Augmented Lagrangian and Proximal Methods for Constrained Structured Optimization

Alberto De Marchi

Vollständiger Abdruck der von der Fakultät für Luft- und Raumfahrttechnik der Universität der Bundeswehr München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Gutachter: 1. Prof. Dr. rer. nat. Matthias Gerdts
2. Prof. Dr. rer. nat. Christian Kirches

Die Dissertation wurde am 1.2.2021 bei der Universität der Bundeswehr München eingereicht und durch die Fakultät für Luft- und Raumfahrttechnik am 27.5.2021 angenommen.
Die mündliche Prüfung fand am 14.6.2021 statt.

Alberto De Marchi
Augmented Lagrangian and Proximal Methods
for Constrained Structured Optimization
MMXXI

DOI: 10.5281/zenodo.4972536

License
This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/ or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Colophon
Document build: 29.06.2021 20:10:01 | 48°14′48.2″ N 12°30′56.4″ E

Typeset by the author using pdfTEX 3.14159265-2.6-1.40.18,
TeX Live 2017/Debian, in TEXstudio 3.1.2,
over Linux Ubuntu 18.04.5 LTS,
on a Dell XPS 13,
by hand.

Augmented Lagrangian and Proximal Methods for Constrained Structured Optimization

Alberto De Marchi

Abstract

This thesis aims at investigating and developing numerical methods for finite dimensional constrained structured optimization problems. These provide a modeling framework for a variety of applications, as they offer a simple yet expressive language to formulate a broad class of problems. An algorithm is proposed that interlaces proximal methods and the augmented Lagrangian scheme. Relying on theoretical results, convergence guarantees are established for nonconvex problems. The inner subproblems can be solved by any method for structured optimization and the overall algorithm can be made matrix-free. Illustrative examples show the benefits of constrained structured programs as a modeling tool and of a careful problem formulation. When tested and compared on small to medium-size nonlinear programming benchmark problems, the proposed method prove competitive against a state-of-the-art solver.

The proposed framework is adopted in the context of switching time optimization for constrained mixed-integer optimal control with switching costs. We describe the reformulation as constrained structured programs via the cardinality function, and discuss possible extensions to deal with more general problems. Then, we prove that this formulation satisfies the assumptions underlying the proximal augmented Lagrangian algorithm. Numerical examples show the filtering action of switching costs, which rules out chattering solutions.

Finally, we develop a primal-dual Newton-type proximal method for convex quadratic programming. This is based on the proposed proximal augmented Lagrangian framework and weaves together the proximal point algorithm and a damped semismooth Newton's method. The outer proximal regularization yields a numerically stable method, and we interpret the proximal operator as the unconstrained minimization of the primal-dual proximal augmented Lagrangian function. The inner tailored Newton's scheme is fast, the linear systems are always solvable, and exact linesearch can be performed. The method handles degenerate problems, provides a mechanism for infeasibility detection, and exploits warm starting, while requiring only convexity. Numerical results against full-fledged solvers demonstrate our method is robust and efficient.

All proposed algorithms are implemented in software packages that allow for the generic, efficient solution of problems using the methods developed in this thesis.

Zusammenfassung

Das Ziel der vorliegenden Arbeit ist die Untersuchung und Entwicklung numerischer Methoden für endlichdimensionale, beschränkte, strukturierte Optimierungsprobleme. Diese bieten einen Modellierungsrahmen für eine Vielzahl von Anwendungen, da sie eine einfache, aber ausdrucksstarke Notation zur Formulierung einer breiten Klasse von Problemen bieten. Es wird ein Algorithmus vorgeschlagen, der proximale Methoden und das erweiterte Lagrange-Schema kombiniert. Für nicht konvexe Probleme werden auf Basis theoretischer Ergebnisse Kriterien, die die Konvergenz garantieren, festgelegt. Die inneren Teilprobleme können mit beliebigen Methoden der strukturierten Optimierung gelöst werden, und der Gesamtalgorithmus kann frei von Matrizen formuliert werden. Mittels anschaulicher Beispiele werden die Vorteile von beschränkten, strukturierten Programmen als Modellierungswerkzeug und einer sorgfältigen Problemformulierung veranschaulicht. Beim Testen und Vergleichen an Benchmarkproblemen der nichtlinearen Programmierung von kleiner bis mittlerer Größe erweist sich die vorgeschlagene Methode als kompetitiv gegenüber einem state-of-the-art Solver.

Die vorgeschlagene Struktur wird im Zusammenhang mit der Optimierung der Schaltzeit für eine eingeschränkte gemischt-ganzzahlige optimale Steuerung mit Schaltkosten übernommen. Wir beschreiben die Umformulierung als ein beschränktes, strukturiertes Programm mittels der Kardinalitätsfunktion und diskutieren mögliche Erweiterungen, um allgemeinere Probleme zu lösen. Dann zeigen wir, dass diese Formulierung die Annahmen erfüllt, die dem proximal erweiterten Lagrange-Algorithmus zugrunde liegen. Numerische Beispiele zeigen die Filtereigenschaft von Schaltkosten, die oszillierende Lösungen ausschließt.

Schließlich entwickeln wir eine proximale Primal-Duale-Newton-Methode für die konvexe quadratische Programmierung. Diese basiert auf dem vorgeschlagenen proximalen Augmented-Lagrange Verfahren und verknüpft den Proximalpunkt-Algorithmus mit einer gedämpften Semismooth-Newton-Methode. Die äußere proximale Regularisierung ergibt eine numerisch stabile Methode und wir interpretieren den proximalen Operator als die uneingeschränkte Minimierung der primal-dualen proximalen, erweiterten Lagrange-Funktion. Das innere maßgeschneiderte Newtonsche Schema ist schnell, die linearen Systeme sind immer lösbar, und es kann eine exakte Liniensuche durchgeführt werden. Die Methode behandelt entartete Probleme, bietet einen Mechanismus zur Erkennung von Unlösbarkeit und nutzt den Warmstart, während nur Konvexität erforderlich ist. Numerische Ergebnisse im Vergleich mit etablierten Problemlösern zeigen, dass unsere Methode robust und effizient ist.

Alle vorgeschlagenen Algorithmen sind in Softwarepaketen implementiert, die die generische und effiziente Lösung von Problemen mithilfe der in dieser Arbeit entwickelten Methoden ermöglichen.

Preface

Mathematical optimization is usually associated with the analysis and characterization of a problem and the development of suitable methods for solving it. However, different formulations of the same problem, although mathematically equivalent, can lead to surprisingly different paths to a solution. We argue in this thesis that formulating a problem impacts the way we can solve it, and thus that modeling is a key part of mathematical optimization. Constrained structured programming offers a versatile framework for modeling a wide spectrum of problems. This abstract model and the methodological developments at the base of this thesis stem from the analysis and reinterpretation of mixed-integer optimal control problems with switching costs. These provided a challenging application that stimulated a shift of paradigm, highlighting a novel perspective for their formulation. Conforming to constrained structured programs and exploiting their features, we devised a suitable numerical method for computing approximate solutions, under mild assumptions. The broad perspective of augmented Lagrangian and proximal methods was then specialized to convex quadratic programming, which is a fundamental topic and forms the basis for many optimization methods and applications. The shifted penalty was found to be nothing but a proximal regularization, leading a robust method. Exploring the area where numerical linear algebra and optimization merge, we were able to design a generic, yet tailored and fast, method for solving quadratic programs. The extension of this approach to general nonlinear programming is ongoing research.

Mühldorf am Inn, June 2021 *Alberto De Marchi*

Grazie

This document owes its existence to many people, some of whom I have never met in person. I deem myself fortunate and privileged to have come across inspiring people and have been given disparate opportunities. During the last years I had the chance to get to know people from all around the world and from very different backgrounds. Wasn't it for all these experiences, I wouldn't be writing these lines. These wonderful memories will stay with me for life.

I would like to thank my Doktorvater, *Matthias Gerdts*, for welcoming me in his group. He motivated and guided me, always with his door open, and giving me more freedom than I could have imagined. His patience, enthusiasm, and scientific rigor were invaluable to me, as well as the encouragement and advices about teaching and doing research. Together with a friendly and relaxed working environment, he gave me the chance to attend several conferences and workshops around the world. Vielen Dank, Matthias.

I would like to thank *Enrico Bertolazzi*, who not only taught me numerical analysis and computational methods back in Trento, but also introduced me to numerical optimal control and encouraged me to pursue my doctorate in Munich. He was co-supervisor of my Master's thesis and co-author of my first scientific publication, together with Francesco Biral, Paolo Bosetti, and Martin M. Hanczyc, all kindly acknowledged.

I would like to thank *Sara Bressan*, who saw my future before I did. After being my high-school teacher, she became a dear friend. I will never forget the countless dinners we had with Andrea and Davide, debating about politics and the skier problem. Her passion and courage have been and remain inspirational.

I wish to express my sincere gratitude to *Christian Kirches* for accepting to review my work and for taking the time to carefully read this document.

The work presented in this thesis was also made possible by the research environment that UniBw has provided me during my studies there. I am grateful to all the incredible people in the IngMathe group for making our working environment so comfortable.

In the years 2017-2019 I attended schools in Oslo, San Diego, Estoril, Würzburg, Bremen, and Heidelberg, and conferences in Lyon, Vienna, Hanoi, Faro, Chengdu, Berlin, Dresden, and Nice. During this long journey I came across great people and shared awesome experiences I will not forget. I am happy we were there.

I am grateful to the control and optimization research communities, which have been a constant motivation throughout my doctoral studies. I have interacted with many people through their works, and some have been particularly important for my research. Also, I have benefited from the tools offered by developers who deserve more credit than they get, in particular LaTeX, Linux, Julia, and the open-source community at large.

Finally, I would like to thank my extended family and my friends for being there. Gabriele & Elena, Christian & Chen, Luca & Camilla, Eleonora, and the Caseificio tonon family. Maria Grazia, Orlando, Anna & Ben, Christian, Gian, Chica with Sara and Raquel, Paola & Giorgio with Marco, Teresa & Elia, Bruna & Tarcisio, Marta ♡

Contents

Chapter 0

Introduction

I think, therefore I sum.
—D. R. Hofstadter [36]

Mathematics pervades our life and is crucial to our understanding of the Universe [2]. Optimization is a fundamental branch of mathematics, and its long history proves it. In the antiquity, in his *Elements* Euclid considered the minimal distance between a point and a line, and Virgil's *Aeneid* gives an account of Dido's problem of enclosing the maximum area within a boundary of fixed length. Still in the prehistory of calculus, J. Kepler studied the wine barrel problem [1], and in 1638 G. Galilei, while reviewing his own work on naturally accelerated motion, suggested the need for a *higher science* than his against possible fallacies [3]. Shortly after, I. Newton and G. W. von Leibniz set the foundations of mathematical analysis [4, 5]. In 1696 the Bernoulli brothers proposed the brachistochrone's problem [6], and the calculus of variations was born. Nowadays, optimization problems are ubiquitous in science and engineering, but affect also economics and biology, among others. In fact, mathematical optimization comes into play whenever some decision variables, possibly subject to restrictions, affect some cost function, or performance index, that is to be minimized.

The discipline of nonlinear programming (NLP) deals with the search of minima of a smooth objective function over a continuous set of real variables, possibly subject to the satisfaction of constraints, usually in the form of equalities and inequalities involving smooth functions. This research field was influenced by unpublished lecture notes of W. Fenchel [9] and took off when H. W. Kuhn and A. W. Tucker [10] reinvented optimality conditions for nonlinear problems, earlier presented by W. Karush [7] in similar form. The exploitation of these conditions lead to numerical methods able to cope with large-scale nonconvex problems with many nonlinear constraints. Augmented Lagrangian (AL), sequential quadratic programming (SQP), and interior point (IP) methods are the most prominent in the field. Following different approaches, these cope with nonlinearities and constraints by introducing a sequence of nontrivial, yet related and simpler, subproblems. The availability of first- and possibly second-order derivatives and the advancement of numerical linear algebra techniques have made possible the design of efficient, reliable, and scalable algorithms. For further details, we refer, *e.g.*, to [100, 31, 34] and [150] for SQP methods, [42, 55] and [72, 90, 108, 151] for IP methods, to [21, 22, 24] and [49, 62, 163] for AL methods, and the textbooks [45, 112, 180]. Further influential works are [20, 26, 38], among others.

