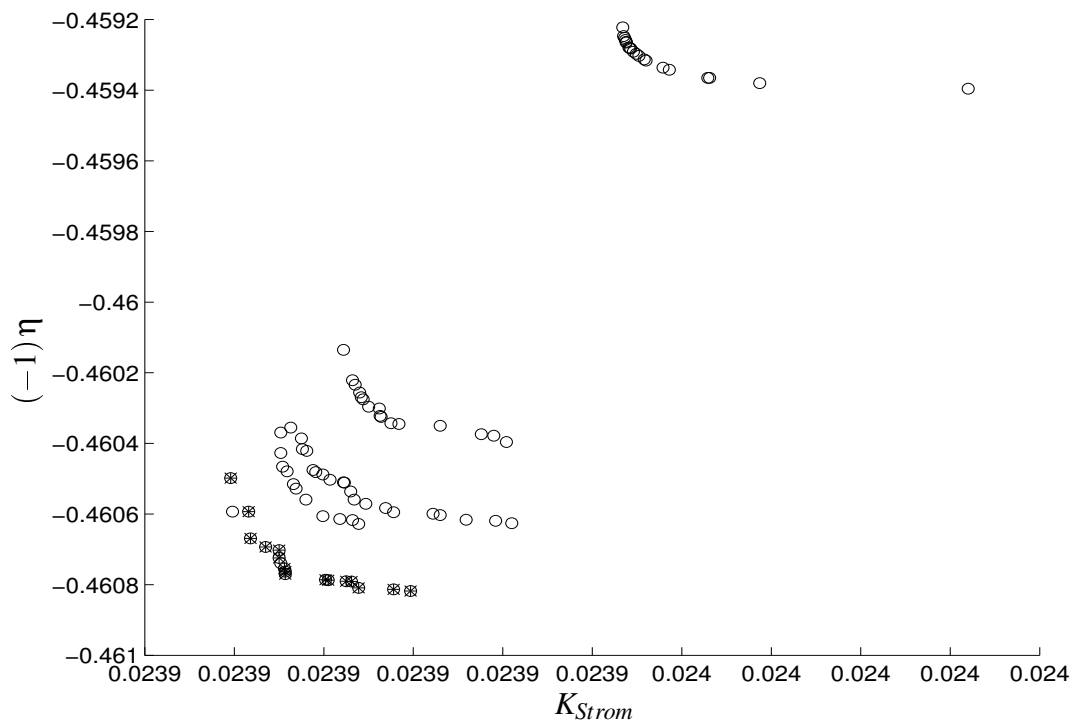


Abbildung 4.16: Suchstrategie PS mit $\underline{t} = 0, \bar{t} = 6, |\bar{R}_D| = 73$

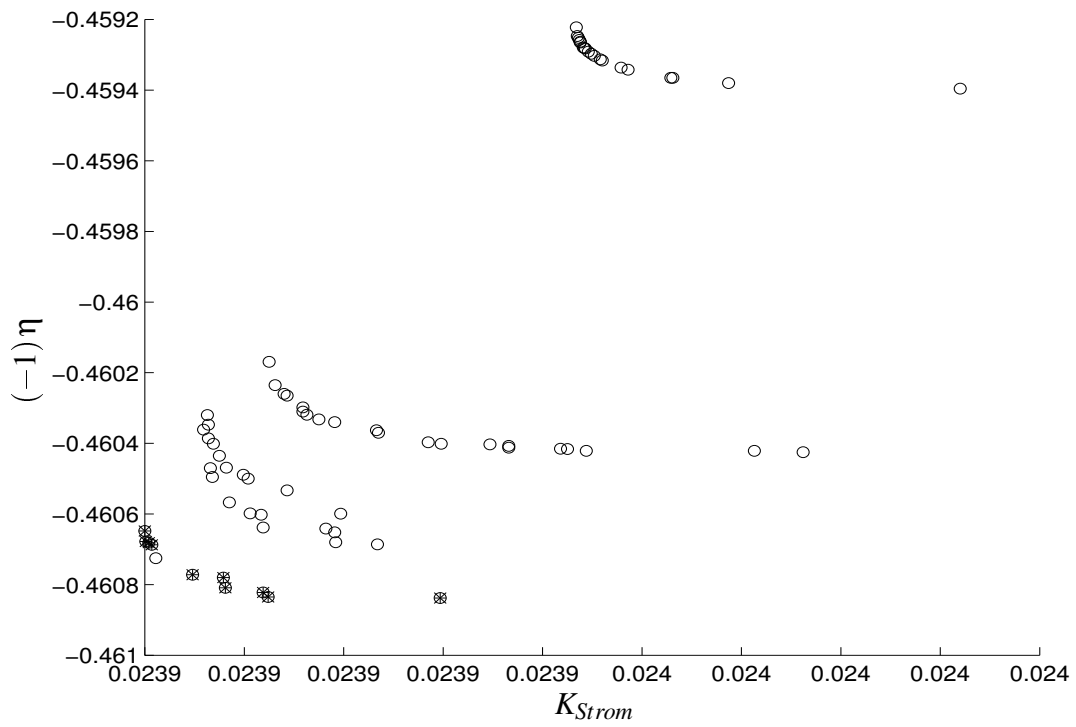
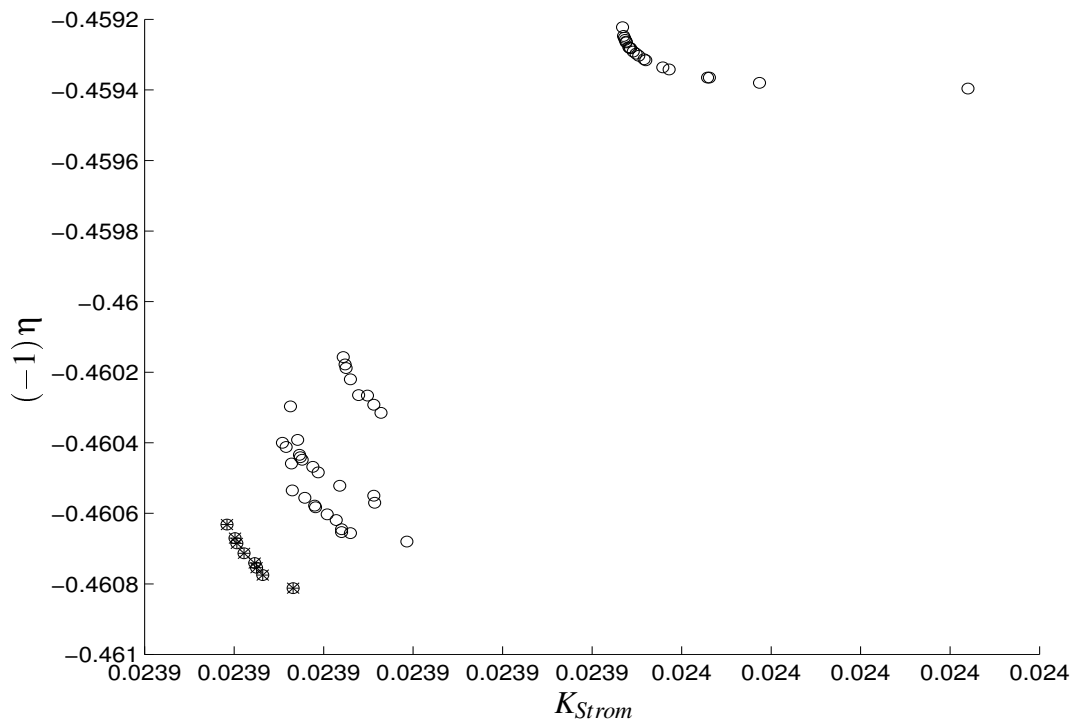
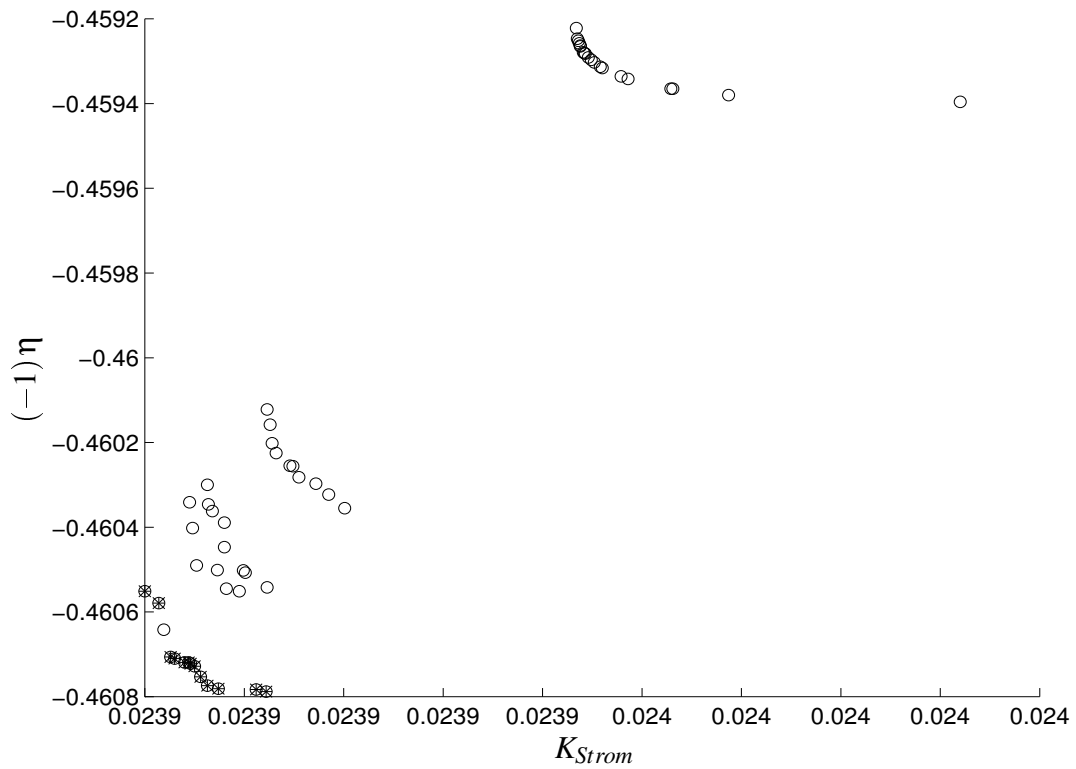


Abbildung 4.17: Suchstrategie PS mit $\underline{t} = 10, \bar{t} = 25, |\bar{R}_D| = 60$ Abbildung 4.18: Suchstrategie PS mit $\underline{t} = 10, \bar{t} = 15, |\bar{R}_D| = 58$ 

4.1.5.2 Suchstrategie SS im Test

Je Testlauf wurde der Startpunkt x_0 zufällig unter Gleichverteilung aus \bar{R}_D gewählt. Sobald ein Testlauf 10 der gefundenen Punkte als effizient eingestuft hatte, wurde er gestoppt. Einige der erzeugten Lösungsalternativen haben gleiche Zielfunktionswerte.