In a different vein, structured optimization is concerned with the minimization of a proper, lower semi-continuous, extended-real valued function, which is the sum of a smooth function and a possibly nonsmooth one. These problems are unconstrained in the sense that, if restrictions on the decision variables are in place, these are enforced via characteristic functions and the like, namely by penalizing infeasible values with an infinite cost. Proximal methods (PM), also known as operator splitting techniques, generate a series of simple subproblems, which are often elementary. Whenever this is the case, these methods easily handle nonsmooth terms, require simple algebraic operations, scale well with the problem size, and naturally lead to matrix-free implementations. For these reasons, they are particularly suitable for applications with limited hardware resources and for high-dimensional

problems, such as embedded control [187], signal processing [140, 229], and statistical learning [138]. However, Newton-type methods with superlinear convergence rate and robustness to ill-conditioning as well as the analysis for fully nonconvex problems have been only recently developed [214]. Indeed, the efficient treatment of general constraints is still a challenge [202, 209]. For an overview and further references, see the survey [170] on proximal algorithms and the textbooks [30, 143, 191].

The augmented Lagrangian (AL) approach is of particular interest in this thesis for several reasons. SQP-type methods build a linear-quadratic model of the problem around the current estimate of the solution. Thus, it is not fully clear how to handle nonsmooth problems. Interior point (IP) methods handle inequality constraints via a barrier term, thus maintaining the iterates strictly feasible. In this context, it is hard, if possible, to manage feasible sets with complicated geometry. Augmented Lagrangian methods adopt a penalty approach to (approximately) satisfy constraints and shifts to avoid unbounded penalization, if possible. Unfortunately, due to the constraint relaxation, the arising subproblems can be unbounded from below [131] and generate infeasible iterates. Moreover, some forcing sequences have to be defined, that drive and control the convergence. On the other hand, the AL framework naturally provides some desirable features. The sequence of subproblems is generated from a high-level perspective, thus maintaining the overall problem structure, these subproblems are usually unconstrained or simply constrained, and a regularization of the constraints is introduced by the penalty term [114, 146]. The interested reader may refer to [49, 62, 114, 163, 242, 228] for more details.

In the classical work [29], the profound connection between the augmented Lagrangian and the proximal point methods was uncovered in the convex setting. Inspired by the key ideas behind these approaches, yet aware of their drawbacks, this thesis attempts to carry on their symbiosis, seeking efficient and robust numerical methods, and to contribute to this *fascinating blend of heuristics and rigour, of theory and experiment* [45] that is the field of mathematical optimization.

0.1 Contributions and Outline

The aim of this thesis is to investigate the theory and develop numerical methods for (nonlinear, nonconvex, nonsmooth) constrained structured programs (NCSP), that is, finite-dimensional optimization problems with structured objective function and smooth constraints.

The foundations of this thesis are laid out in Chapter 1, where we state the problem formulation and characterize its solutions. Then, relying on both the augmented Lagrangian framework and proximal methods, we design an algorithm for its numerical solution and investigate its convergence properties. The methodology is then verified with numerical tests on a variety of problems. Although some ideas have been recently published by the author, this chapter revisits and extends those results, providing a unified framework and a more detailed analysis. We implemented the algorithms introduced in this chapter in `Bazinga`, an open-source software package for `Julia` [192]. This toolbox contains generic implementations of several algorithms, and allows to apply them on a variety of problems, as well as to modify and extend the code. We also provide `OptiMo`, a modelling tool for NCSPs in `Julia`, available online. We report numerical results on some illustrative examples and compare the proposed method against a state-of-the-art solver on some nonlinear programming benchmark problems.

Based on:

[235] A. De Marchi. "Constrained and Sparse Switching Times Optimization via Augmented Lagrangian Proximal Methods". In: *2020 American Control Conference (ACC)*. Denver, CO, USA: IEEE, 2020, pp. 3633–3638. DOI: 10.23919/ACC45564.2020.9147892;

- `Bazinga`, 2020. URL: https://github.com/aldma/Bazinga.jl;
- `OptiMo`, 2020. URL: https://github.com/aldma/OptiMo.jl.

Chapter 2 deals with switching time optimization (STO) problems with switching costs and constraints. These problems are relevant for applications and interesting in that they combine optimal control and discrete optimization in many ways. However, their hybrid nature makes them difficult

to harness by both continuous and discrete optimization techniques. We propose to interpret the switching costs as a sparsity-inducing regularization of the switching times. Following a *first discretize, then optimize* approach, this yields a constrained structured optimization problem. Despite the cardinality function being discontinuous, we show the numerical method developed in Chapter 1 is still applicable. The proposed methodology is applied to constrained STO problems with nonlinear dynamics and switching costs. We provide `ScSTO`, a modelling tool in `Julia` for STO problems with switching costs, available online. Solvers available in `Bazinga` can be invoked through the interface provided by `OptiMo`.

Based on:

[221] A. De Marchi. "On the Mixed-Integer Linear-Quadratic Optimal Control with Switching Cost". In: *IEEE Control Systems Letters* 3.4 (Oct. 2019), pp. 990–995. DOI: 10.1109/LCSYS.2019.2920425;

[235] A. De Marchi. "Constrained and Sparse Switching Times Optimization via Augmented Lagrangian Proximal Methods". In: *2020 American Control Conference (ACC)*. Denver, CO, USA: IEEE, 2020, pp. 3633–3638. DOI: 10.23919/ACC45564.2020.9147892;

[237] A. De Marchi and M. Gerdts. "Sparse Switching Times Optimization and a Sweeping Hessian Proximal Method". In: *Operations Research Proceedings 2019*. Ed. by J. S. Neufeld, U. Buscher, R. Lasch, D. Möst, and J. Schönberger. Cham: Springer, 2020, pp. 89–95. DOI: 10.1007/978-3-030-48439-2_11;

- `ScSTO`, 2020. URL: https://github.com/aldma/ScSTO.jl.

Chapter 3 deals with convex quadratic programming (QP), a fundamental topic in optimization. Building upon augmented Lagrangian and proximal methods, we develop a simple, yet efficient and robust, numerical method for convex QPs. The tailored design exploits and preserves their structure, while taking advantage of the regularization induced by the proximal augmented Lagrangian approach. The theoretical convergence properties of the proposed scheme are investigated, as well as the relationships with other numerical methods for convex quadratic programming. We implemented our algorithm in open-source `C` code and benchmarked it against state-of-the-art QP solvers, with promising results.

Based on:

[236] A. De Marchi. *On a Primal-Dual Newton Proximal Method for Convex Quadratic Programs*. Submitted. Dec. 2020. DOI: 10.13140/RG.2.2.33215.12964;

[256] `QPDO`, 2021. URL: https://github.com/aldma/qpdo.

Chapter 4 contains some final remarks and conclusions, and outlines directions of future research.

0.2 Notation and Preliminaries

Throughout this thesis, the notation aims to be simple and intuitive, yet precise; it follows the standard notation of optimization and analysis books [191, 180, 143, 163]. For the sake of clarity, we now properly specify the adopted conventions, and briefly recap known definitions and facts. The interested reader is referred for more details to the aforementioned monographs.

Numbers and sets

The set of natural numbers is denoted by $\mathbb{N}$, and we adopt the convention that $0 \in \mathbb{N}$. The set of integer, real, and extended-real numbers are denoted by $\mathbb{Z}$, $\mathbb{R}$, and $\overline{\mathbb{R}} := \mathbb{R} \cup \{\infty\}$, respectively. The symbol $:=$ denotes a definition. Unless differently specified, we adopt the convention that $1/0 = \infty$. The set of positive real numbers is indicated as $\mathbb{R}_+ := [0, \infty)$, and that of strictly positive real numbers as $\mathbb{R}_{++} := (0, \infty)$. In $\mathbb{R}^n$, the relations $<$, $\leq$, $=$, $\geq$, and $>$ are understood component-wise.

Given $a, b \in \overline{\mathbb{R}}$, we indicate with (a, b) and $[a, b]$, respectively, the open and closed (possibly extended-real) intervals having a and b as endpoints. Intervals $(a, b]$ and $[a, b)$ are defined accordingly. Occasionally, (a, b) may also indicate a pair or a vector in $\mathbb{R}^2$, however the context will always be explicit enough to avoid confusion. $[a; b]$, $(a; b)$, $[a; b)$, and $(a; b]$ stand for discrete intervals, *e.g.*, $[a; b] = [a, b] \cap \mathbb{Z}$.

The closure and interior of $E \subseteq \mathbb{R}^n$ are denoted as $\operatorname{cl} E$ and $\operatorname{int} E$, respectively. The boundary of E is $\partial E := \operatorname{cl} E \setminus \operatorname{int} E$. With $B_r(\mathbf{x})$ we indicate the closed ball centered at $\mathbf{x}$ with radius r.

Vectors and matrices

The $n \times n$ identity matrix is denoted as $\boldsymbol{I}_n$, and the $\mathbb{R}^n$ vector with all elements equal to 1 as $\mathbf{1}_n$; whenever n is clear from context we simply write $\boldsymbol{I}$ and $\mathbf{1}$, respectively. We use the Kronecker symbol $\delta_{i,j}$ for the (i, j)-th entry of $\boldsymbol{I}$. Given a vector $\mathbf{v} \in \mathbb{R}^n$, $\mathbf{v}^\top$ denotes its transpose and $\mathbf{v}^i$ its i-th component. With $\operatorname{diag} \mathbf{v}$ we indicate the $n \times n$ diagonal matrix whose i-th diagonal entry is $\mathbf{v}^i$.

$\operatorname{Sym}(\mathbb{R}^n)$, $\operatorname{Sym}_+(\mathbb{R}^n)$, and $\operatorname{Sym}_{++}(\mathbb{R}^n)$ denote respectively the set of symmetric, symmetric positive semidefinite, and symmetric positive definite matrices in $\mathbb{R}^{n \times n}$. For $\boldsymbol{Q}, \boldsymbol{R} \in \operatorname{Sym}(\mathbb{R}^n)$ we write $\boldsymbol{Q} \succeq \boldsymbol{R}$ to indicate that $\boldsymbol{Q} - \boldsymbol{R} \in \operatorname{Sym}_+(\mathbb{R}^n)$, and similarly $Q \succ R$ indicates that $\boldsymbol{Q} - \boldsymbol{R} \in \operatorname{Sym}_{++}(\mathbb{R}^n)$. Any matrix $\boldsymbol{Q} \in \operatorname{Sym}_+(\mathbb{R}^n)$ induces the semi-norm $\|\cdot\|_{\boldsymbol{Q}}$ on $\mathbb{R}^n$, where $\|\mathbf{x}\|_{\boldsymbol{Q}}^2 := \langle \mathbf{x}, \boldsymbol{Q}\mathbf{x} \rangle$; in case $\boldsymbol{Q} = \boldsymbol{I}$, that is, for the Euclidean norm, we omit the subscript and simply write $\|\cdot\|$. No ambiguity occurs in adopting the same notation for the induced matrix norm, namely $\|\boldsymbol{M}\| := \max\{\|\boldsymbol{M}\mathbf{x}\| : \mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\| = 1\}$ for $\boldsymbol{M} \in \mathbb{R}^{n \times n}$. For $p \in [1, \infty]$, the ℓ^p norm on $\mathbb{R}^n$ is denoted by $\|\cdot\|_p$, where

$$\|\mathbf{x}\|_\infty := \max\{|x_i| : i = 1, \dots, n\}, \qquad \text{and} \qquad \|\mathbf{x}\|_p := \left(\sum_{i=1}^n |x_i|^p \right)^{1/p},$$

for $p \in [1, \infty)$. The definition extends to $p \in (0, 1)$ as well, although in this case $\|\cdot\|_p$ is not subadditive and thus is only a quasi-norm. The ℓ^0 quasi-norm, namely $\|\mathbf{x}\|_0 := \operatorname{nnz}(\mathbf{x})$ the number of nonzero entries of $\mathbf{x}$, additionally fails to be homogeneous.

Sequences

The notation $\{a_k\}_{k \in K}$ represents a sequence indexed by elements of the set K, and given a set A we write $\{a_k\}_{k \in K} \subset A$ to indicate that $a_k \in A$ for all indices $k \in K$. We may omit the index set, and write just $\{a_k\} \subset A$, when $K = \mathbb{N}$ or the K is clear from the context without ambiguity.

Definition 0.2.1. A sequence $\{\mathbf{x}_k\} \subset \mathbb{R}^n$ of iterates is said to be q-convergent with limit $\mathbf{x}^\star \in \mathbb{R}^n$ if there exists $p \geq 1$ and $\mu \in [0, 1) \subset \mathbb{R}$ such that

$$\lim_{k \to \infty} \frac{\|\mathbf{x}_{k+1} - \mathbf{x}^\star\|}{\|\mathbf{x}_k - \mathbf{x}^\star\|^p} = \mu.$$

If $p = 1$, the sequence is said to converge q-linearly. If in addition $\mu = 0$, the sequence is said to converge q-superlinearly. If $p = 2$, the sequence is said to converge q-quadratically.

We will adopt the big-O and small-o notation: given sequences $\{x_k\} \subset \mathbb{R}$ and $\{\epsilon_k\} \subset \mathbb{R}_{++}$, we write $x_k \in O(\epsilon_k)$ and $x_k \in o(\epsilon_k)$ to indicate that

$$\limsup_{k \to \infty} \frac{|x_k|}{\epsilon_k} < \infty \qquad \text{and} \qquad \lim_{k \to \infty} \frac{|x_k|}{\epsilon_k} = 0,$$

respectively.

Extended-real-valued functions

Given a function $h : \mathbb{R}^n \to \overline{\mathbb{R}}$, its *domain* is the set $\operatorname{dom} h := \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}) < \infty\}$, while its *epigraph* is $\operatorname{epi} h := \{(\mathbf{x}, y) \in \mathbb{R}^n \times \mathbb{R} : h(\mathbf{x}) \leq y\}$. Function h is said to be *proper* if $\operatorname{dom} h \neq \emptyset$. We say h is *lower semicontinuous* if $\operatorname{epi} h$ is a closed set in $\mathbb{R}^{n+1}$.

Set-valued mappings

We use the notation $H : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ to indicate a point-to-set function, that is, a mapping from $\mathbb{R}^n$ to the set of all subsets of $\mathbb{R}^m$. The *domain* of H is the set $\operatorname{dom} H := \{\mathbf{x} \in \mathbb{R}^n : H(\mathbf{x}) \neq \emptyset\}$, while its *graph* is $\operatorname{gph} H := \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^m : \mathbf{y} \in H(\mathbf{x})\}$.

For notational simplicity, in case $H(\mathbf{x})$ is a singleton we may treat it as a point rather than a set, allowing notational abuses such as $H(\mathbf{x}) = \mathbf{y}$ as opposed to $H(\mathbf{x}) = \{\mathbf{y}\}$.

The *projection* onto a nonempty and closed set $S \subseteq \mathbb{R}^n$ will be meant in the set-valued sense; namely, $\Pi_S : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is defined by $\Pi_S(\mathbf{x}) := \arg\min_{\mathbf{z} \in S} \|\mathbf{z} - \mathbf{x}\|$. With $\operatorname{dist}_S(\mathbf{x}) := \inf_{\mathbf{z} \in S} \|\mathbf{z} - \mathbf{x}\|$ we indicate the *distance* of $\mathbf{x}$ from S.

Subdifferential

Consider a proper and lower semicontinuous function $h : \mathbb{R}^n \to \overline{\mathbb{R}}$ and a point $\overline{\mathbf{x}}$ with $h(\overline{\mathbf{x}})$ finite. A vector $\mathbf{v} \in \mathbb{R}^n$ is a regular subgradient of h at $\overline{\mathbf{x}}$, denoted by $\mathbf{v} \in \hat{\partial} h(\overline{\mathbf{x}})$ [143, Def. 8.3], if

$$\lim_{\substack{\mathbf{x} \to \overline{\mathbf{x}} \\ \mathbf{x} \neq \overline{\mathbf{x}}}} \frac{h(\mathbf{x}) - h(\overline{\mathbf{x}}) - \mathbf{v}^\top (\mathbf{x} - \overline{\mathbf{x}})}{\|\mathbf{x} - \overline{\mathbf{x}}\|} \geq 0.$$

We denote by $\hat{\partial} h : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ the *regular subdifferential* of h. The following result is given in [143, Thm 10.1].

Lemma 0.2.2. *Let $h : \mathbb{R}^n \to \overline{\mathbb{R}}$ be proper and lower semicontinuous. If $\overline{\mathbf{x}}$ is a local minimizer for h, then $\mathbf{0} \in \hat{\partial} h(\overline{\mathbf{x}})$.*

The (limiting) *subdifferential* of h is $\partial h : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$, and $\mathbf{v} \in \partial h(\mathbf{x})$ if and only if there exists a sequence $\{(\mathbf{x}_k, \mathbf{v}_k)\} \subseteq \operatorname{gph} \hat{\partial} h$ such that $\lim_{k \to \infty} (\mathbf{x}_k, h(\mathbf{x}_k), \mathbf{v}_k) = (\mathbf{x}, h(\mathbf{x}), \mathbf{v})$.

Let $X \subseteq \mathbb{R}^n$ be a convex set and $f : X \to \mathbb{R}$. The *conjugate* function of f is $f^* : X^* \to \mathbb{R}$ defined as

$$f^*(\mathbf{z}) := \sup_{\mathbf{x} \in X} \mathbf{z}^\top \mathbf{x} - f(\mathbf{x}),$$

where

$$X^* := \left\{ \mathbf{z} \in \mathbb{R}^n : \sup_{\mathbf{x} \in X} \mathbf{z}^\top \mathbf{x} - f(\mathbf{x}) < \infty \right\}.$$

Proximal map and Moreau envelope

The *proximal mapping* [18] of a function $g : \mathbb{R}^n \to \overline{\mathbb{R}}$ with parameter $\gamma > 0$ is the set-valued map $\operatorname{prox}_{\gamma g} : \mathbb{R}^n \rightrightarrows \operatorname{dom} g$ defined by

$$\operatorname{prox}_{\gamma g}(\mathbf{x}) := \operatorname*{arg\,min}_{\mathbf{z} \in \mathbb{R}^n} \left\{ g(\mathbf{z}) + \frac{1}{2\gamma} \|\mathbf{z} - \mathbf{x}\|^2 \right\}.$$

This can be interpreted as an approximate gradient step for g [170]; when g is differentiable and γ is sufficiently small, it is $\operatorname{prox}_{\gamma g}(\mathbf{x}) \approx \mathbf{x} - \gamma \nabla g(\mathbf{x})$. We say that a function g is prox-bounded if $g + \frac{1}{2\gamma} \|\cdot\|^2$ is bounded from below for some $\gamma > 0$. The supremum of all such γ — which is possibly infinite, as it is the case when g is lower bounded or convex — is the threshold of prox-boundedness of g, denoted as γ_g. The value function of the minimization problem defining the proximal mapping is the *Moreau envelope* with parameter γ, denoted $g^\gamma : \mathbb{R}^n \to \mathbb{R}$, namely

$$g^\gamma(\mathbf{x}) := \inf_{\mathbf{z} \in \mathbb{R}^n} \left\{ g(\mathbf{z}) + \frac{1}{2\gamma} \|\mathbf{z} - \mathbf{x}\|^2 \right\}.$$

The proximal mapping can be regarded as a generalized projection, in the sense that if χ_S is the *characteristic function* of a nonempty set $S \subseteq \mathbb{R}^n$, *i.e.*,

$$\chi_S(\mathbf{x}) := \begin{cases} 0 & \text{if } \mathbf{x} \in S, \\ +\infty & \text{otherwise,} \end{cases} \tag{0.2.1}$$

then $\operatorname{prox}_{\gamma\chi_S} = \Pi_S$ is the projection onto S for any $\gamma > 0$. Properties of the Moreau envelope and the proximal mapping are well documented in the literature [143, 140, 191]. For a proper, lower semicontinuous function g, it holds $g^\gamma \leq g$.

If g is also convex, then $\operatorname{prox}_{\gamma g}$ is single-valued and continuous, and g^γ is convex and continuously differentiable, with gradient

$$\nabla g^\gamma(\mathbf{x}) = \frac{\mathbf{x} - \operatorname{prox}_{\gamma g}(\mathbf{x})}{\gamma}, \tag{0.2.2}$$

which is γ^{-1}-Lipschitz continuous [191, Prop. 12.29].

Structured optimization [229, 200, 214] is concerned with problems in the form

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad \varphi(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x}). \tag{0.2.3}$$

Here, function $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable with L_f-Lipschitz continuous gradient, function $g : \mathbb{R}^n \to \overline{\mathbb{R}}$ is proper, lower semicontinuous and prox-bounded with threshold γ_g, and $\arg\min \varphi \neq \emptyset$; cf. [215, Ass. I]. Points of interest can be classified based on their strength as solution candidate.

Definition 0.2.3. A point $\mathbf{x}^\star \in \operatorname{dom}\varphi$ is called

- *optimal* if $\mathbf{x}^\star \in \arg\min \varphi$, *i.e.*, if it solves (0.2.3),
- *critical* if $\mathbf{x}^\star \in \operatorname{prox}_{\gamma g}\left(\mathbf{x}^\star - \gamma\nabla f(\mathbf{x}^\star)\right)$ for some $\gamma \in (0, \gamma_g)$,
- *stationary* if $\mathbf{0} \in \hat{\partial}\varphi(\mathbf{x}^\star)$.

It is shown in [215, Prop. 3.5] that

$$\text{optimality} \quad \Rightarrow \quad \text{criticality} \quad \Rightarrow \quad \text{stationarity}.$$

Thus, criticality is a halfway property between stationarity and optimality. In light of these relations, critical points satisfy a stronger necessary condition than mere stationary points. We refer to [215, §3] for a detailed discussion.

The forward-backward splitting (FBS) algorithm, also known as proximal gradient method, is a well-known algorithm for addressing structured optimization problems. Based on the recurrence

$$\mathbf{x}_{k+1} \in \operatorname{prox}_{\gamma g}\left(\mathbf{x}_k - \gamma\nabla f(\mathbf{x}_k)\right), \qquad \gamma \in (0, \gamma_g),$$

it only requires ∇f and $\operatorname{prox}_{\gamma g}$ as oracles and converges to a critical point under very mild assumptions [155]. The key is the following sufficient decrease property, whose proof can be found in [164, Lem. 2].

Lemma 0.2.4. *Suppose g is prox-bounded with threshold γ_g. Let $\mathbf{x} \in \mathbb{R}^n$ be arbitrary. Then, for all $\gamma \in (0, \gamma_g)$ and for any $\overline{\mathbf{x}} \in \operatorname{prox}_{\gamma g}(\mathbf{x} - \gamma\nabla f(\mathbf{x}))$, it holds*

$$\varphi(\overline{\mathbf{x}}) \leq \varphi(\mathbf{x}) - \frac{1 - \gamma L_f}{2\gamma}\|\mathbf{x} - \overline{\mathbf{x}}\|^2.$$

Apparently, selecting any $\gamma \in (0, \min\{\gamma_g, 1/L_f\})$ guarantees that every forward-backward (or proximal-gradient) step yields sufficient decrease of the objective function φ.

The forward-backward envelope (FBE), first proposed in [161], serves as a real-valued, continuously differentiable, exact penalty function for the original problem (0.2.3). Hence, it allows to bridge the gap between structured optimization and smooth unconstrained optimization; cf. [200, 214]. The FBE of φ with parameter $\gamma > 0$ is given by

$$\varphi_\gamma(\mathbf{x}) := \inf_{\mathbf{z}\in\mathbb{R}^n}\left\{f(\mathbf{x}) + \nabla f(\mathbf{x})^\top(\mathbf{z} - \mathbf{x}) + \frac{1}{2\gamma}\|\mathbf{z} - \mathbf{x}\|^2 + g(\mathbf{z})\right\}. \tag{0.2.4}$$

Remarkably, the FBE can be computed solely based on the same oracles required by the FBS, namely ∇f and $\operatorname{prox}_{\gamma g}$, and it is a surrogate of the Moreau envelope [18] for structured problems of the form (0.2.3).

First-order necessary conditions

Let us consider a general nonlinear program (NLP) in the form

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{c}^i(\mathbf{x}) = \mathbf{0} \qquad i \in \mathcal{E} \\ & \mathbf{c}^i(\mathbf{x}) \leq \mathbf{0} \qquad i \in \mathcal{I}, \end{aligned} \tag{0.2.5}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $\mathbf{c} : \mathbb{R}^n \to \mathbb{R}^m$ are all smooth, real-valued functions, and $\mathcal{E}$ and $\mathcal{I}$ are two finite sets of indices, such that $\mathcal{E} \cup \mathcal{I} = [1; m]$. As before, we call f the objective function, while $\mathbf{c}^i$, $i \in \mathcal{E}$, are the equality constraints and $\mathbf{c}^i$, $i \in \mathcal{I}$, are the inequality constraints. The Lagrangian function for the constrained problem (0.2.5) is defined as

$$\mathcal{L}(\mathbf{x}, \mathbf{y}) := f(\mathbf{x}) + \mathbf{y}^\top \mathbf{c}(\mathbf{x}). \tag{0.2.6}$$

Regularity conditions known as *constraint qualifications* are adopted to ensure degenerate behavior does not occur at points of interest. One such constraint qualification, often used albeit strong, is the following one [112, Def. 12.1].