Abbildung 4.19: Suchstrategie SS mit $\underline{t} = 0, \bar{t} = 25, |\bar{R}_D| = 202, nrp = 13$

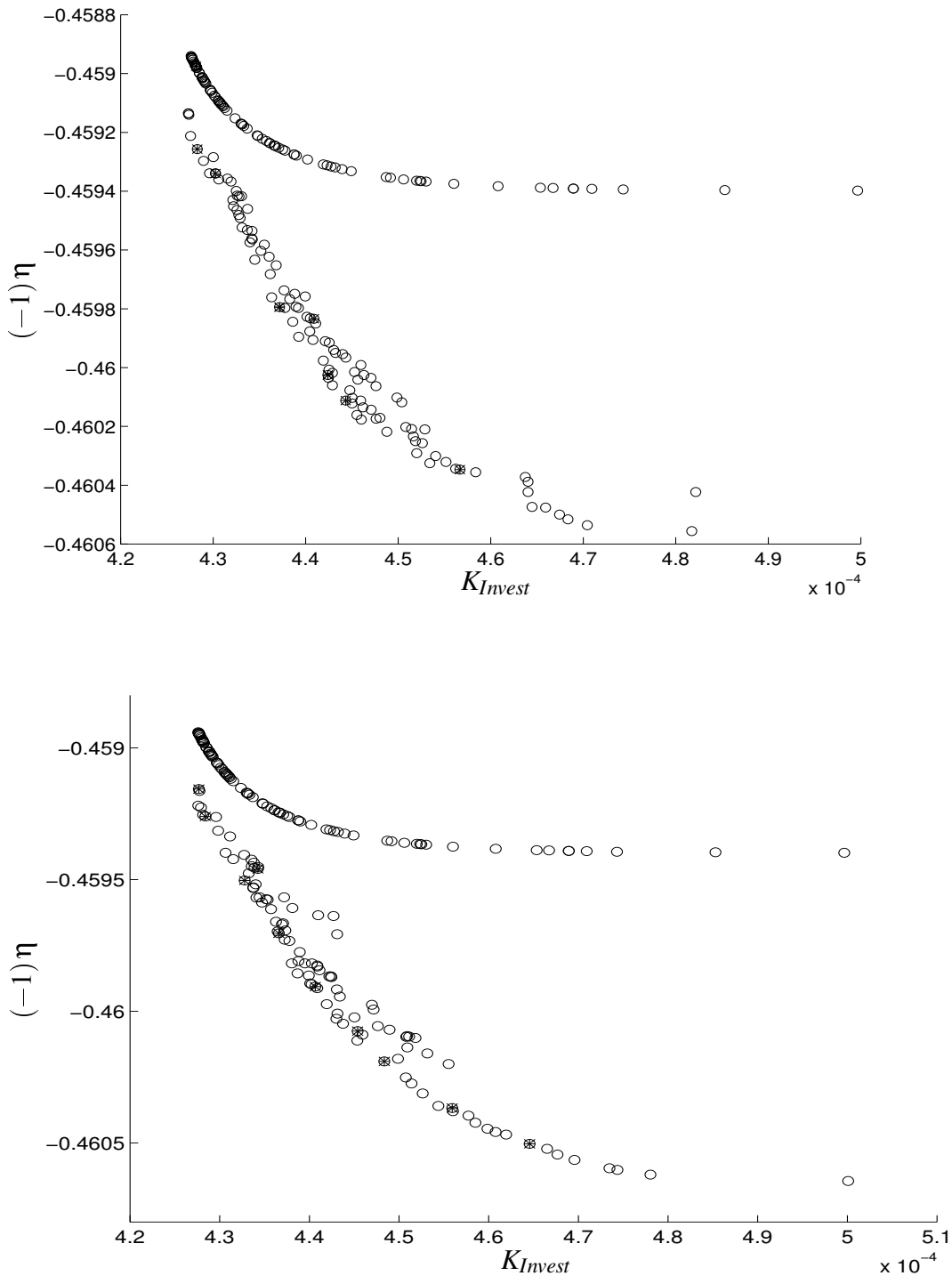


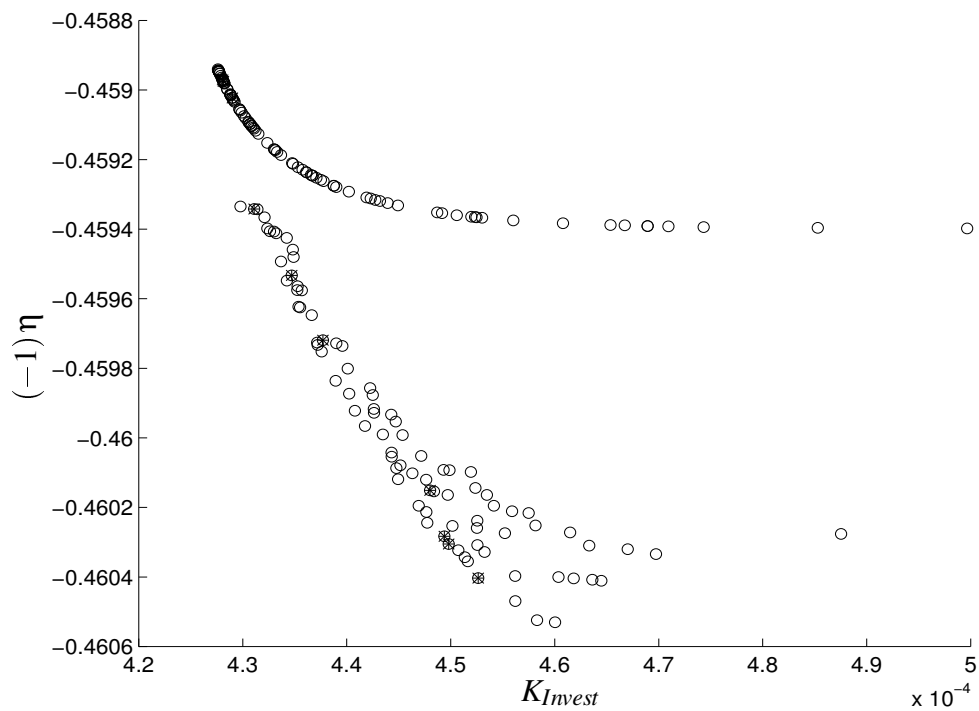
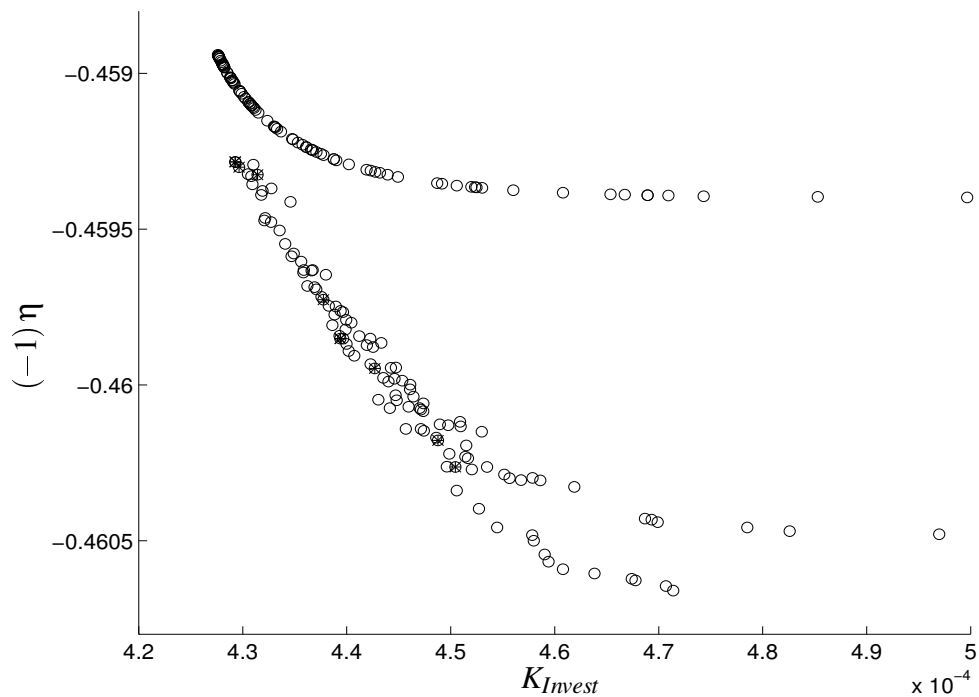
Abbildung 4.21: Suchstrategie SS mit $\underline{t} = 0, \bar{t} = 6, |\bar{R}_D| = 182, nrp = 13$ Abbildung 4.22: Suchstrategie SS mit $\underline{t} = 10, \bar{t} = 25, |\bar{R}_D| = 208, nrp = 15$ 

Abbildung 4.23: Suchstrategie SS mit $\underline{t} = 10, \bar{t} = 15, |\bar{R}_D| = 194, nrp = 13$