Definition 0.2.5. Given a point $\mathbf{x}$, the *linear independence constraint qualification* (LICQ) holds if the set of active constraint gradients $\{\nabla \mathbf{c}^i(\mathbf{x}) : i \in \mathcal{E} \vee \mathbf{c}^i(\mathbf{x}) = 0\}$ is linearly independent.

Note that if this condition holds, none of the active constraint gradients can be zero. In practice, weaker constraint qualifications are preferred since they provide stronger optimality conditions [180, Chap. 4].

Constraint qualifications allow us to establish necessary optimality conditions for the general NLP in (0.2.5) [112, Thm. 12.1]. We refer to these as first-order conditions because they involve the gradients of the objective and constraint functions. The conditions (0.2.7) below are known as the Karush–Kuhn–Tucker (KKT) conditions, tracing back to [7, 10].

Theorem 0.2.6. *Suppose that $\mathbf{x}_\star \in \mathbb{R}^n$ is a local solution of (0.2.5) and that the LICQ holds at $\mathbf{x}_\star$. Then there exists a unique Lagrange multiplier vector $\mathbf{y}_\star \in \mathbb{R}^m$ such that*

$$\nabla_x \mathcal{L}(\mathbf{x}_\star, \mathbf{y}_\star) = \mathbf{0} \tag{0.2.7a}$$

$$\mathbf{c}^i(\mathbf{x}_\star) = 0 \qquad i \in \mathcal{E} \tag{0.2.7b}$$

$$\max\left(\mathbf{c}^i(\mathbf{x}_\star), -\mathbf{y}^i_\star\right) = 0 \qquad i \in \mathcal{I}. \tag{0.2.7c}$$

Points that satisfy the KKT conditions will be called *KKT points*. From this perspective, the KKT conditions are *pointwise*, while the approximate, or asymptotic, KKT (AKKT) conditions are *sequential* optimality conditions. These appear in the following definition, from [163, Def. 3.1].

Definition 0.2.7. We say that $\mathbf{x}_\star \in \mathbb{R}^n$ satisfies the AKKT conditions for (0.2.5) if $\mathbf{x}_\star$ is feasible and there exist sequences $\{\mathbf{x}_k\} \subseteq \mathbb{R}^n$ and $\{\mathbf{y}_k\} \subseteq \mathbb{R}^m$ such that

$$\lim_{k\to\infty} \mathbf{x}_k = \mathbf{x}_\star \tag{0.2.8a}$$

$$\lim_{k\to\infty} \nabla_x \mathcal{L}(\mathbf{x}_k, \mathbf{y}_k) = \mathbf{0} \tag{0.2.8b}$$

$$\lim_{k\to\infty} \mathbf{c}^i(\mathbf{x}_k) = 0 \qquad i \in \mathcal{E} \tag{0.2.8c}$$

$$\lim_{k\to\infty} \max\left(\mathbf{c}^i(\mathbf{x}_k), -\mathbf{y}^i_k\right) = 0 \qquad i \in \mathcal{I}. \tag{0.2.8d}$$

Note that, unlike KKT, the AKKT conditions hold at every local minimizer of an optimization problem independently of the fulfillment of constraint qualifications. Constraint qualifications are properties of the constraints of optimization problems that, when satisfied at a local minimizer $\mathbf{x}$, independently of the objective function, imply that $\mathbf{x}$ fulfills the KKT condition [163, §3.1.2]. Sequential optimality conditions constitute an active field of research [129, 136, 216], also in relation with the design and analysis of optimization algorithms [228].

Semismoothness

We present here some general results concerning the concept of semismoothness, firstly introduced in [32]; see also [57, 56, 65, 76] and [47, 54].

A function $h : \mathbb{R}^n \to \overline{\mathbb{R}}$ is directionally differentiable at $\mathbf{x} \in \operatorname{dom} h$ if for every $\mathbf{d} \in \mathbb{R}^n$ the limit

$$h'(\mathbf{x}; \mathbf{d}) := \lim_{\tau \to 0^+} \frac{h(\mathbf{x} + \tau \mathbf{d}) - h(\mathbf{x})}{\tau} \tag{0.2.9}$$

exists. The quantity $h'(\mathbf{x}; \mathbf{d})$ is the directional derivative of h at $\mathbf{x}$ along direction $\mathbf{d}$. For a vector-valued function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$, $\nabla \mathbf{f}(\mathbf{x}) := \mathbf{f}'(\mathbf{x})^\top$ denotes the Jacobian matrix of $\mathbf{f}$ at $\mathbf{x} \in \mathbb{R}^n$. The following definitions are taken from [69].

Definition 0.2.8. A locally Lipschitzian function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ is *semismooth* at $\mathbf{x} \in \mathbb{R}^n$ if

$$\lim_{\substack{V \in \partial \mathbf{f}(\mathbf{x} + t\mathbf{d}') \\ \mathbf{d}' \to \mathbf{d},\, t \downarrow 0}} \{V \mathbf{d}'\}$$

exists for any $\mathbf{d} \in \mathbb{R}^n$.

Herein, $\partial \mathbf{f}$ denotes the generalized subdifferential of $\mathbf{f}$ [40]

$$\partial \mathbf{f}(\mathbf{x}) := \operatorname{convh} \left\{ \lim_{\substack{\mathbf{x}^k \in D_f \\ \mathbf{x}^k \to \mathbf{x}}} \nabla \mathbf{f}(\mathbf{x}^k) \right\}$$

where D_f is the set where $\mathbf{f}$ is differentiable and convh denotes the convex hull. If $\mathbf{f}$ is semismooth at $\mathbf{x}$, then $\mathbf{f}$ is directionally differentiable at $\mathbf{x}$ and $\mathbf{f}'(\mathbf{x}; \mathbf{d})$, that is, the directional derivative of $\mathbf{f}$ at $\mathbf{x}$ in the direction $\mathbf{d}$, is equal to the limit in Definition 0.2.8.

Definition 0.2.9. Suppose $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ is semismooth at $\mathbf{x} \in \mathbb{R}^n$. Then, $\mathbf{f}$ is *strongly semismooth* at $\mathbf{x}$ if for any $V \in \partial \mathbf{f}(\mathbf{x} + \mathbf{d})$, $\mathbf{d} \to \mathbf{0}$,

$$V \mathbf{d} - \mathbf{f}'(\mathbf{x}; \mathbf{d}) = O\left(\|\mathbf{d}\|^2\right).$$

Note that strong semismoothness is also referred to as 1-order semismoothness [57].

Chapter 1

Constrained Structured Optimization

This chapter introduces the problem class considered in this thesis, develops a numerical method based on the augmented Lagrangian and proximal frameworks, investigates its convergence properties, and reports on numerical results.

Some ideas contained in this chapter appear in [235].

1.1 Introduction

Augmented Lagrangian and proximal methods have recently attracted revived and grown interest. While the latter easily handle nonsmooth and extended-real valued terms, the former are able to efficiently tackle large-scale constrained problems. This chapter introduces the class of constrained structured optimization problems, and develops a numerical method for their solution. This builds upon both the augmented Lagrangian framework, which traces back to the classical work of Hestenes [21], Powell [22, 35], and Rockafellar [29], and proximal methods inaugurated by Moreau [18]. More recent accounts on these topics can be found in [49, 62, 163] and [170, 214]. Part of this chapter is based on the generalized and proximal augmented Lagrangian approaches, described in details in [114, 149] and [222].

We are interested in nonlinear, nonconvex, nonsmooth Constrained Structured Programs (NCSPs), namely optimization problems of the form

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad \varphi(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x}) \qquad \text{subject to} \quad \mathbf{c}(\mathbf{x}) \in S, \tag{1.1.1}$$

where $\mathbf{x}$ is the decision variable, f and g form the objective function φ, $\mathbf{c}$ represents the constraints, and S is the constraint set. As such, NCSPs are finite-dimensional optimization problems with structured objective function subject to constraints.

The following blanket assumptions are considered throughout the rest of this chapter:

- $f : \mathbb{R}^n \to \mathbb{R}$ and $\mathbf{c}_i : \mathbb{R}^n \to \mathbb{R}$, $i \in [1; m]$, are continuously differentiable functions with Lipschitz continuous gradient;
- $g : \mathbb{R}^n \to \overline{\mathbb{R}}$ is a proper, lower semi-continuous, extended-real valued function; g is continuous on its domain $\Omega := \operatorname{dom} g$, which is a convex compact set in $\mathbb{R}^n$;
- $S \subseteq \mathbb{R}^m$ is a nonempty, closed, possibly nonconvex set;
- the feasible set $D := \{\mathbf{x} \in \mathbb{R}^n : \mathbf{c}(\mathbf{x}) \in S\} \cap \Omega$ is nonempty and closed;
- the set of feasible minimizers is nonempty, namely $\arg\min_{\mathbf{x}\in D} \varphi(\mathbf{x}) \neq \emptyset$.

Moreover, we work under the practical assumption that the proximal mapping of g and the projection onto S can be efficiently evaluated at any point, namely that

- given any $\mathbf{x} \in \mathbb{R}^n$, any $\gamma > 0$, and any $\mathbf{v} \in \mathbb{R}^m$, it is required negligible computational effort to find arbitrary $\overline{\mathbf{x}} \in \operatorname{prox}_{\gamma g}(\mathbf{x})$ and $\overline{\mathbf{v}} \in \Pi_S(\mathbf{v})$.

The method proposed in this chapter heavily relies on these two oracles, which are, indeed, the only way both the objective g and the constraint set S are accessed.

Note. Signal processing, statistics, and numerical optimization have always influenced each other [229]. Classical examples are regularized inverse problems, such as penalized least-squares problems, among others [63, 116, 174]. In common applications, function g represents a regularization or sparsity-inducing term, and can be, *e.g.*, the ℓ^0 or ℓ^1 norms. For these and many other functions, the proximal operator can be expressed analytically or evaluated very efficiently. The same applies to the projection operator of the constraint set S, which often is convex, a polytope, or consists of the union of closed, convex sets [206]. For an exhaustive list of functions arising in many applications, see [138, 170] and the collections in [186, 258, 252].

Structured optimization deals with the minimization of $\varphi := f + g$, without any (explicit) restrictions. Proximal algorithms (also known as operator splitting techniques) are often the methods of choice due to their simplicity and versatility [214]. For an overview on this topic, we refer to [139, 140, 138, 200, 229, 234]. In this context, constraints are usually enforced via the nonsmooth term g, including the characteristic function of the feasible set. An analogous procedure can be adopted to deal with (1.1.1). As function g is allowed to be extended-real valued and nonsmooth, one could move the constraint $\mathbf{c}(\mathbf{x}) \in S$ into the objective function by replacing g with $\tilde{g} : \mathbb{R}^n \to \overline{\mathbb{R}}$ defined as $\tilde{g}(\mathbf{x}) := g(\mathbf{x}) + \chi_S(\mathbf{c}(\mathbf{x}))$ for any $\mathbf{x} \in \mathbb{R}^n$. In this thesis, however, we face NCSPs from a different perspective, in order to shed light on the relationships between methods arising in different optimization contexts, such as proximal techniques for structured optimization and augmented Lagrangian methods for nonlinear programming. The latter approach is considered here because it is based on a sequence of unconstrained or simply constrained subproblems, it can handle nonconvex constraints and is often superior to pure penalty methods, and it enjoys good warm-starting capabilities; see [49, 180, 242]. It allows to avoid ill-conditioning due to a pure penalty approach and to deal with constraints without softening them, in contrast with [202, 209, 257]. In this framework, proximal methods play a key role, since NCSPs yield subproblems in the form of structured optimization problems.

The contribution is outlined as follows. Necessary conditions are derived in §1.2, which characterize solutions and form the basis for designing a numerical method. The shifted penalty approach is introduced and discussed in §1.3. Further development in §1.4 yields the augmented Lagrangian proximal method. The designed algorithm is presented in §1.5, along with its convergence analysis. Finally, implementation details and numerical evaluations are reported in §1.6.

1.2 Optimality Conditions

Constrained optimization aims at finding the lowest possible value of an objective function within a given domain, the feasible set of the decision variable. If the feasible set D is nonempty, we say the problem is feasible, and a point $\mathbf{x} \in \mathbb{R}^n$ is called feasible if $\mathbf{x} \in D$. A point $\mathbf{x}^\star \in D$ is referred to as a (strict) global minimizer if $\varphi(\mathbf{x}^\star) \leq \varphi(\mathbf{x})$ ($<$) for all $\mathbf{x} \in D$. Instead, it is referred to as a (strict) local minimizer if there exists $\epsilon > 0$ such that $\varphi(\mathbf{x}^\star) \leq \varphi(\mathbf{x})$ ($<$) for all $\mathbf{x} \in D \cap B_\epsilon(\mathbf{x}^\star)$. The value of φ at a (local or global) minimizer will be called (local or global) minimum. If D is compact (closed and bounded in $\mathbb{R}^n$), the Bolzano–Weierstrass theorem guarantees that a (global) minimizer of φ over D exists.

Global optimization techniques are available that tackle the task of finding a global minimizer, which is usually very hard. In this thesis, we focus on *affordable, iterative algorithms* [163, §3], which only guarantee convergence to points that satisfy some necessary optimality condition. These are conditions that necessarily hold at every local (or global) minimizer. In general, points that satisfy necessary optimality conditions are only candidate, *i.e.*, probable, minimizers. In nonlinear programming, points that satisfy necessary optimality conditions are usually said to be stationary or critical. In structured optimization, instead, a hierarchy of optimal, critical, and stationary points exist [215, Prop. 3.5]:

$$\text{optimality} \quad \Rightarrow \quad \text{criticality} \quad \Rightarrow \quad \text{stationarity}.$$

This means optimal points are necessarily critical, and critical ones are necessarily stationary. Moreover, for iterative methods, it is useful to rely on sequential optimality conditions. These consider sequences of points, instead of single points, and thus inspire termination criteria for iterative methods; see, *e.g.*, [136, 228] and [163, 173]. The following result gives necessary, sequential optimality conditions for the NCSP in (1.1.1).

Theorem 1.2.1. *Let $\mathbf{x}^\star$ be a feasible local minimizer for problem (1.1.1). Assume that f and $\mathbf{c}$ admit first derivatives in a neighborhood of $\mathbf{x}^\star$. Then, there exist sequences $\{\mathbf{x}_k\} \subset \mathbb{R}^n$, $\{\mathbf{y}_k\} \subset \mathbb{R}^m$, and $\{\eta_k\} \subset \mathbb{R}_{++}$ such that*

$$\lim_{k\to+\infty} \mathbf{x}_k = \mathbf{x}^\star \tag{1.2.1}$$

$$\lim_{k\to+\infty} \operatorname{dist}_S(\mathbf{c}(\mathbf{x}_k)) = 0 \tag{1.2.2}$$

$$\lim_{k\to+\infty} \|\mathbf{c}(\mathbf{x}_k) - \Pi_S(\mathbf{c}(\mathbf{x}_k) + \eta_k \mathbf{y}_k)\| = 0 \tag{1.2.3}$$

$$\lim_{k\to+\infty} \left\| \mathbf{x}_k - \operatorname{prox}_{\gamma_k g}\left(\mathbf{x}_k - \gamma_k \left[\nabla f(\mathbf{x}_k) + \nabla \mathbf{c}(\mathbf{x}_k)^\top \mathbf{y}_k\right]\right)\right\| = 0 \tag{1.2.4}$$

for all $\{\gamma_k\} \subset \mathbb{R}_{++}$ sufficiently small.

Proof. The proof is divided into four main steps, one for each condition, and follows standard arguments [163, 180].

(i) First, based on an auxiliary problem, we construct a sequence $\{\mathbf{x}_k\}$ which admits a limit point $\overline{\mathbf{x}}$ and then we show that $\overline{\mathbf{x}}$ coincides with $\mathbf{x}^\star$. By hypothesis, there exists $\epsilon > 0$ such that $\mathbf{x}^\star$ is a global minimizer of φ on $D \cap B_\epsilon(\mathbf{x}^\star)$. Therefore, $\mathbf{x}^\star$ is the unique global minimizer of $\varphi(\mathbf{x}) + \|\mathbf{x} - \mathbf{x}^\star\|^2$ on $D \cap B_\epsilon(\mathbf{x}^\star)$. Consider, for all $k \in \mathbb{N}$, the problem

$$\underset{\mathbf{x}\in B_\epsilon(\mathbf{x}^\star)}{\text{minimize}} \quad \varphi(\mathbf{x}) + \frac{k}{2}\operatorname{dist}_S^2(\mathbf{c}(\mathbf{x})) + \|\mathbf{x} - \mathbf{x}^\star\|^2, \tag{1.2.5}$$

which admits a solution $\mathbf{x}_k \in B_\epsilon(\mathbf{x}^\star)$, by the Bolzano–Weierstraß theorem. Since $B_\epsilon(\mathbf{x}^\star)$ is compact and, by definition, $\mathbf{x}_k \in B_\epsilon(\mathbf{x}^\star)$ for all $k \in \mathbb{N}$, there exists a limit point $\overline{\mathbf{x}} \in B_\epsilon(\mathbf{x}^\star)$ and a subsequence $K \subset \mathbb{N}$ such that $\lim_{k\in K} \mathbf{x}_k = \overline{\mathbf{x}}$. By definition of $\mathbf{x}_k$, for all $k \in \mathbb{N}$, it holds

$$\begin{aligned}\varphi(\mathbf{x}_k) + \frac{k}{2}\operatorname{dist}_S^2(\mathbf{c}(\mathbf{x}_k)) + \|\mathbf{x}_k - \mathbf{x}^\star\|^2 &\le \varphi(\mathbf{x}^\star) + \frac{k}{2}\operatorname{dist}_S^2(\mathbf{c}(\mathbf{x}^\star)) + \|\mathbf{x}^\star - \mathbf{x}^\star\|^2 \\ &= \varphi(\mathbf{x}^\star),\end{aligned} \tag{1.2.6}$$

since $\mathbf{x}^\star$ is a feasible point. By continuity of φ and $\mathbf{c}$, taking the limit for $k \in K$, (1.2.6) yields $\operatorname{dist}_S^2(\mathbf{c}(\overline{\mathbf{x}})) = 0$. Thus, $\overline{\mathbf{x}}$ is a feasible point too. Furthermore, since the term $k \operatorname{dist}_S^2(\mathbf{c}(\mathbf{x}_k))$ is non-negative, we have from (1.2.6) that $\varphi(\mathbf{x}_k) + \|\mathbf{x}_k - \mathbf{x}^\star\|^2 \le \varphi(\mathbf{x}^\star)$. Taking the limit for $k \in K$ gives $\varphi(\overline{\mathbf{x}}) + \|\overline{\mathbf{x}} - \mathbf{x}^\star\|^2 \le \varphi(\mathbf{x}^\star)$. However, since $\overline{\mathbf{x}} \in B_\epsilon(\mathbf{x}^\star)$ and $\mathbf{x}^\star$ is the global minimizer of $\varphi(\mathbf{x}) + \|\mathbf{x} - \mathbf{x}^\star\|^2$ on $D \cap B_\epsilon(\mathbf{x}^\star)$, it must be that $\overline{\mathbf{x}} = \mathbf{x}^\star$. This proves (1.2.1).

(ii) By (i), feasibility of $\mathbf{x}^\star$, continuity of $\mathbf{c}$ and dist_S, (1.2.2) readily follows.

(iii) Due to (i), for sufficiently large $k \in K$, it is $\mathbf{x}_k \in \operatorname{int} B_\epsilon(\mathbf{x}^\star)$. Hence, $\mathbf{x}_k$ is a unconstrained optimal point for (1.2.5), therefore it is also critical [215]. Let $\psi_k : \mathbb{R}^n \to \mathbb{R}$ collect the terms in the objective function of (1.2.5) apart from g, namely

$$\psi_k(\mathbf{x}) := f(\mathbf{x}) + \frac{k}{2}\operatorname{dist}_S^2(\mathbf{c}(\mathbf{x})) + \|\mathbf{x} - \mathbf{x}^\star\|^2.$$

Without explicit constraints on $\mathbf{x}$ for (1.2.5), since $\mathbf{x}_k \in \operatorname{int} B_\epsilon(\mathbf{x}^\star)$, it necessarily holds

$$\mathbf{x}_k \in \operatorname{prox}_{\gamma_k g}(\mathbf{x}_k - \gamma_k \nabla \psi_k(\mathbf{x}_k)) \tag{1.2.7}$$

for some $\gamma_k > 0$ sufficiently small. Direct calculation yields

$$\nabla \psi_k(\mathbf{x}_k) = \nabla f(\mathbf{x}_k) + k \nabla \mathbf{c}(\mathbf{x}_k)^\top [\mathbf{c}(\mathbf{x}_k) - \mathbf{p}(\mathbf{x}_k)] + 2(\mathbf{x}_k - \mathbf{x}^\star), \tag{1.2.8}$$

with $\mathbf{p}(\mathbf{x}_k) \in \Pi_S(\mathbf{c}(\mathbf{x}_k))$, showing that $\nabla\psi_k$ is possibly set-valued. By defining

$$\mathbf{y}_k := k\,[\mathbf{c}(\mathbf{x}_k) - \mathbf{p}(\mathbf{x}_k)] \tag{1.2.9}$$

and taking the limit for $k \in K$, (1.2.7)–(1.2.9) yield (1.2.4).

(iv) It remains to prove that the sequences $\{\mathbf{x}_k\}$, $\{\mathbf{y}_k\}$ constructed in (i) and (iii) satisfy (1.2.3) for some sequence $\{\eta_k\} \subset \mathbb{R}_{++}$. Let us consider two cases: (a) there exists $k_0 \in \mathbb{N}$ such that $\mathbf{c}(\mathbf{x}_k) \in S$ for all $k \geq k_0$, $k \in K$, or (b) $\mathbf{c}(\mathbf{x}_k) \in \mathbb{R}^m \setminus S$ for all $k \in K$.

(iv-a) It is $\Pi_S(\mathbf{c}(\mathbf{x}_k)) = \mathbf{c}(\mathbf{x}_k)$, and thus $\mathbf{y}_k = \mathbf{0}$ by (1.2.9), for all $k \geq k_0$, $k \in K$. Then, by continuity of dist_S, (1.2.3) follows.

(iv-b) For such sequence, it is $\mathbf{p}_k := \mathbf{p}(\mathbf{x}_k) \neq \mathbf{c}(\mathbf{x}_k)$ and $\mathbf{y}_k \neq \mathbf{0}$ for all $k \in K$. Let us denote $\mathbf{w}_k := \mathbf{c}(\mathbf{x}_k) + \eta_k \mathbf{y}_k$, for some $\eta_k > 0$, and $\mathbf{z}_k \in \Pi_S(\mathbf{w}_k)$ for any $k \in K$. Hence, (1.2.3) can be expressed as $\lim_{k\to+\infty} \|\mathbf{c}(\mathbf{x}_k) - \mathbf{z}_k\| = 0$. Moreover, we have the following upper bound

$$\begin{aligned}
\|\mathbf{c}(\mathbf{x}_k) - \mathbf{z}_k\| &\leq \|\mathbf{c}(\mathbf{x}_k) - \mathbf{w}_k\| + \|\mathbf{w}_k - \mathbf{z}_k\| \\
&\leq \|\mathbf{c}(\mathbf{x}_k) - \mathbf{w}_k\| + \|\mathbf{w}_k - \mathbf{p}_k\| \\
&= \|\mathbf{w}_k - \mathbf{c}(\mathbf{x}_k)\| + \|\mathbf{w}_k - \mathbf{c}(\mathbf{x}_k) + \mathbf{c}(\mathbf{x}_k) - \mathbf{p}_k\| \\
&= \|\eta_k \mathbf{y}_k\| + \|\eta_k \mathbf{y}_k + \mathbf{c}(\mathbf{x}_k) - \mathbf{p}_k\| \\
&= k\eta_k \|\mathbf{c}(\mathbf{x}_k) - \mathbf{p}_k\| + (k\eta_k + 1)\|\mathbf{c}(\mathbf{x}_k) - \mathbf{p}_k\| \\
&= (2k\eta_k + 1)\operatorname{dist}_S(\mathbf{c}(\mathbf{x}_k)),
\end{aligned} \tag{1.2.10}$$

where the first line is due to the triangle inequality, and the second holds because, by definition, $\mathbf{z}_k$ has minimum distance to $\mathbf{w}_k$. The following lines are obtained by considering (1.2.9), $\eta_k > 0$, $k \geq 0$, and the definitions of $\mathbf{w}_k$ and dist_S. Therefore, by using (ii), it is sufficient to select $\{\eta_k\} \subset \mathbb{R}_{++}$ such that $\{k\eta_k\}$ is bounded in order to guarantee that $\lim_{k\to+\infty} \|\mathbf{c}(\mathbf{x}_k) - \mathbf{z}_k\| = 0$. This implies (1.2.3), and concludes the proof. □

Note that parameters γ_k and η_k are positive scalars in Theorem 1.2.1. Nonetheless, with minor modifications, they can be replaced by positive definite matrices, as in [208, 224] and [49]. In practice, algorithms may benefit from a finer tuning of parameters, for controlled ill-conditioning and improved regularization.