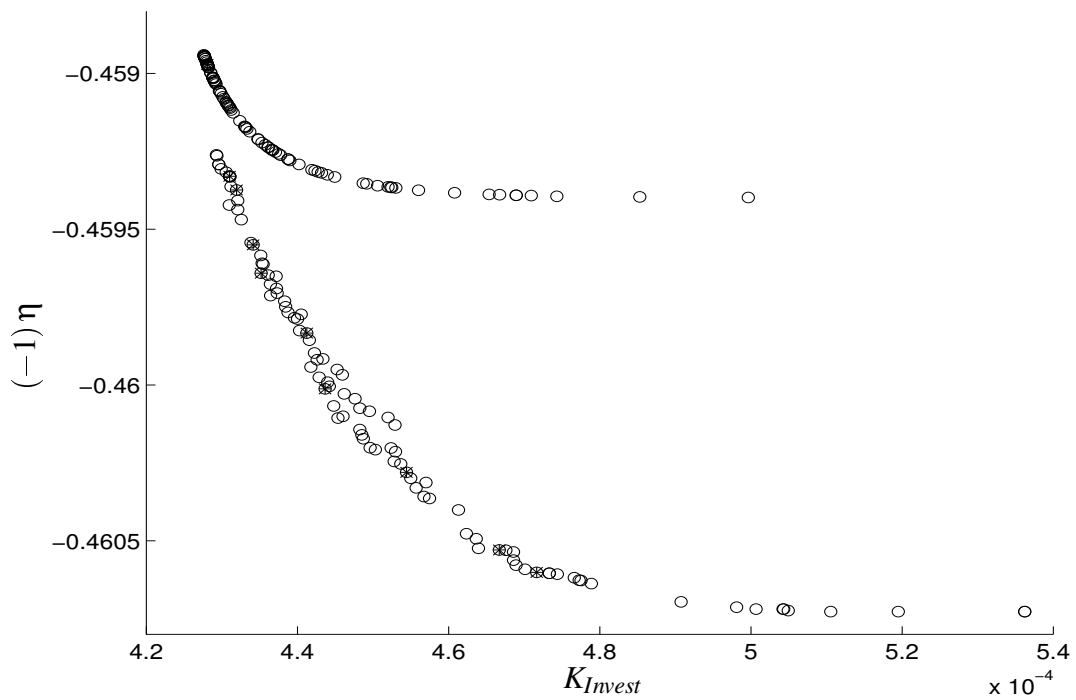


Abbildung 4.24: Suchstrategie SS mit $\underline{t} = 0, \bar{t} = 25, |\bar{R}_D| = 73, nrp = 26$

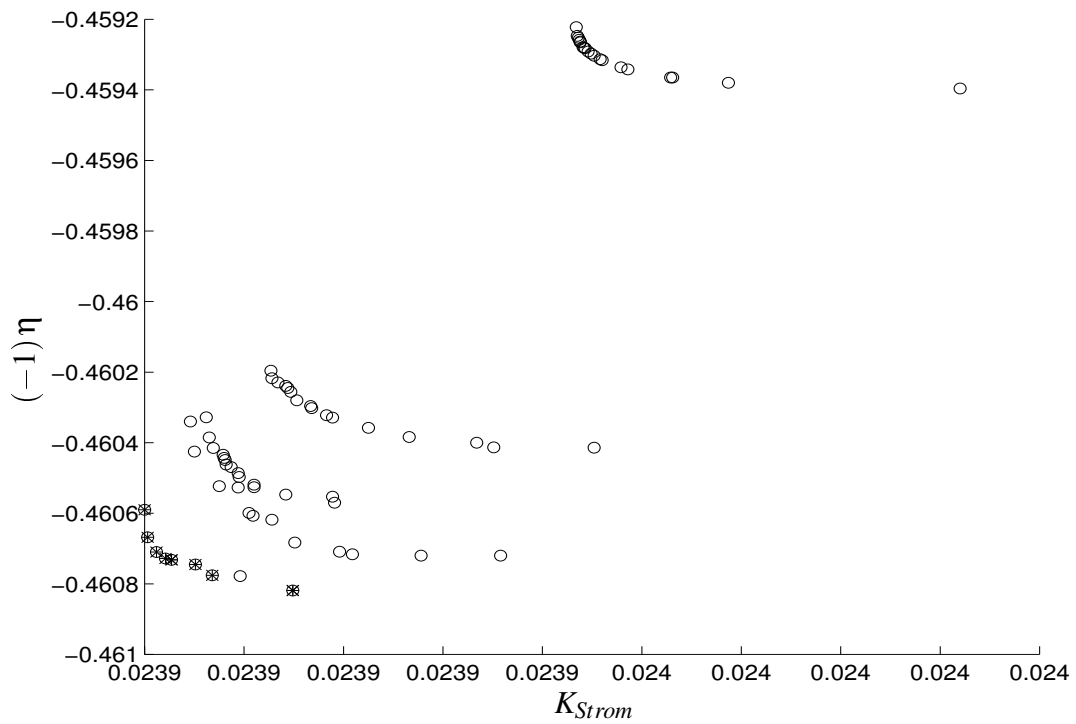
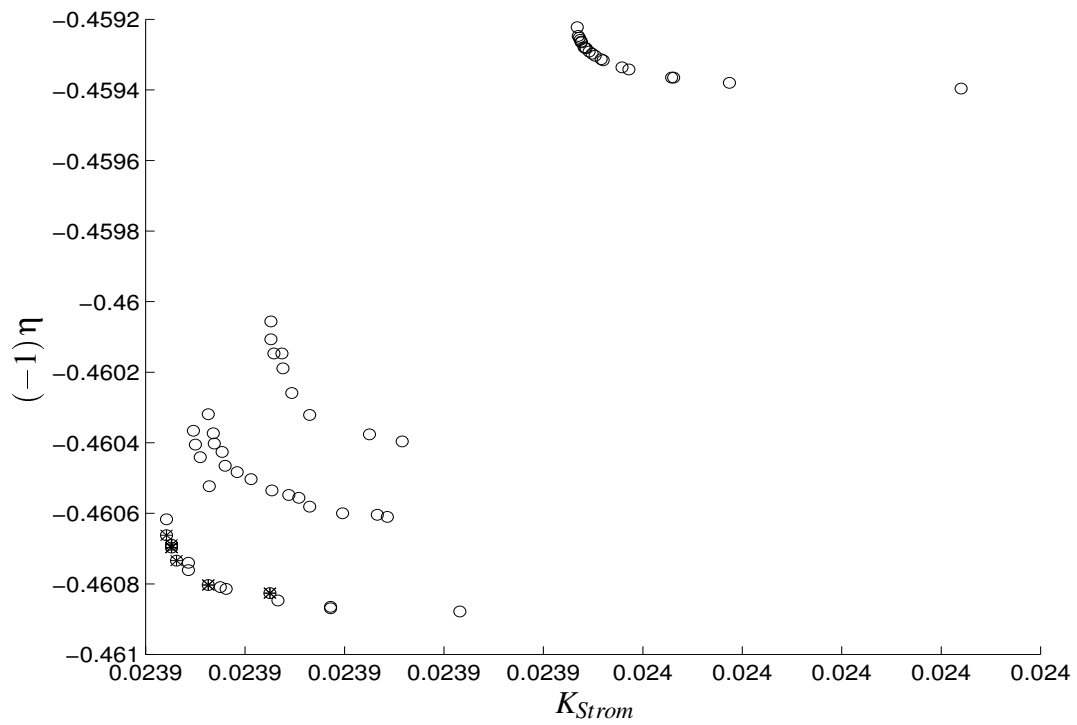
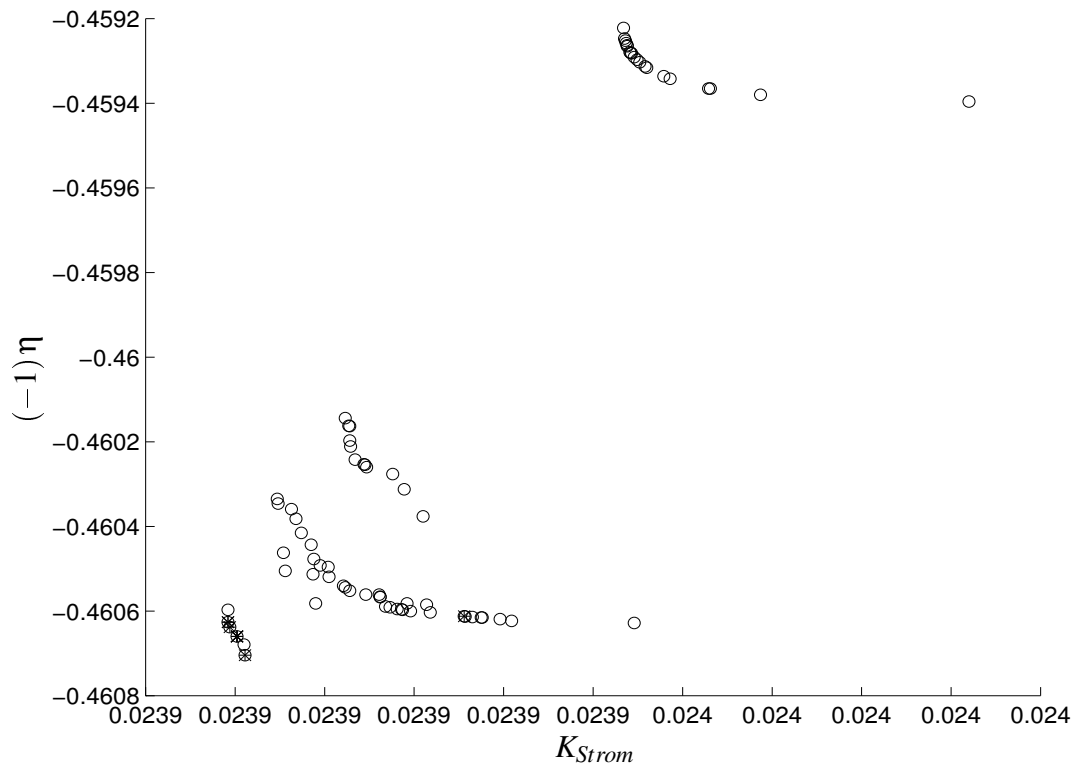
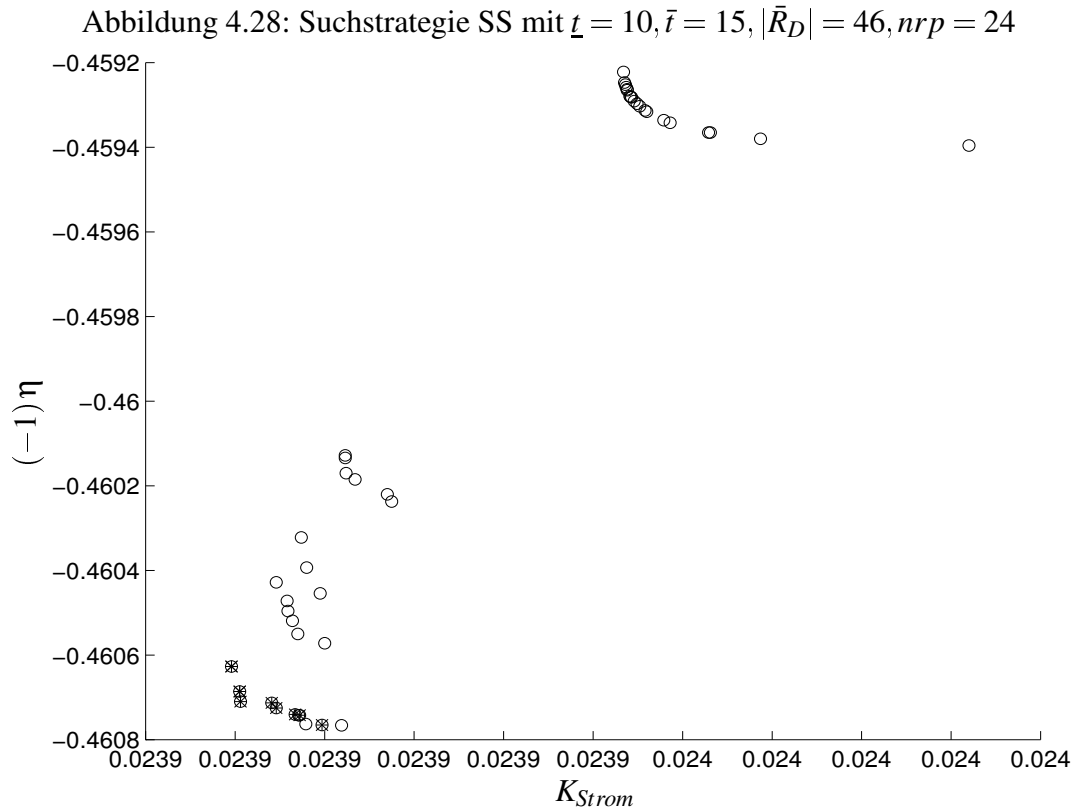
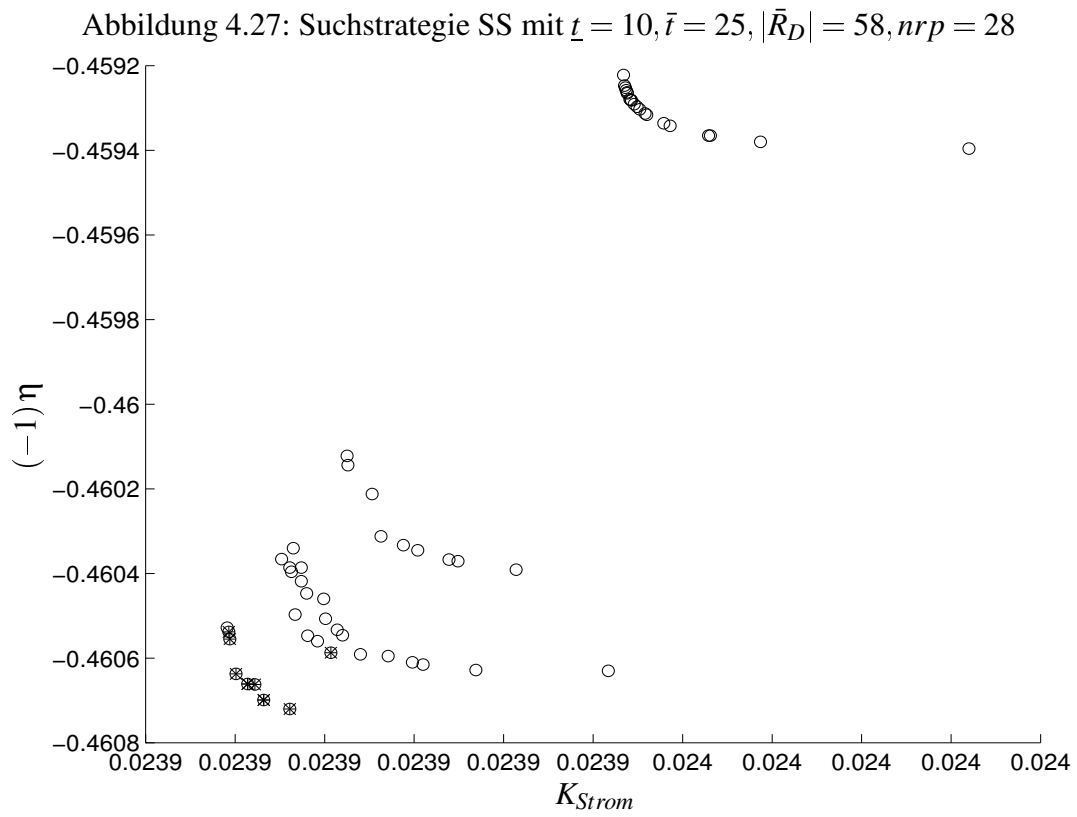


Abbildung 4.25: Suchstrategie SS mit $\underline{t} = 0, \bar{t} = 15, |\bar{R}_D| = 62, nrp = 26$ Abbildung 4.26: Suchstrategie SS mit $\underline{t} = 0, \bar{t} = 6, |\bar{R}_D| = 76, nrp = 25$ 



4.2 Testanwendungen

Die in den nachfolgenden Abschnitten beschriebenen Mehrzielprobleme sind diskret-kontinuierliche Zwei- und Dreizielminimierungsprobleme. Die Mengen $\bar{R}_D := \cup_{x_D \in D} \bar{E}^{x_D}$ seien jeweils bereits vorhandene numerische Näherungen der entsprechenden Mengen R_D , die Mengen $\bar{E}^{x_D}, x_D \in D$, numerische Approximationen der Mengen E^{x_D} , die irgendwie erzeugt wurden. Bei jedem Beispiel ist die Effizienzmenge in \bar{R}_D bereits bekannt. Außerdem ist die Menge \bar{R}_D jeweils von sehr großer Mächtigkeit.

Um die gute Leistung der Suchstrategien SS und PS auf \bar{R}_D (und somit auch auf R_D) zum Ausdruck zu bringen, führen wir einige Symbole ein:

$\underline{t} \in \mathbb{N}$	erste Iteration während einer Metropolis-Suche (NMA), die ein (vorläufiges) Suchergebnis liefern könnte
$\bar{t} \in \mathbb{N}$	maximal erlaubte Iterationszahl für eine Metropolis-Suche (NMA), $\bar{t} \geq \underline{t}$
$zg \in \mathbb{N}$	Anzahl der Listengenerationen $L_k, k \in \mathbb{N}$, die in Suchstrategie PS bis zum Abbruch des Suchlaufes erzeugt und bearbeitet wurden; Anzahl der Referenzpunkte aus R_D , für die in Suchstrategie SS je ein Problem (3.21) zu lösen war
$SOL \subseteq \bar{R}_D$	Menge der Lösungsalternativen, die in einem Lauf des getesteten Mehrzielverfahrens erzeugt wurde, vom Verfahren als paretooptimal eingestufte Punkte
$PO \subseteq \bar{R}_D$	tatsächliche Effizienzmenge $\in \bar{R}_D$
$FPO \subseteq \bar{R}_D$	Menge von Alternativen $\in \bar{R}_D$, die als "fast paretooptimal" definiert werden
$NPO \subseteq \bar{R}_D$	Menge von Alternativen $\in \bar{R}_D$, die als "nahe der Effizienz" definiert werden
Σ	$\frac{ PO \cap SOL + FPO \cap SOL + NPO \cap SOL }{ SOL }$ bzw. $\frac{ PO \cap L_0 + FPO \cap L_0 + NPO \cap L_0 }{ L_0 }$

$PO \cap FPO \cap NPO = \emptyset$ und wir sehen Alternativen aus NPO als "schlechter" an als die Alternativen aus FPO . Die Elemente beider Mengen gelten uns dennoch bereits als recht zufriedenstellend. Σ symbolisiert daher den Anteil an Punkten in SOL bzw. in L_0 , den man als nützlich und gut betrachten kann. Tabellen sollen nun anstelle von Grafiken die guten Testergebnisse, ausgedrückt in den Größen $|PO \cap SOL|/|SOL|$, $|FPO \cap SOL|/|SOL|$, $|NPO \cap SOL|/|SOL|$ und Σ , vor Augen führen.

Die Nachbarschaftsstruktur in jedem der Metropolis-Suchräume ist recht einfach konstruiert: Zu den Nachbarn eines Standpunktes gehören stets alle Punkte des Suchraumes ohne den Standpunkt selbst. Alle Generierungswahrscheinlichkeiten haben den Wert $\frac{1}{|S|-1}$. Die Symmetrie der Generierungswahrscheinlichkeiten sowie Aperiodizität und Irreduzibilität der Markov Kette, die für ein $c \in \mathbb{R}^+$ den IMA, dessen numerische Ausführung das jeweilige diskrete Optimierungsproblem bearbeitet, modelliert, sind daher offensichtlich erfüllt. In jedem Problem (3.21) waren min mit -1 , mid mit 0 und max mit 1 belegt. Die Kontrollparameter $c \in \mathbb{R}^+$, die die Akzeptanzwahrscheinlichkeiten für Zielwertverschlechterungen beim Lösen der verschiedenen Probleme (3.21) steuern, wurden von Metropolis-Suchlauf (NMA) zu Metropolis-Suchlauf variiert, wobei jedoch kein bestimmtes Akzeptanzniveau für Verschlechterungen besonders hervorstach. Je nach Wahl von c galten Akzeptanzwahrscheinlichkeiten für eine Verschlechterung von min auf mid bzw. mid auf max von etwa 0.01, 0.5 oder 0.95.