Theorem 1.2.1 closely matches the approximate KKT, or AKKT, conditions in NLP, which are the sequential counterpart of the classical KKT conditions [136], widely discussed in the literature [112, 163, 180]. Condition (1.2.4) corresponds to the stationarity of the Lagrangian function, extended to structured optimization. Similarly, (1.2.3) replaces the transversality conditions. Indeed, by considering constraints of the form $\mathbf{c}(\mathbf{x}) \leq \mathbf{0}$, and hence the set $S := \mathbb{R}^m_-$, (1.2.3) simplifies and the classical condition is recovered, namely $\lim_{k\to+\infty} \|\max\{\mathbf{c}(\mathbf{x}_k), -\mathbf{y}_k\}\| = 0$. We highlight the latter does not depend on the sequence $\{\eta_k\}$, by convexity of S. Indeed, due to the hidden, possibly nontrivial structure of S, defining a complementarity condition for NCSP is not straightforward. The following result shows yet another necessary condition, on the vein of [129, 173], which aims at enforcing feasibility and complementarity separately.

Proposition 1.2.2. *Let $\mathbf{x}^\star$ be a feasible local minimizer for problem (1.1.1). Assume that f and $\mathbf{c}$ admit first derivatives in a neighborhood of $\mathbf{x}^\star$. Let the sequences $\{\mathbf{x}_k\} \subset \mathbb{R}^n$, $\{\mathbf{y}_k\} \subset \mathbb{R}^m$, $\{\gamma_k\} \subset \mathbb{R}_{++}$, and $\{\eta_k\} \subset \mathbb{R}_{++}$ be constructed as in Theorem 1.2.1. Then, it holds*

$$\lim_{k\to+\infty} \langle \mathbf{y}_k, \mathbf{c}(\mathbf{x}_k) - \Pi_S(\mathbf{c}(\mathbf{x}_k)) \rangle = 0. \tag{1.2.11}$$

Proof. By continuity of φ, (1.2.1), and nonnegativity of the distance, taking the limit for $k \to +\infty$ in (1.2.6) yields $\lim_{k\to+\infty} k \operatorname{dist}_S^2(\mathbf{c}(\mathbf{x}_k)) = 0$. Thus, it is

$$0 = \lim_{k\to+\infty} k \operatorname{dist}_S^2(\mathbf{c}(\mathbf{x}_k)) = \lim_{k\to+\infty} k\|\mathbf{c}(\mathbf{x}_k) - \mathbf{p}(\mathbf{x}_k)\|^2 = \lim_{k\to+\infty} \langle \mathbf{y}_k, \mathbf{c}(\mathbf{x}_k) - \mathbf{p}(\mathbf{x}_k) \rangle$$

with $\mathbf{p}(\mathbf{x}_k)$ and $\mathbf{y}_k$ as defined in Theorem 1.2.1. The result readily follows. □

Condition (1.2.11) resembles a transversality condition, but it is not. In fact, since $\mathbf{c}(\mathbf{x}) = \Pi_S(\mathbf{c}(\mathbf{x}))$ for every $\mathbf{c}(\mathbf{x}) \in S$, it does not guarantee $\lim_{k\to\infty} \mathbf{y}_k = \mathbf{0}$ in the case $\lim_{k\to\infty} \mathbf{c}(\mathbf{x}_k) \in \operatorname{int} S$. Instead, condition (1.2.3) generalizes the classical transversality condition, in that it combines feasibility and complementarity. Unfortunately, it introduces the need for an additional positive parameter η, which scales the dual variable $\mathbf{y}$ and makes the transversality conditions dependent on its direction only. It is apparent that (1.2.3) implies feasibility of the limit point. The following result shows that, for a strictly feasible minimizer, *i.e.* $\mathbf{c}(\mathbf{x}^\star) \in \operatorname{int} S$, condition (1.2.3) guarantees the dual variable vanishes, *i.e.* $\lim_{k\to+\infty} \mathbf{y}_k = \mathbf{0}$.

Proposition 1.2.3. *Suppose S has nonempty interior. Let $\mathbf{x}^\star$ be a feasible (local) minimizer for problem (1.1.1). Let the sequences $\{\mathbf{x}_k\} \subset \mathbb{R}^n$, $\{\mathbf{y}_k\} \subset \mathbb{R}^m$, and $\{\eta_k\} \subset \mathbb{R}_{++}$ satisfy (1.2.3). Let $K \subset \mathbb{N}$ be a subsequence such that $\lim_{k\in K} \mathbf{x}_k = \mathbf{x}^\star$. Then, if $\mathbf{c}(\mathbf{x}^\star) \in \operatorname{int} S$, it is $\mathbf{y}_k = \mathbf{0}$ for all $k \in K$ sufficiently large.*

Proof. By continuity of $\mathbf{c}$, there exists $k_0 \in \mathbb{N}$ such that $\mathbf{c}(\mathbf{x}_k) \in \operatorname{int} S$ for all $k \geq k_0$, $k \in K$. Therefore, since $\eta_k > 0$ for all k, it follows from (1.2.3) that $\mathbf{y}_k = \mathbf{0}$ for all $k \geq k_0$, $k \in K$, yielding the result. □

It remains to show that, for $\mathbf{c}(\mathbf{x}^\star) \in \partial S$, the dual variable points outwards and is normal to S at $\mathbf{c}(\mathbf{x}^\star)$, in the sense of the following definition, which extends concepts usually adopted for convex sets.

Definition 1.2.4. Let a closed set $S \subset \mathbb{R}^m$ and a point $\mathbf{s} \in \partial S$ be given. A vector $\mathbf{v} \in \mathbb{R}^m$ is said to *point outwards* S at $\mathbf{s}$ if and only if $\mathbf{v} \neq \mathbf{0}$ and there exists a scalar $\overline{\alpha} > 0$ such that $\mathbf{s} + \alpha\mathbf{v} \notin S$ for all $\alpha \in (0, \overline{\alpha}]$. A vector $\mathbf{v}$ is said to be *normal* to S at $\mathbf{s}$ if and only if $\mathbf{v} \neq \mathbf{0}$ and there exists a scalar $\overline{\alpha} > 0$ such that $\mathbf{s} \in \Pi_S(\mathbf{s} + \alpha\mathbf{v})$ for all $\alpha \in [0, \overline{\alpha}]$.

Notice that, for a nonconvex set S, the scaling factor $\alpha > 0$ is needed in a projection-based definition, in order to account only for the direction of $\mathbf{v}$, and not for its magnitude, analogously to η_k in (1.2.3). Moreover, a vector $\mathbf{v}$ normal to S at $\mathbf{s}$ necessarily points outwards S at $\mathbf{s}$. Since Theorem 1.2.1 gives sequential optimality conditions, we seek a sequential transversality condition, *i.e.*, the dual variable is expected to be pointing outwards and normal to S only asymptotically.

Proposition 1.2.5. *Let $\mathbf{x}^\star$ be a feasible local minimizer for problem (1.1.1). Let the sequences $\{\mathbf{x}_k\} \subset \mathbb{R}^n$, $\{\mathbf{y}_k\} \subset \mathbb{R}^m$, and $\{\eta_k\} \subset \mathbb{R}_{++}$ satisfy (1.2.3). Let $K \subset \mathbb{N}$ be a subsequence such that $\lim_{k\in K} \mathbf{x}_k = \mathbf{x}^\star$. Moreover, suppose that for all $k_0 \in K$ there exists $k \geq k_0$ such that $\mathbf{y}_k \neq \mathbf{0}$. Then, it is $\mathbf{c}(\mathbf{x}^\star) \in \partial S$ and either $\lim_{k\to+\infty} \mathbf{y}_k = \mathbf{0}$ or the elements of the sequence $\{\mathbf{y}_k\}_{k\in K}$ asymptotically point outwards and are normal to S at $\mathbf{c}(\mathbf{x}^\star)$, in the sense of Definition 1.2.4.*

Proof. From the feasibility of $\mathbf{x}^\star$ and (the negation of) Proposition 1.2.3, we deduce that $\mathbf{c}(\mathbf{x}^\star) \in \partial S$. By (1.2.3), it is $\lim_{k\to\infty} \|\mathbf{c}(\mathbf{x}_k) - \Pi_S(\mathbf{c}(\mathbf{x}_k) + \eta_k\mathbf{y}_k)\| = 0$. Since $\eta_k > 0$ for all k, this implies that either the dual sequence vanishes or it is asymptotically normal to S, and thus pointing outwards, at $\lim_{k\to+\infty} \mathbf{c}(\mathbf{x}) = \mathbf{c}(\mathbf{x}^\star)$. □

Let us briefly elaborate on the optimality condition (1.2.7), given above for the unconstrained subproblem. Here we show that (1.2.7) is a valid necessary optimality condition, despite the gradient of ψ being possibly a set-valued mapping, in case S is nonconvex. Firstly, we equivalently reformulate the subproblem (1.2.5) as

$$\underset{\mathbf{x}\in B_\epsilon(\mathbf{x}^\star),\ \mathbf{z}\in\mathbb{R}^m}{\text{minimize}} \quad \varphi(\mathbf{x}) + \chi_S(\mathbf{z}) + \frac{k}{2}\|\mathbf{z} - \mathbf{c}(\mathbf{x})\|^2 + \|\mathbf{x} - \mathbf{x}^\star\|^2, \tag{1.2.12}$$

which is a structured optimization problem, whose objective function is proper, lower semi-continuous, and extended-real valued, and consists of two terms. Let us denote

$$\psi^z(\mathbf{x}, \mathbf{z}) := f(\mathbf{x}) + \frac{k}{2}\|\mathbf{z} - \mathbf{c}(\mathbf{x})\|^2 + \|\mathbf{x} - \mathbf{x}^\star\|^2$$

the continuously differentiable term, whose gradient reads

$$\nabla\psi^z(\mathbf{x},\mathbf{z}) = \begin{pmatrix} \nabla f(\mathbf{x}) + k\nabla\mathbf{c}(\mathbf{x})^\top[\mathbf{c}(\mathbf{x}) - \mathbf{z}] + 2(\mathbf{x}-\mathbf{x}^\star) \\ k[\mathbf{z} - \mathbf{c}(\mathbf{x})] \end{pmatrix}.$$

The remaining, nonsmooth term is given by $g^z(\mathbf{x},\mathbf{z}) := g(\mathbf{x}) + \chi_S(\mathbf{z})$ and, thanks to the separable structure, for any given $\gamma > 0$, its proximal mapping is

$$\operatorname{prox}_{\gamma g^z}(\mathbf{x},\mathbf{z}) = \begin{pmatrix} \operatorname{prox}_{\gamma g}(\mathbf{x}) \\ \Pi_S(\mathbf{z}) \end{pmatrix}.$$

An unconstrained solution $(\mathbf{x}_k^z, \mathbf{z}_k^z)$ of (1.2.12) necessarily satisfies

$$\mathbf{x}_k^z \in \operatorname{prox}_{\gamma_k^z g}\left(\mathbf{x}_k^z - \gamma_k^z \nabla_x \psi^z(\mathbf{x}_k^z, \mathbf{z}_k^z)\right)$$

for some $\gamma_k^z > 0$ sufficiently small [170, 214]. Moreover, by looking at (1.2.12), it is apparent the optimal value $\mathbf{z}_k^z$ for $\mathbf{z}$ must be the point in S closest to $\mathbf{c}(\mathbf{x}_k^z)$, that is, $\mathbf{z}_k^z \in \Pi_S(\mathbf{c}(\mathbf{x}_k^z))$. Thus, it is $\mathbf{z}_k^z = \mathbf{c}(\mathbf{x}_k^z)$ if $\mathbf{c}(\mathbf{x}_k^z) \in S$ and, if $\mathbf{c}(\mathbf{x}_k^z) \notin S$, $\mathbf{z}_k^z \in \partial S$ and the vector $\mathbf{v}_k^z := \mathbf{c}(\mathbf{x}_k^z) - \mathbf{z}_k^z \neq \mathbf{0}$ points outwards and is normal to S at $\mathbf{z}_k^z$, according to Definition 1.2.4, since

$$\mathbf{z}_k^z \in \Pi_S\left(\mathbf{c}(\mathbf{x}_k^z)\right) = \Pi_S\left(\mathbf{z}_k^z + \mathbf{v}_k^z\right)$$

and therefore $\mathbf{z}_k^z \in \Pi_S(\mathbf{z}_k^z + \alpha\mathbf{v}_k^z)$ for all $\alpha \in [0,1]$. Comparing with Theorem 1.2.1, it is sufficient to (arbitrarily but accordingly) select $\mathbf{p}(\mathbf{x}_k^z) = \mathbf{z}_k^z \in \Pi_S(\mathbf{c}(\mathbf{x}_k^z))$ in order to recover the same expressions. Not only this shows the optimality condition (1.2.4) is valid, but also provides a glimpse on how to exploit the structure of NCSP for an implicit treatment of slack variables. This idea is made more precise in §1.4, by introducing the augmented Lagrangian proximal framework.