4.2.1 Testanwendung 1

4.2.1.1 Das Mehrzielproblem

$$\begin{aligned} \text{"min"}_{(x_C, x_D) \in R_D} F(x_C, x_D), F = \begin{pmatrix} f_1(x_C, x_D) \\ f_2(x_C, x_D) \end{pmatrix} := \begin{pmatrix} x_C \\ x_D - x_C^2 \end{pmatrix}, \\ \bar{R}_D := \{0, 1, \dots, 1000\} \times \{0, 1, \dots, 1000\}. \end{aligned} \tag{4.23}$$

R könnte z.B. die Menge $C \times D := [0; 1000] \times \{0, 1, \dots, 1000\}$ sein.

$\bar{C} \subseteq C$ sei definiert als $\bar{C} := \{0, 1, \dots, 1000\}$.

$\bar{E}^{\bar{x}_D} := \{(x_C, \bar{x}_D - x_C^2) | x_C \in \bar{C}\}$ stellt die numerisch errechnete Approximation der Menge $E^{\bar{x}_D}, \bar{x}_D \in D$, dar.

$|\bar{R}_D| = 1.002.001$ - der Suchraum ist also sehr groß.

Die Effizienzmenge von \bar{R}_D ist bekannt, nämlich $PO := \{(x_C, 0) | x_C \in \bar{C}\}$.

Die Menge FPO belegen wir mit $FPO := \{(x_C, x_D) | x_C \in \bar{C}, x_D \in \{1, \dots, 9\}\}$ und die Menge NPO mit $NPO := \{(x_C, x_D) | x_C \in \bar{C}, x_D \in \{10, \dots, 99\}\}$.

4.2.1.2 Testergebnisse

\underline{t}	0	90000	0	9000	0		
\bar{t}	100000	100000	10000	10000	1000		
$ SOL $	116	81	90	60	57	$ L_0 $	50
zg	5	5	5	4	4		
$ PO \cap SOL / SOL $	0.27	0.05	0.06	0.00	0.00	$ PO \cap L_0 / L_0 $	0.00
$ FPO \cap SOL / SOL $	0.68	0.21	0.22	0.02	0.02	$ FPO \cap L_0 / L_0 $	0.00
$ NPO \cap SOL / SOL $	0.05	0.72	0.63	0.38	0.35	$ NPO \cap L_0 / L_0 $	0.10
Σ	1.00	0.98	0.91	0.40	0.37	Σ	0.10

Tabelle 4.1: verschiedene Tests mit Suchstrategie PS

\underline{t}	0	90000	0	9000	0
\bar{t}	100000	100000	10000	10000	1000
$ SOL $	50	50	50	50	50
zg	266	164	164	75	77
$ PO \cap SOL / SOL $	0.14	0.02	0.00	0.00	0.00
$ FPO \cap SOL / SOL $	0.74	0.24	0.14	0.04	0.00
$ NPO \cap SOL / SOL $	0.12	0.56	0.74	0.20	0.20
Σ	1.00	0.82	0.88	0.24	0.20

Tabelle 4.2: verschiedene Tests mit Suchstrategie SS

4.2.2 Testanwendung 2

4.2.2.1 Das Mehrzielproblem

$$\begin{aligned}
 \text{"min"}_{(x_C, x_D) \in R_D} F(x_C, x_D), F &= \begin{pmatrix} f_1(x_C, x_D) \\ f_2(x_C, x_D) \end{pmatrix}, \\
 f_1(x_C, x_D) &:= x_C, \\
 f_2(x_C, x_D) &:= \begin{cases} x_D - x_C^4, & \text{falls } x_D \bmod 2 = 0, \\ x_D - 0.25x_C^2, & \text{falls } x_D \bmod 2 = 1, \end{cases} \\
 \bar{R}_D &:= \{0, 1, \dots, 1000\} \times \{0, 1, \dots, 1000\}.
 \end{aligned} \tag{4.24}$$

Wieder könnte R die Menge $C \times D := [0; 1000] \times \{0, 1, \dots, 1000\}$ sein.

$\bar{C} \subseteq C$ sei festgelegt als $\bar{C} := \{0, 1, \dots, 1000\}$.

$\bar{E}^{\bar{x}_D} := \{(x_C, \bar{x}_D - x_C^4) | x_C \in \bar{C}\}$ für $\bar{x}_D \in D$ mit $\bar{x}_D \bmod 2 = 0$,

$\bar{E}^{\bar{x}_D} := \{(x_C, \bar{x}_D - 0.25x_C^2) | x_C \in \bar{C}\}$ für $\bar{x}_D \in D$ mit $\bar{x}_D \bmod 2 = 1$.

$|\bar{R}_D| = 1.002.001$.

Die Effizienzmenge in \bar{R}_D ist $PO := \{(x_C, 0) | x_C \in \bar{C}\}$.

$FPO := \{(x_C, x_D) | x_C \in \bar{C}, x_D \in \{2, 4, 6, 8\}\} \cup (\{0, 1\} \times \{1, 3, 5, 7, 9\})$.

$NPO := \{(x_C, x_D) | x_C \in \bar{C}, x_D \in \{10, 12, \dots, 98\}\} \cup (\{0, 1\} \times \{11, 13, \dots, 99\})$.

4.2.2.2 Testergebnisse

\underline{t}	0	90000	0		
\bar{t}	100000	100000	10000		
$ SOL $	53	50	50	$ L_0 $	50
zg	4	5	5		
$ PO \cap SOL / SOL $	0.17	0.00	0.02	$ PO \cap L_0 / L_0 $	0.00
$ FPO \cap SOL / SOL $	0.53	0.10	0.10	$ FPO \cap L_0 / L_0 $	0.00
$ NPO \cap SOL / SOL $	0.30	0.60	0.54	$ NPO \cap L_0 / L_0 $	0.10
Σ	1.00	0.70	0.66	Σ	0.10

Tabelle 4.3: verschiedene Tests mit Suchstrategie PS

\underline{t}	0	90000	0
\bar{t}	100000	100000	10000
$ SOL $	50	50	50
zg	247	150	149
$ PO \cap SOL / SOL $	0.16	0.00	0.04
$ FPO \cap SOL / SOL $	0.44	0.08	0.06
$ NPO \cap SOL / SOL $	0.40	0.52	0.42
Σ	1.00	0.60	0.52

Tabelle 4.4: verschiedene Tests mit Suchstrategie SS

4.2.3 Testanwendung 3

4.2.3.1 Das Mehrzielproblem

$$\begin{aligned}
 \text{"min"}_{(x_C, x_D) \in R_D} F(x_C, x_D), F = \begin{pmatrix} f_1(x_C, x_D) \\ f_2(x_C, x_D) \\ f_3(x_C, x_D) \end{pmatrix} := \begin{pmatrix} (x_C)_1 \\ (x_C)_2 \\ x_D - (x_C)_1^2 - (x_C)_2^2 \end{pmatrix}, \quad (4.25) \\
 \bar{R}_D := (\{0, 1, \dots, 100\} \times \{0, 1, \dots, 100\}) \times \{0, 1, \dots, 100\}.
 \end{aligned}$$

Unter R kann man sich die Menge $C \times D := ([0; 100] \times [0; 100]) \times \{0, 1, \dots, 100\}$ mit $C := [0; 100] \times [0; 100]$ und $D := \{0, 1, \dots, 100\}$ vorstellen.

$\bar{C} \subseteq C$ sei definiert durch $\bar{C} := \{0, 1, \dots, 100\} \times \{0, 1, \dots, 100\}$.

$\bar{E}^{\bar{x}_D} := \{(x_C, \bar{x}_D - (x_C)_1^2 - (x_C)_2^2) \mid x_C \in \bar{C}\}$ für $\bar{x}_D \in D$.

$|\bar{R}_D| = 1.030.301$.

Die Effizienzmenge in \bar{R}_D entspricht der Menge $PO := \{(x_C, 0) \mid x_C \in \bar{C}\}$.

$FPO := \{(x_C, 1) \mid x_C \in \bar{C}\}$.

$NPO := \{(x_C, x_D) \mid x_C \in \bar{C}, x_D \in \{2, 3, \dots, 9\}\}$.

4.2.3.2 Testergebnisse

\underline{t}	0	90000	0		
\bar{t}	100000	100000	10000		
$ SOL $	94	56	56	$ L_0 $	50
zg	7	3	3		
$ PO \cap SOL / SOL $	0.26	0.05	0.05	$ PO \cap L_0 / L_0 $	0.00
$ FPO \cap SOL / SOL $	0.14	0.02	0.00	$ FPO \cap L_0 / L_0 $	0.00
$ NPO \cap SOL / SOL $	0.51	0.21	0.27	$ NPO \cap L_0 / L_0 $	0.04
Σ	0.91	0.28	0.32	Σ	0.04

Tabelle 4.5: verschiedene Tests mit Suchstrategie PS

\underline{t}	0	90000	0
\bar{t}	100000	100000	10000
$ SOL $	50	50	50
zg	164	82	81
$ PO \cap SOL / SOL $	0.30	0.06	0.04
$ FPO \cap SOL / SOL $	0.10	0.00	0.00
$ NPO \cap SOL / SOL $	0.48	0.20	0.30
Σ	0.88	0.26	0.34

Tabelle 4.6: verschiedene Tests mit Suchstrategie SS

4.2.4 Testanwendung 4

4.2.4.1 Das Mehrzielproblem

$$\begin{aligned}
 \text{"min"}_{(x_C, x_D) \in R_D} F(x_C, x_D), F &= \begin{pmatrix} f_1(x_C, x_D) \\ f_2(x_C, x_D) \\ f_3(x_C, x_D) \end{pmatrix}, \\
 f_1(x_C, x_D) &:= (x_C)_1, \\
 f_2(x_C, x_D) &:= (x_C)_2, \\
 f_3(x_C, x_D) &:= \begin{cases} x_D - (x_C)_1^4 - (x_C)_2^4, & \text{falls } x_D \bmod 2 = 0, \\ x_D - 0.25(x_C)_1^2 - 0.25(x_C)_2^2, & \text{falls } x_D \bmod 2 = 1, \end{cases} \\
 \bar{R}_D &:= (\{0, 1, \dots, 100\} \times \{0, 1, \dots, 100\}) \times \{0, 1, \dots, 100\}.
 \end{aligned} \tag{4.26}$$

Wiederum könnte R gleich der Menge $C \times D := ([0; 100] \times [0; 100]) \times \{0, 1, \dots, 100\}$ mit $C := [0; 100] \times [0; 100]$ und $D := \{0, 1, \dots, 100\}$ sein.

$\bar{C} \subseteq C$ sei festgelegt durch $\bar{C} := \{0, 1, \dots, 100\} \times \{0, 1, \dots, 100\}$.

$\bar{E}^{\bar{x}_D} := \{(x_C, \bar{x}_D - (x_C)_1^4 - (x_C)_2^4) \mid x_C \in \bar{C}\}$ für $\bar{x}_D \in D$ mit $\bar{x}_D \bmod 2 = 0$,

$\bar{E}^{\bar{x}_D} := \{(x_C, \bar{x}_D - 0.25(x_C)_1^2 - 0.25(x_C)_2^2) \mid x_C \in \bar{C}\}$ für $\bar{x}_D \in D$ mit $\bar{x}_D \bmod 2 = 1$.

$|\bar{R}_D| = 1.030.301$.

Die Effizienzmenge in \bar{R}_D ist $PO := \{(x_C, 0) \mid x_C \in \bar{C}\}$.

$FPO := \{(x_C, 1) \mid x_C \in \{(0, 0), (0, 1), (1, 0)\}\} \cup \{(x_C, 2) \mid x_C \in \bar{C}\}$.

$NPO := \{(x_C, x_D) \mid x_C \in \{(0, 0), (0, 1), (1, 0)\}, x_D \in \{3, 5, 7, 9\}\} \cup \{(x_C, x_D) \mid x_C \in \bar{C}, x_D \in \{4, 6, 8\}\}$.

4.2.4.2 Testergebnisse

\underline{t}	0		
\bar{t}	100000		
$ SOL $	127	$ L_0 $	50
zg	6		
$ PO \cap SOL / SOL $	0.20	$ PO \cap L_0 / L_0 $	0.02
$ FPO \cap SOL / SOL $	0.24	$ FPO \cap L_0 / L_0 $	0.02
$ NPO \cap SOL / SOL $	0.31	$ NPO \cap L_0 / L_0 $	0.08
Σ	0.75	Σ	0.12

Tabelle 4.7: verschiedene Tests mit Suchstrategie PS

\underline{t}	0
\bar{t}	100000
$ SOL $	50
zg	149
$ PO \cap SOL / SOL $	0.18
$ FPO \cap SOL / SOL $	0.04
$ NPO \cap SOL / SOL $	0.26
Σ	0.48

Tabelle 4.8: verschiedene Tests mit Suchstrategie SS

Kapitel 5

Resümee und Ausblicke

Der Leser hat nun erfahren,

- was es auf dem Gebiet der Vektoroptimierung für Lösungsansätze gibt und welchen Grundideen diese folgen,
- warum in dieser Arbeit eine Generierungstechnik entwickelt wird,
- warum für die diskret-kontinuierlichen Vektoroptimierungsprobleme, die in dieser Arbeit zu bearbeiten sind, erst noch eine Generierungstechnik zu entwickeln war,
- mit welcher Klasse von diskret-kontinuierlichen Vektoroptimierungsproblemen sich diese Arbeit befaßt,
- wie das hier entwickelte numerische Verfahren DKMOP (in den Versionen SS (SS1/SS2/SS3) und PS) zur Lösung diskret-kontinuierlicher Vektoroptimierungsprobleme funktioniert, warum es so konstruiert wurde und was man an mathematischen Konvergenzaussagen darüber machen kann,
- wie es in Beispielen auf konkrete Problemstellungen zugeschnitten werden kann,
- wie erfolgreich es in Beispielanwendungen arbeitet.

Nach dieser Fülle von präsentierten Informationen sollen abschließend die Vor- und Nachteile der Mehrzielmethode DKMOP mit Suchstrategien SS und PS zusammengefaßt werden: **Vorteile:**

- Das Verfahren liefert eine **repräsentative Teilmenge** oder Approximation einer repräsentativen Teilmenge **der Paretooptima**, ganz unabhängig von der Form von R und $F(R)$.
- Die Lösungstechnik vollführt abwechselnd diskrete und kontinuierliche Suchschritte. Die mit jeder erzeugten diskreten Variation angesprochenen kontinuierlichen Subprobleme sind nach Voraussetzung zweimal stetig differenzierbar, man verlangt von ihnen jedoch keine Konvexität der Alternativen- oder Kriterienmengen. Unter **Ausnutzung aller zur Verfügung stehenden Ableitungsinformationen** kann sie ein guter kontinuierlicher Mehrzieloptimierer (natürlich eine Generierungstechnik) numerisch lösen.

Die diskreten Suchschritte auf R_D **kommen mit lediglich den Zielwertvektoren zu-recht.**

Fazit: Der Lösungsansatz nutzt so viel Zielfunktionsinformation wie möglich (Ableitungsinformation in den kontinuierlichen Subproblemen) und kommt dort, wo nur Zielwertinformation zur Verfügung steht (auf R_D), mit dieser wenigen Zielfunktionsinformation zurecht.

- R_D , wo die diskreten Suchschritte vollführt werden, weist die Struktur $R_D := \cup_{x_D \in D} E^{x_D}$ auf. Innerhalb der Mengen $E^{x_D}, x_D \in D$, gibt es keine paarweise Dominanz. Diese **spezielle Struktur nutzen die Suchstrategien SS und PS aus, um die Suchräume** der in Folge zu bearbeitenden diskreten skalaren Minimierungsprobleme (Suche nach dominierendem Punkt für Referenzalternative) **mehr und mehr zu verkleinern.**
- Der **Entscheidungsträger muß weder vor noch während der Anwendung der Lösungstechnik Präferenzen innerhalb der Ziele festlegen**, da er es mit einer Generierungstechnik zu tun hat. Das Lösungsverfahren bringt ihm erst den nötigen Informationsstand für Präferenzfestlegungen, da es ihm eine bestmögliche Idee vom Aussehen der Effizienzmeng in R erarbeitet.
- Für DKMOP mit Suchstrategien SS und PS ist grundsätzlich jedes Paretooptimum mit positiver Wahrscheinlichkeit erreichbar.
- Es sind gewisse Aussagen über die Wahrscheinlichkeiten, mit denen effiziente Alternativen gefunden werden, möglich.
- Das Verfahren bewältigt sehr große Suchräume.
- Die numerische Technik zur Lösung diskret-kontinuierlicher Vektoroptimierungsprobleme folgt einer simplen Idee und ist leicht zu implementieren.

Nachteile:

- Als stochastisches Verfahren kann die Methode DKMOP kein Paretooptimum mit Wahrscheinlichkeit eins finden.
- Einerseits zählt der Lösungsansatz zu seinen Vorteilen, daß er große Suchräume bewältigen kann, andererseits erhebt sich bei kleiner Mächtigkeit $|D|$ und geringer Mächtigkeit der numerischen Näherung \bar{E}^{x_D} für E^{x_D} bei jedem generierten $x_D \in D$ die Frage, ob nicht etwa die komplette Erzeugung von $\cup_{x_D \in D} \bar{E}^{x_D}$ mit anschließender vollständiger Enumeration darauf mindestens genauso gute Arbeit leisten kann.

Der potenzielle Anwender des in dieser Arbeit entwickelten numerischen Verfahrens zur Lösung diskret-kontinuierlicher Vektoroptimierungsprobleme mag sich mit Hilfe der vorhergehenden Pro- und Kontra-Gegenüberstellung darüber klar werden, wie dieses Verfahren einzuschätzen ist. Er kann sich auch Variationen und Verbesserungen überlegen. Beispielsweise sind die Suchstrategien SS und PS nicht nur auf Anwendung des Metropolis Algorithmus festgelegt. Man könnte auch versuchen, andere diskrete skalare Optimierungsverfahren für die Suchstrategien einzusetzen und dafür mathematische Aussagen über Suchverhalten und Konvergenz erarbeiten.

Literaturverzeichnis

- [1] Emile Aarts and Jan Korst. *Simulated Annealing and Boltzmann Machines - A Stochastic Approach to Combinatorial Optimization and Neural Computing*. Wiley - Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons Inc., Chichester, New York, Brisbane, Toronto, Singapore, 1989.
- [2] R.M. Beeson and W.S. Meisel. The optimization of complex systems with respect to multiple criteria. *IEEE Systems, Man and Cybernetics, Group Annual Symposium Record, IEEE*, pages 144–149, 1971.
- [3] R. Benayoun, J. de Montgolfier, J. Tergny, and O. Larichev. Linear programming with multiple objective functions: Step method (STEM). *Mathematical Programming*, 1(3):366–375, 1971.
- [4] R. Benayoun, O. Larichev, J. de Montgolfier, and J. Tergny. Linear programming with multiple objective functions: The method of constraints. *Automatic and Remote Control*, 32(8):1257–1264, 1971.
- [5] R. Benayoun, J. Tergny, and D. Keuneman. Mathematical programming with multi-objective function: A solution by p.o.p (progressive orientation procedure). *METRA*, 9(2):279–299, 1970.
- [6] H.P. Benson. An improved definition of proper efficiency for vector optimization with respect to cones. *Journal of Mathematical Analysis and Applications*, 71:232–241, 1979.
- [7] R.G. Benson. *Interactive Multiple Criteria Optimization Using Satisfactory Goals*. PhD thesis, University of Iowa, 1975.
- [8] J.M. Borwein. On the existence of pareto efficient points. *Mathematics of Operations Research*, 8(1):64–73, 1983.
- [9] Vira Chankong and Yakov Y. Haimes. *Multiobjective Decision Making - Theory and Methodology*. North Holland, New York, 1983.
- [10] A. Charnes and A. Cooper. Goal programming and multiple objective optimization - part 1. *European Journal of Operational Research*, 1(1):39–54, 1977.
- [11] Jared L. Cohon. *Multiobjective Programming and Planning*. Academic Press, New York San Francisco London, 1978.
- [12] H. W. Corley. A new scalar equivalence for pareto optimization. *IEEE Transactions of Automatic Control*, 25:829–830, 1980.