1.3 Shifted Penalty Method

Consider the NCSP in (1.1.1). The penalty method is a simple yet effective approach for constrained optimization [20, 180]. Thanks to the continuity assumptions on φ and $\mathbf{c}$, it can be applied to solve (1.1.1). This relies on the concept of penalty function. A function $p : \mathbb{R}^m \to \mathbb{R}$ is called a *penalty function* for the nonempty set $S \subseteq \mathbb{R}^m$ if the following hold:

$$p(\mathbf{z}) = 0 \quad \Leftrightarrow \quad \mathbf{z} \in S \tag{1.3.1a}$$

$$p(\mathbf{z}) > 0 \quad \Leftrightarrow \quad \mathbf{z} \notin S \tag{1.3.1b}$$

As the name suggests, this function is adopted for discouraging constraint violations. In fact, a constrained problem is tackled by solving a sequence of unconstrained problems of the form

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad q_k(\mathbf{x}) := \varphi(\mathbf{x}) + \eta_k p(\mathbf{c}(\mathbf{x})) \tag{1.3.2}$$

for some strictly increasing sequence $\{\eta_k\} \subset \mathbb{R}_+$, with $\lim_{k\to+\infty} \eta_k = +\infty$. Note that q_k plays the role of a (parametric) merit function, which balances objective and constraint violations. Denoting $\mathbf{x}_k$ an unconstrained minimizer of q_k, it is well-known that any limit point $\overline{\mathbf{x}}$ of the sequence $\{\mathbf{x}_k\}$ is indeed a solution to the original constrained problem (1.1.1) [180]. The following result collects some basic properties exhibited by the penalty method.

Lemma 1.3.1 (Penalty lemma). *The following hold:*

$$(i) \quad q_k(\mathbf{x}_k) \leq q_{k+1}(\mathbf{x}_{k+1}) \qquad (iii) \quad \varphi(\mathbf{x}_k) \leq \varphi(\mathbf{x}_{k+1})$$

$$(ii) \quad p(\mathbf{c}(\mathbf{x}_k)) \geq p(\mathbf{c}(\mathbf{x}_{k+1})) \qquad (iv) \quad \varphi(\mathbf{x}_k) \leq q_k(\mathbf{x}_k) \leq \varphi(\mathbf{x}^\star)$$

Proof. For the sake of brevity, we use subscripts and omit arguments whenever the meaning is clear, *e.g.*, $\varphi_k = \varphi(\mathbf{x}_k)$ and $p_k = p(\mathbf{c}(\mathbf{x}_k))$. Using definition of q_k and $\mathbf{x}_k$, we have

$$q_k(\mathbf{x}_k) = \varphi_k + \eta_k p_k \leq \varphi_{k+1} + \eta_k p_{k+1} \leq \varphi_{k+1} + \eta_{k+1} p_{k+1} = q_{k+1}(\mathbf{x}_{k+1}),$$

which proves (i). Furthermore, by noticing that $\varphi_k + \eta_k p_k \leq \varphi_{k+1} + \eta_k p_{k+1}$ and $\varphi_{k+1} + \eta_{k+1} p_{k+1} \leq \varphi_k + \eta_{k+1} p_k$, we obtain

$$\eta_k [p_k - p_{k+1}] \leq \varphi_{k+1} - \varphi_k \leq \eta_{k+1} [p_k - p_{k+1}],$$

which implies (ii) since $\eta_{k+1} > \eta_k$. Using $\varphi_k + \eta_k p_k \leq \varphi_{k+1} + \eta_k p_{k+1}$ and $\eta_k \geq 0$, (ii) yields (iii). The definition of penalty function yields $q_k(\mathbf{x}_k) = \varphi_k + \eta_k p_k \geq \varphi_k$ and, with feasibility of $\mathbf{x}^\star$, for any η_k

$$\varphi(\mathbf{x}^\star) = \varphi(\mathbf{x}^\star) + \eta_k p(\mathbf{x}^\star) \geq \varphi_k + \eta_k p_k = q_k(\mathbf{x}_k),$$

which prove (iv). □

The next result concerns convergence of the penalty method under some continuity assumptions on the problem and the penalty function; cf. [180].

Theorem 1.3.2 (Penalty convergence). *Suppose that φ, $\mathbf{c}$, and p are continuous functions. Let a strictly increasing sequence $\{\eta_k\} \subset \mathbb{R}_+$ be given, with $\lim_{k\to+\infty} \eta_k = +\infty$. Let $\{\mathbf{x}_k\}$ be a sequence of solutions $\mathbf{x}_k$ to (1.3.2). Then, any limit point $\overline{\mathbf{x}}$ of $\{\mathbf{x}_k\}$ solves (1.1.1).*

Proof. Let $\overline{\mathbf{x}}$ be any limit point of $\{\mathbf{x}_k\}$, and let $K \subset \mathbb{N}$ be any subsequence such that $\lim_{k\in K} \mathbf{x}_k = \overline{\mathbf{x}}$. From continuity of φ, we have $\lim_{k\in K} \varphi_k = \varphi(\overline{\mathbf{x}})$. Moreover, due to Lemma 1.3.1(iv),

$$q^\star := \lim_{k\in K} q_k(\mathbf{x}_k) \leq \varphi(\mathbf{x}^\star)$$

is bounded from above. Then

$$\lim_{k\in K} \eta_k p_k = \lim_{k\in K} [q_k(\mathbf{x}_k) - \varphi_k] = q^\star - \varphi(\overline{\mathbf{x}})$$

remains bounded despite $\eta_k \to +\infty$, which implies $\lim_{k\in K} p_k = 0$. From the continuity of $\mathbf{c}$ and p, $p(\mathbf{c}(\overline{\mathbf{x}})) = 0$, and so, by definition of penalty function, $\overline{\mathbf{x}}$ is a feasible point, since $\mathbf{c}(\overline{\mathbf{x}}) \in S$. From Lemma 1.3.1(iv), $\varphi(\mathbf{x}^\star) \geq \varphi_k$ for all k, and so $\varphi(\overline{\mathbf{x}}) \leq \varphi(\mathbf{x}^\star)$, which proves that $\overline{\mathbf{x}}$ is a feasible minimizer for (1.1.1). □

1.3.1 Shifting the constraints

The classical penalty method, as represented by (1.3.2), introduces a penalization term based directly on constraints violation. The parametric merit function q combines both optimality and constraint violation requirements, as a weighted sum of the two terms. As parameter η grows, the constraint violation is expected to reduce, by Lemma 1.3.1(ii), and, eventually, a feasible minimizer is found, cf. Theorem 1.3.2. However, increasing η makes the unconstrained subproblems (1.3.2) more and more ill-conditioned, although benignly [94]. A classical approach for alleviating this phenomenon, if not for avoiding it, is to penalize infeasibility with respect to shifted constraints, namely considering unconstrained subproblems of the form

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad \varphi(\mathbf{x}) + \eta p\left(\mathbf{c}(\mathbf{x}) + \frac{\mathbf{y}}{\eta}\right) \tag{1.3.3}$$

for some vector $\mathbf{y} \in \mathbb{R}^m$. The quantities $\mathbf{y}/\eta$ and $\mathbf{y}$ are usually referred to as shift and Lagrange multiplier. The idea behind shifting the constraints is that, for some specific shift $\mathbf{y}$, the solution to subproblem (1.3.3) may, perhaps approximately, coincide with the desired minimizer of (1.1.1), even with a bounded penalty parameter η [94, 163].

The classical formulation (1.3.3) has a peculiar form which suggests some crucial properties of the shift and of the resulting method. Firstly, let us observe that, as the penalty parameter η tends to

infinity, the shift should vanish. In fact, if this is not the case, some feasible points would be hardly penalized, and thus considered as infeasible. Hence, we expect $\mathbf{y}_k/\eta_k \to \mathbf{0}$ as $\eta_k \to +\infty$. This is the peculiar feature of safeguarded augmented Lagrangian methods [163, 196, 211]. This designation stems from the observation that, if a suitable shift cannot be found that solves the problem with a bounded penalty parameter, subproblems (1.3.3) tend to (1.3.2) and the method falls back to the classical penalty method, and then relies on Theorem 1.3.2. Perhaps the simplest and most common strategy, yet not the only one, to guarantee the safeguarding vanishing property is to consider bounded multipliers $\mathbf{y}_k \in Y \subset \mathbb{R}^m$.

The function $p(\mathbf{z}) := \operatorname{dist}_S^2(\mathbf{z})/2$ is a continuous penalty function for the constraint $\mathbf{c}(\mathbf{x}) \in S$. In general, the gradient of p is given by $\nabla p(\mathbf{z}) = \mathbf{z} - \Pi_S(\mathbf{z})$ and therefore is a set-valued mapping. Only if the set S is convex, function p is continuously differentiable with Lipschitz continuous gradient. Nevertheless, this quadratic penalization gives a favorable expression for its gradient, despite the possibly complicated structure of S, and relies only on the projection operator Π_S. From (1.3.3), it leads to the shifted penalty subproblem

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{x}) + g(\mathbf{x}) + \frac{\eta}{2} \operatorname{dist}_S^2\left(\mathbf{c}(\mathbf{x}) + \frac{\mathbf{y}}{\eta}\right), \tag{1.3.4}$$

which is an unconstrained, structured optimization problem that can be handled by proximal algorithms such as, *e.g.*, PANOC [202]. In §1.5, an algorithm for solving (1.1.1) is framed around subproblem (1.3.4) and its convergence properties are investigated. In particular, §1.5.4 discusses convergence guarantees for proximal algorithms solving the subproblem in the case the set S is nonconvex (and g lower semicontinuous), on the vein of (1.2.5) and (1.2.12).

1.3.2 Discontinuous objective

It is of great interest the possibility to drop the continuity assumption on the nonsmooth term g. In such case, however, the penalty method is not guaranteed to converge to solutions of the original, constrained problem. Nonetheless, there are some approaches to overcome this issue, which are briefly discussed in the following.

Enveloped objective A simple idea is to find an equivalent yet continuous reformulation of the problem, so to recover the convergence guarantee given by the penalty method. The forward-backward envelope (FBE), proposed in [161], has been exploited for adopting algorithms for smooth optimization on nonsmooth problems [202, 200, 214]. Extending the Moreau envelope [18], the FBE $\varphi_\gamma^{\text{FB}}$ of φ is an exact, continuous, real-valued penalty function for the unconstrained problem $\operatorname{minimize}_{\mathbf{x}\in\mathbb{R}^n} \varphi(\mathbf{x})$ [215, Prop. 4.2], for some sufficiently small stepsize $\gamma > 0$. The following result shows that replacing φ with $\varphi_\gamma^{\text{FB}}$ yields a suitable problem reformulation. In the following, we adopt the simpler notation φ_γ for the FBE of φ, without superscript whenever clear from context.

Lemma 1.3.3. *Let φ_γ be the FBE of $\varphi := f + g$ with $\gamma \in (0, \min\{1/L_f, \gamma_g\})$. Then, the problem*

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad \varphi_\gamma(\mathbf{x}) \qquad \text{subject to} \quad \mathbf{c}(\mathbf{x}) \in S \tag{1.3.5}$$

is equivalent to (1.1.1).