- [13] Indraneel Das and John Dennis. Normal-boundary intersection: An alternate method for generating pareto optimal points in multicriteria optimization. Technical report, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA 23681-0001, 1996.
- [14] Indraneel Das and John Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization. *Structural Optimization*, 14(1):63–69, 1997.
- [15] Matthias Ehrgott. *Lexicographic Max-Ordering - A Solution Concept for Multicriteria Combinatorial Optimization*. PhD thesis, Universität Kaiserslautern, 1995.
- [16] J. Ester and B. Schwartz. Ein erweitertes Effizienztheorem. *Mathematische Operationsforschung und Statistik, Serie Optimization*, 14(3):331–342, 1983.
- [17] P. Farquhar. *A Survey of Multiattribute Utility Theory and Applications*, pages 59–90. In [55], 1977.
- [18] A. Feinberg. *An Experimental Investigation of an Interactive Approach for Multi-criterion Optimization, with an Application to Academic Resource Allocation*. PhD thesis, University of California, Los Angeles, 1972.
- [19] William Feller. *An Introduction to Probability Theory and its Applications*, volume 1 of *Wiley Series in Probability and Mathematical Statistics*. John Wiley & Sons, Inc, New York, London, Sidney, 1968.
- [20] P.C. Fishburn. Lexicographic orders, utilities and decision rules: A survey. *Management Science*, 20(11):1442–1471, 1974.
- [21] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. *Proceedings of the Fifth International Conference on Genetic Algorithms (S. Forrest, ed.)*, pages 416–423, 1993.
- [22] Wolfgang Geber. *Diskrete Vektoroptimierung*, volume 591 of *Europäische Hochschulschriften: Reihe 5, Volks- und Betriebswirtschaft*. Verlag Peter Lang GmbH, Frankfurt am Main, 1985.
- [23] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Co., Reading, Mass., 1989.
- [24] Alfred Göpfert and Reinhard Nehse. *Vektoroptimierung - Theorie, Verfahren und Anwendungen*. Teubner, Leipzig, 1990.
- [25] Y. Haimes. Integrated system identification and optimization. *Control and Dynamic Systems: Advances in Theory and Applications*, 10:435–518, 1973.
- [26] Y.Y. Haimes and W.A. Hall. Multiobjectives in water resources systems analysis: The surrogate worth trade-off method. *Water Resources Research*, 10(4):615–623, 1974.
- [27] Y.Y. Haimes, W.A. Hall, and H.T. Freedman. *Multiobjective Optimization in Water Resources Systems, The Surrogate Worth Trade-Off Method*. Elsevier Scientific, New York, 1975.

- [28] P. Hajela and C. Y. Lin. Genetic search strategies in multicriterion optimal design. *Structural Optimization*, 4:99–107, 1992.
- [29] M.I. Henig. Proper efficiency with respect to cones. *Journal of Optimization Theory and Applications*, 36(3):387–407, 1982.
- [30] Claus Hillermeier. *Eine Homotopiemethode zur Vektoroptimierung*. Habilitationsschrift, Technische Universität München, Februar 1999.
- [31] J. Horn and N. Nafpliotis. Multiobjective optimization using the niched pareto genetic algorithm. Technical report, IlliGAI Report 93005, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, 1993.
- [32] Yuda Hu and Erjiang Sun. The comparison-number method for finding major optimal solutions and major efficient solutions of the discrete multiobjective programming. *Systems Science and Mathematical Sciences*, 6(3):284–289, 1993.
- [33] G.P. Huber. Multi-attribute utility models: A review of field and field-like studies. *Management Science*, 20(10):1393–1402, 1974.
- [34] Steffen Hüster. *Stochastische Algorithmen zur Optimierung der Auslegung einer technischen Anlage*. Masterthesis, Universität Augsburg, November 1998.
- [35] Ching-Lai Hwang and Abu Syed Md. Masud. *Multiple Objective Decision Making - Methods and Applications*, volume 164 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, Berlin Heidelberg New York, 1979.
- [36] J.P. Ignizio. *Goal Programming and Extensions*. Lexington Books, Massachusetts, 1976.
- [37] Y. Ijiri. *Management Goals and Accounting for Control*. North Holland, Amsterdam, 1965.
- [38] R.L. Keeney. Utility functions for multiattributed consequences. *Management Science*, 18(5):276–287, 1972.
- [39] R.L. Keeney and H. Raiffa. *Decision with Multiple Objectives: Preferences and Value Tradeoffs*. Wiley, New York, 1976.
- [40] H. Kuhn and A. Tucker. Nonlinear programming. In J. Neyman, editor, *Proc. Berkeley Symp. Math. Statist. Probability*, volume 2, pages 481–492, Berkeley, 1951. University of California Press.
- [41] F. Kursawe. A variant of evolution strategies for vector optimization. *Lecture Notes in Computer Science: Parallel Problem Solving from Nature. 1st Workshop, PPSNI (H. P. Schwefel and R. Männer, eds.)*, 496:193–197, 1991.
- [42] U. Lampe. Dualität und eigentliche Effizienz in der Vektoroptimierung. *Seminarberichte Humboldt-Universität Berlin*, 37:45–54, 1981.
- [43] S.M. Lee. *Goal Programming for Decision Analysis*. Auerbach Publishers, Philadelphia, Pennsylvania, 1972.

- [44] E.R. Lieberman. *Multiobjective Programming in the USSR*. Academic Press, INC, San Diego, 1991.
- [45] J. G. Lin. Multiple objective problems: Pareto-optimal solutions by method of proper equality constraints. *IEEE Transactions on Automatic Control*, 21:641–650, 1976.
- [46] J. G. Lin. *Three methods for determining Pareto-optimal solutions of multiple objective optimization problems*. Plenum, New York, 1976.
- [47] J. G. Lin. Proper inequality constraints and maximization of index vectors. *Journal of Optimization Theory and Applications*, 21:505–521, 1977.
- [48] S. Marglin. *Public Investment Criteria*. MIT Press, Cambridge, Massachusetts, 1967.
- [49] K. Miettinen. *On the Methodology of Multiobjective Optimization with Applications*. PhD thesis, University of Jyväskylä, Faculty of Mathematics and Natural Sciences, Jyväskylä, Finland, 1994.
- [50] L.I. Polishchuk. Piecewise linear approximation of the pareto boundary of convex bi-criteria problems. In E.L. Berlyand and B.G. Mirkin, editors, *Modeli i metody issledovaniya ekonomicheskikh sistem*, pages 108–116, Novosibirsk, 1979. USSR Academy of Sciences, Siberian Branch.
- [51] J.D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 93–100, 1985.
- [52] Stefan Schäffler, Reinhart Schultz, and Klaus Weinzierl. A stochastic method for the solution of unconstrained vector optimization problems. Technical report, Siemens AG, Corporate Research, 1999.
- [53] H.P. Schwefel. *Numerical Optimization of Computer Models*. John Wiley & Sons, Great Britain, 1981.
- [54] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. Technical report, Department of Mechanical Engineering, Indian Institute of Technology, Kanpur, India, 1993.
- [55] M.K. Starr and M. Zeleny (eds.). *Multiple Criteria Decision Making*. North Holland, New York, 1977.
- [56] R. E. Steuer. *Multiple Criteria Optimization: Theory, Computation and Applications*. John Wiley & Sons Inc., 1986.
- [57] R. E. Steuer. *Trends in Interactive Multiple Objective Programming*, volume 337 of *Lecture Notes in Economics and Mathematical Systems*. Springer Verlag, Berlin, Heidelberg, 1989.
- [58] R. E. Steuer and E. U. Choo. An interactive weighted tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26(3):326–344, 1983.

- [59] R.E. Steuer. *An Interactive Multiple Objective Linear Programming Procedure*, pages 225–239. In [55], 1977.
- [60] R.E. Steuer and A.T. Schuler. An interactive multiple objective linear programming approach to a problem in forest management. *Operations Research*, 26(2):254–269, 1978.
- [61] G. Timmel. Statistische Suchverfahren zur Bestimmung von Kompromißlösungen. *ZKI - Informationen*, 2:17–31, 1978.
- [62] G. Timmel. *Ein globales stochastisches Suchverfahren zur Bestimmung der Kompromißmenge bei statischen Optimierungsaufgaben mit mehrfacher Zielstellung*. PhD thesis, TH Karl-Marx-Stadt, Karl-Marx-Stadt (Chemnitz), FDR, 1979.
- [63] G. Timmel. Ein stochastisches Suchverfahren zur Bestimmung der optimalen Kompromißlösungen bei statischen polykriteriellen Optimierungsaufgaben. *Wiss. Z. TH Illmenau*, 6:159–174, 1980.
- [64] G. Timmel. Modifikation eines statistischen Suchverfahrens der Vektoroptimierung. *Wiss. Z. TH Illmenau*, 5:139–148, 1982.
- [65] J. Wallenius. Interactive multiple criteria decision methods: An investigation and approach. Technical report, The Helsinki School of Economics, Helsinki, 1975.
- [66] R. E. Wendel and D.N. Lee. Efficiency in multiobjective optimization problems. *Mathematical Programming*, 12:406–515, 1977.
- [67] D.V. Winterfeldt and G.W. Fischer. *Multiattribute Utility Theory: Models and Assessment Procedures*, pages 47–85. D. Reidel Pub. Co., Boston, 1975.
- [68] P. Yu. A class of decisions for group decision problems. *Management Sciences*, 19:936, 1973.
- [69] L. Zadeh. Optimality and non-scalar-valued performance criteria. *IEEE Transactions Automatic Control*, 8(59), 1963.
- [70] M. Zeleny. A concept of compromise solutions and the method of the displaced ideal. *Comput. Operations Research*, 1:479, 1974.
- [71] M. Zeleny. *Linear Multiobjective Programming*. Springer Verlag, Berlin and New York, 1974.
- [72] S. Zionts and J. Wallenius. An interactive programming method for solving the multiple criteria problem. *Management Science*, 22(6):652–663, 1976.

Index

- Alternativenmenge, **1**
 diskret, **1**
 diskret-kontinuierlich, **1, 54**
 kontinuierlich, **1**
- Analyst, **11**
- Anzapfung, **103**
- bedingte Wahrscheinlichkeit, **65**
- Dampferzeuger, **99**
- Dominanz, **3, 55**
 dominiert, **3, 4**
 inferior, **4**
- Dvoretzkys Konvergenzbedingungen, **40**
- Effizienz, **5, 55**
 effizient, **5**
 Effizienzmenge, **7**
 funktionaleffizient, **5**
 lokal funktionaleffizient, **5**
 lokal paretooptimal, *siehe* lokal funktionaleffizient
 Menge aller Paretooptima, *siehe* Effizienzmenge
 nichtdominiert, *siehe* effizient
 nichtinferior, *siehe* effizient
 paretooptimal, **5, 14**
 Paretooptimalität, **5**
 relativ effizient, **38**
- eigentlich (funktional)effizient nach Geoffrion, **9**
 streng eigentlich effizient, **9**
- Entscheidungsbaum, **105**
 Entscheidungsknoten, **105**
 Verzweigungsknoten, **105**
- Entscheidungsträger, **1**
- Ersatzwert Trade-Off Methode, **19**
- Evolutionäre Algorithmen, *siehe* Genetische Algorithmen
- Evolutionäre Strategien, **46**
- fossil befeuertes Kraftwerk, **99**
 Betriebskosten, **107**
 Brennstoffkosten, **107**
 Gesamtwirkungsgrad, **107**
 Investitionskosten, **107**
 spezifische Stromerzeugungskosten, **107**
- Generator, **99**
- Generierungstechniken, *siehe* Methoden mit a posteriori Präferenzäußerung, **21**
- Genetische Algorithmen, **45**
 Building Block, **45**
 Building Block Hypothese, **45**
 Chromosomen, **45**
 Fitness Wert, **45, 47, 49**
 Gen, **45**
 Kreuzung, **45**
 Nische, **47**
 Permutation, **45**
 Reproduktion, **45**
 Schema, **45**
 Sharing, **48**
 Fitness Wert nach Sharing, **48**
 Sharing Funktion, **48**
 Sharing Parameter, **48**
 Spezie, **46**
- Genetischer Algorithmus mit Nichtdominanz-Sortierung, **49**
- Genetischer Mehrzielalgorithmus, **49**
- Grädigkeit, **102**
- Hybrider Wichtungs- und Nebenbedingungsansatz, **25**
- Hyperbel Methode, **26**

- ideale Alternative, 3
 Idealpunkt, 16
 Indifferenzkurve, **13**, 31
 Interaktive ε -Gitter Methode, 21
 Interaktive Gewichtete Tchebycheff Prozedur, 20
 interaktive Methoden, *siehe* Methoden mit fortschreitender Präferenzäußerung
 iterative präferenzorientierte Techniken, *siehe* Methoden mit fortschreitender Präferenzäußerung
- Kompromißlösung, 11
 Kondensator, 99
 kontinuierlich-effizient, 62
 konvexe Hülle, 8
 Kriterienmenge, 2
 Kuhn-Tucker Bedingungen, **21**, 22, 29
- Lexikographische Methode, 15
 lokales Optimum, 5
- marginale Substitutionsrate, 13
 Markov Kette, 65, **65**, 73, 84
 Übergangsmatrix, 66, 73
 Übergangswahrscheinlichkeit, **65**, 66
 Anfangszustand, 66, **66**, 69
 aperiodisch, 69, **69**, 73, 119
 ergodisch, 69, **69**
 höhere Übergangswahrscheinlichkeiten, 66
 irreduzibel, 69, **69**, 73, 119
 mittlere Wiederkehrzeit, **68**
 stationär, *siehe* zeithomogen
 stationäre Verteilung, 69, **69**, 73, **73**, 75, 76, 82, 119
 Wahrscheinlichkeitsverteilung der Anfangszustände, 66
 zeithomogen, 65, **65**, 70, 73
 Zustand, 66, **67**, 69
 aperiodisch, **67**
 ergodisch, **68**, 69
 null-rekurrent, 68, **68**
 Periode, 67
 periodisch, 67
 rekurrent, 68, **68**
 transient, 68, **68**
- Markov Prozeß, *siehe* Markov Kette
- Mehrzielfunktion, 2
 Mehrzieloptimierung, *siehe* Vektoroptimierung
 Menge der Entscheidungsvariablen, *siehe* Alternativenmenge
 Methode der ε -Schranken (Generierungstechnik), **23**, 25
 Methode der ε -Schranken (Methode mit a posteriori Präferenzäußerung), 14
 Methode der beschränkten Ziele, 14
 Methode der Effizienzmengenschätzung, 30
 Methode der gewichteten L_p -Metrik, 25
 Methode der Gleichungsnebenbedingungen, 24
 Methode der sich anpassenden Suche, 37
 Methode der Vergleichszahlen, 35
 Hocheffizienz, 35
 Hooptimalität, 35
 Vergleichszahl, 36
 Methode der zufriedenstellenden Zielniveaus, 19
 Methode des globalen Kriteriums, **16**, 25
 Methode von Geoffrion, 19
 Methode von Steuer, 20
 Methode von Zions-Wallenius, 20
 Methoden mit a posteriori Präferenzäußerung, **21**, 51
 deterministisches Suchen, 21
 diskretes stochastisches Suchen, 45
 kontinuierliches stochastisches Suchen, 37
 Methoden mit a priori Präferenzäußerung, **12**, 50
 Methoden mit fortschreitender Präferenzäußerung, **18**, 50
 Methoden mit expliziter Trade-Off Information, 19, **19**
 Methoden mit impliziter Trade-Off Information, **19**, 20
 Metropolis Algorithmus, 64, 65, 70, **70**
 Übergangsmatrix, 73
 Übergangswahrscheinlichkeit, 72
 Akzeptanzmatrix, 73
 Akzeptanzwahrscheinlichkeit, 71
 Generierungsmatrix, 73
 Generierungswahrscheinlichkeit, 71

- idealisierte Metropolis Algorithmus, 65, 70, **70**, 73
- IMA, *siehe* idealisierte Metropolis Algorithmus
- Kontrollparameter, 72, **72**, 73
- Nachbar, **70**, **71**, 114
- Nachbarschaft, 70, **71**
- Nachbarschaftsstruktur, **71**, 118
- NMA, *siehe* numerischer Metropolis Algorithmus
- numerischer Metropolis Algorithmus, 70
- Sammelqualitäten, 77
- n-dimensionale Brownsche Bewegung, 43
- Nebenbedingungsfunktion, 2
- Niched Pareto GA, 50
- Nischenzählung, 50
- pareto domination tournaments, 50
- Vergleichsmenge, 50
- nicht-iterative präferenzorientierte Techniken, *siehe* Methoden mit a priori Präferenzäußerung
- NISE, *siehe* Methode der Effizienzmengenschätzung, 33
- Normalen-Rand Intersektion, 27
- konvexe Hülle der individuellen Minima, 27
- Nutzenfunktionen, 12
- Nutzenfunktionsmethoden, 12
- Optimierung mit Zielvorgaben, 16
- parametrische Methode, *siehe* Wichtungsmethode
- Pareto-basierte Fitnessbewertung, 48
- Rand, 7
- Referenzziel, 14
- schwach (funktional)effizient, *siehe* schwach effizient
- Speisewasser, 100
- Speisewasserpumpe, 99
- Speisewasservorwärmer, 100
- Enthitzer, 101
- Kondensationsvorwärmer, 101
- Kondensatkühler, 101
- Speisewasservorwärmung, 100
- STEM, *siehe* STEP-Methode
- STEP-Methode, 20
- stochastische Itô-Differentialgleichungen, 43
- stochastische Matrix, 66
- stochastische Stoppzeit, 43
- stochastischer Prozeß, 43
- Stochastisches Suchen nach Timmel, 38
- Stochastisches Verfahren zur Lösung von Vektroptimierungsproblemen ohne Nebenbedingungen, 41
- Stückweise Lineare Approximation, 33
- Suche auf Alternativenpaaren, 56
- erster Partner, 56, 57
- Referenzalternative, *siehe* Referenzpunkt, 84
- Referenzpunkt, **56**, 57, 64
- stochastischer Superprozeß, 83, **84**, **87**, **90**, 92
- zweiter Partner, 57
- Suchrichtung, 37, 40
- Summierung der gewichteten Zielfunktionen, **13**, 29–31, 47
- Trade-Off, 18
- partieller Trade-Off, 18
- vollständiger Trade-Off, 18
- Turbine, 99, 100
- unbedingte Wahrscheinlichkeit, 67
- VEGA, *siehe* Vektorwertiger Genetischer Algorithmus
- Vektroptimierung, 1
- diskret-kontinuierliches Problem, **54**, 61
- kontinuierliches Subproblem, 60, **60**, 62, 113
- Vektroptimierungsproblem, 2
- Vektorwertiger Genetischer Algorithmus, 46
- Vorwärmstrecke, 100
- Wasser-/Dampfkreislauf, 99
- wesentlich (funktional)effizient, 8
- Wichtungsansatz, 47
- Wichtungsmethode, **21**, 25
- Wiener Maß, 43

Wirkungsgrad eines Dampfkraftwerkes,
100

Zielfunktion, 2

Zielraum, *siehe* Kriterienmenge

Zufallsvariable, 43, 65

 diskrete Zufallsvariable, 65

 Realisierung, 66

zulässiger Bereich, *siehe* Alternativenmen-
ge