Proof. Rewrite (1.1.1) as $\operatorname{minimize}_{\mathbf{x}\in D} \varphi(\mathbf{x})$, and move the constraint into the objective via the characteristic function χ_D of D: $\operatorname{minimize}_{\mathbf{x}\in\mathbb{R}^n} \varphi(\mathbf{x}) + \chi_D(\mathbf{x})$. Denoting $\tilde{\varphi} := \varphi + \chi_D$ and $\tilde{\varphi}_\gamma$ its FBE, the previous is equivalent to $\operatorname{minimize}_{\mathbf{x}\in\mathbb{R}^n} \tilde{\varphi}_\gamma(\mathbf{x})$, since the FBE shares infima and minimizers, under the Lemma's standing assumptions [215]. Due to χ_D, the minimizers of $\tilde{\varphi}_\gamma$ are feasible, hence one can rewrite $\operatorname{minimize}_{\mathbf{x}\in D} \tilde{\varphi}_\gamma(\mathbf{x})$. Indeed, for feasible points, the term χ_D does not give any contribution to $\tilde{\varphi}_\gamma$, which then collapses to φ_γ. This gives $\operatorname{minimize}_{\mathbf{x}\in D} \varphi_\gamma(\mathbf{x})$, proving the result. □

Based on Theorem 1.3.2, we can find a solution $\mathbf{x}_\gamma^\star$ to the associated constrained enveloped problem (1.3.5) by using the (classical) penalty method, that is, by considering a sequence of unconstrained

penalty enveloped problems

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad \varphi_\gamma(\mathbf{x}) + \eta p(\mathbf{c}(\mathbf{x})) \tag{1.3.6}$$

Then, due to Lemma 1.3.3, $\mathbf{x}_\gamma^\star$ is feasible and optimal for φ_γ and therefore a solution to (1.1.1). Despite the fact that the FBE φ_γ is smoother than φ, even continuously differentiable if f is twice continuously differentiable, evaluating its gradient $\nabla\varphi_\gamma$ can be computationally expensive, in that it involves the Hessian of f [215]. Nevertheless, algorithms not requiring the gradient of the FBE could be investigated; compare, *e.g.*, [201] and [215]. Finally, it should be highlighted that, for g not necessarily continuous, one could use proximal methods for solving (1.3.2), and these could implicitly construct the FBE of $\varphi + \eta p \circ \mathbf{c}$ and minimize it. Despite the strong resemblance with (1.3.6), these problems are not equivalent in general.

Characteristic function Another approach to cope with (1.1.1) is to move the constraints to the objective via characteristic functions. As mentioned in §1.1, this is perhaps the standard approach. This reformulation gives the equivalent problem

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{x}) + g(\mathbf{x}) + \chi_S(\mathbf{c}(\mathbf{x})) \tag{1.3.7}$$

and three-terms splitting algorithms can be directly applied [197, 193]. However, these usually require evaluating the proximal operator of the nonsmooth terms, which for $\chi_S \circ \mathbf{c}$ corresponds to solving the problem

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad \chi_S(\mathbf{c}(\mathbf{x})) + \frac{1}{2\gamma}\|\mathbf{x} - \hat{\mathbf{x}}\|^2$$

for some given $\hat{\mathbf{x}} \in \mathbb{R}^n$ and $\gamma > 0$. In fact, this is equivalent to projecting $\hat{\mathbf{x}}$ onto the feasible set D, and this can be as difficult as the original problem (1.1.1). Defining $\tilde{g} = g + \chi_S \circ \mathbf{c}$ and adopting two-terms splitting algorithms yields similar, if not more difficult, subproblems, and thus carry comparable drawbacks.

A slightly different reformulation of (1.1.1) introduces an auxiliary variable $\mathbf{z} \in \mathbb{R}^m$ and reads

$$\underset{\mathbf{x}\in\mathbb{R}^n,\ \mathbf{z}\in\mathbb{R}^m}{\text{minimize}} \quad f(\mathbf{x}) + g(\mathbf{x}) + \chi_S(\mathbf{z}) \qquad \text{subject to} \quad \mathbf{c}(\mathbf{x}) = \mathbf{z}. \tag{1.3.8}$$

Then, using the characteristic function of $\{\mathbf{0}\}$ and moving the equality constraints to the objective, one obtains the unconstrained, structured problem

$$\underset{\mathbf{x}\in\mathbb{R}^n,\ \mathbf{z}\in\mathbb{R}^m}{\text{minimize}} \quad f(\mathbf{x}) + g(\mathbf{x}) + \chi_S(\mathbf{z}) + \chi_{\{\mathbf{0}\}}(\mathbf{c}(\mathbf{x}) - \mathbf{z}). \tag{1.3.9}$$

This gives rise to two additional ways to adopt three-terms splittings, depending on the pairing of the nonsmooth terms.

The first comprises the nonsmooth terms $g(\mathbf{x}) + \chi_S(\mathbf{z})$ and $\chi_{\{\mathbf{0}\}}(\mathbf{c}(\mathbf{x}) - \mathbf{z})$. Evaluating the proximal operator of the former turns out to be fairly simple, as it boils down to evaluating the proximal operator of g and projecting onto S, which are both cheap operations by assumption. The proximal operator of the latter, instead, is equivalent to solving a nonlinear least-squares problem (1.3.10) or projecting onto a manifold generated by the constraints (1.3.11).

$$\begin{aligned}
&\underset{\mathbf{x}\in\mathbb{R}^n,\mathbf{z}\in\mathbb{R}^m}{\text{minimize}} \quad \chi_{\{\mathbf{0}\}}(\mathbf{c}(\mathbf{x}) - \mathbf{z}) + \frac{1}{2\gamma}\|\mathbf{x} - \hat{\mathbf{x}}\|^2 + \frac{1}{2\gamma}\|\mathbf{z} - \hat{\mathbf{z}}\|^2 \\
&\quad\Rightarrow \quad \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad \left\|\begin{pmatrix}\mathbf{x}\\ \mathbf{c}(\mathbf{x})\end{pmatrix} - \begin{pmatrix}\hat{\mathbf{x}}\\ \hat{\mathbf{z}}\end{pmatrix}\right\|^2 && (1.3.10)\\
&\quad\Rightarrow \quad \underset{\mathbf{x}\in\mathbb{R}^n,\mathbf{z}\in\mathbb{R}^m}{\text{minimize}} \quad \left\|\begin{pmatrix}\mathbf{x}\\ \mathbf{z}\end{pmatrix} - \begin{pmatrix}\hat{\mathbf{x}}\\ \hat{\mathbf{z}}\end{pmatrix}\right\|^2 && (1.3.11)\\
&\qquad\qquad \text{subject to} \quad \mathbf{c}(\mathbf{x}) = \mathbf{z}
\end{aligned}$$

Efficient and robust algorithms are available for solving such problems, possibly depending on the structure of $\mathbf{c}$, based upon Gauß–Newton, Levenberg–Marquardt [8, 17], and Powell's dogleg [23] methods, among others.

The second three-terms splitting of (1.3.9) is based on the terms $\chi_S(\mathbf{z})$ and $g(\mathbf{x}) + \chi_{\{0\}}(\mathbf{c}(\mathbf{x}) - \mathbf{z})$. Clearly, the former has an inexpensive proximal operator, which corresponds to the projection onto the set S. However, the latter may require more computational effort. This, again, offers two reformulations, which mirror (1.3.10) and (1.3.11).

$$\begin{aligned}
&\underset{\mathbf{x}\in\mathbb{R}^n,\mathbf{z}\in\mathbb{R}^m}{\text{minimize}} \quad g(\mathbf{x}) + \chi_{\{0\}}(\mathbf{c}(\mathbf{x}) - \mathbf{z}) + \frac{1}{2\gamma}\|\mathbf{x} - \hat{\mathbf{x}}\|^2 + \frac{1}{2\gamma}\|\mathbf{z} - \hat{\mathbf{z}}\|^2 \\
&\quad\Rightarrow \quad \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad g(\mathbf{x}) + \frac{1}{2\gamma}\left\|\begin{pmatrix}\mathbf{x}\\ \mathbf{c}(\mathbf{x})\end{pmatrix} - \begin{pmatrix}\hat{\mathbf{x}}\\ \hat{\mathbf{z}}\end{pmatrix}\right\|^2 \qquad (1.3.12)\\
&\quad\Rightarrow \quad \underset{\mathbf{x}\in\mathbb{R}^n,\mathbf{z}\in\mathbb{R}^m}{\text{minimize}} \quad g(\mathbf{x}) + \frac{1}{2\gamma}\left\|\begin{pmatrix}\mathbf{x}\\ \mathbf{z}\end{pmatrix} - \begin{pmatrix}\hat{\mathbf{x}}\\ \hat{\mathbf{z}}\end{pmatrix}\right\|^2 \qquad (1.3.13)\\
&\qquad\qquad \text{subject to} \quad \mathbf{c}(\mathbf{x}) = \mathbf{z}
\end{aligned}$$

The constrained formulation given in (1.3.13) has the form of the original problem (1.1.1) and thus, at a first glance, it gives a *cul-de-sac*. Nevertheless, it may be easier, since it involves a strongly convex smooth objective function and only equality constraints, and thus solvable in different ways [257, 209]. On the other hand, the unconstrained (sub)problem (1.3.12) is a structured, nonsmooth problem itself. Hence, generic proximal methods can be adopted to solve it [202, 220]. Depending on the structure of g, more specific methods may apply, such as those discussed in [122, 128, 137, 139, 168, 182].

Considering both three-terms formulations with auxiliary variable, evaluating the proximal operator of one of such terms requires solving a subproblem. Nevertheless, this nested, two-loops structure is a common feature of iterative methods for nonlinearly constrained optimization. Although not further investigated here, these reformulations offer opportunities for novel analysis, interpretations, and algorithms for problem (1.1.1) without continuity assumption on g.

1.4 Augmented Lagrangian and Proximal Approaches

Here we consider different ways to approach the possibly nonsmooth subproblems introduced in the shifted penalty method. Two interpretations are given.

Let us consider once more the reformulation given in (1.3.8)

$$\underset{\mathbf{x}\in\mathbb{R}^n,\ \mathbf{z}\in\mathbb{R}^m}{\text{minimize}} \quad f(\mathbf{x}) + g(\mathbf{x}) + \chi_S(\mathbf{z}) \qquad \text{subject to} \quad \mathbf{c}(\mathbf{x}) = \mathbf{z},$$

which introduces an auxiliary variable $\mathbf{z} \in \mathbb{R}^m$. This closely resembles the reformulation behind the alternating direction method of multipliers (ADMM), which traces back to [11, 28, 37, 52]; more recent works are [138, 170, 185]. We now proceed as standard in the AL framework for NLP [62, 112, 163, 180]. Let us define the Lagrangian function

$$\mathcal{L}^z(\mathbf{x}, \mathbf{z}, \mathbf{y}) := f(\mathbf{x}) + g(\mathbf{x}) + \chi_S(\mathbf{z}) + \mathbf{y}^\top[\mathbf{c}(\mathbf{x}) - \mathbf{z}] \qquad (1.4.1)$$

where $\mathbf{y} \in \mathbb{R}^m$ is the Lagrange multiplier associated with the equality constraint. Note that, $\mathbf{x}$ and $\mathbf{y}$ are often referred to as primal and dual variable, respectively, and a pair $(\mathbf{x}, \mathbf{y})$ as a primal-dual pair. Then, let us consider the (Powell–Hestenes–Rockafellar) augmented Lagrangian function [22, 21, 25, 27, 35]

$$\mathcal{L}^z_\mu(\mathbf{x}, \mathbf{z}, \mathbf{y}) := \mathcal{L}^z(\mathbf{x}, \mathbf{z}, \mathbf{y}) + \frac{1}{2\mu}\|\mathbf{c}(\mathbf{x}) - \mathbf{z}\|^2 \qquad (1.4.2)$$

with $\mu > 0$ a given penalty parameter. In the classical AL framework, problem (1.1.1) is replaced by a sequence of subproblems, each of which consists of minimizing the AL function $\mathcal{L}^z_\mu$ with respect to the primal variables, here $\mathbf{x}$ and $\mathbf{z}$, for some given penalty parameter $\mu > 0$ and (safeguarded)

estimate of Lagrange multiplier $\mathbf{y}$. We better expose the elements in $\mathcal{L}_\mu^z$ by completion of squares and rearranging; this yields

$$\begin{aligned}\mathcal{L}_\mu^z(\mathbf{x},\mathbf{z},\mathbf{y}) &= f(\mathbf{x}) + g(\mathbf{x}) + \chi_S(\mathbf{z}) + \mathbf{y}^\top[\mathbf{c}(\mathbf{x}) - \mathbf{z}] + \frac{1}{2\mu}\|\mathbf{c}(\mathbf{x}) - \mathbf{z}\|^2 \\ &= f(\mathbf{x}) + g(\mathbf{x}) + \chi_S(\mathbf{z}) + \frac{1}{2\mu}\|\mathbf{z} - [\mathbf{c}(\mathbf{x}) + \mu\mathbf{y}]\,\|^2 - \frac{\mu}{2}\|\mathbf{y}\|^2. \end{aligned} \tag{1.4.3}$$

Owing to the structure exhibited in (1.4.3), the minimization of $\mathcal{L}_\mu^z(\cdot,\cdot,\mathbf{y})$ can proceed inspired by several strategies. In the following, we discuss two approaches to carry out this task, namely solving the AL subproblems. First, we introduce the augmented Lagrangian proximal (ALP) approach, adopted in [235, 257]. The AL subproblem is solved as a structured optimization problem, *i.e.*, auxiliary variable $\mathbf{z}$ is left to optimization along with $\mathbf{x}$. Then, we discuss the proximal augmented Lagrangian (PAL) approach [29, 194, 222], which explicitly minimizes $\mathcal{L}_\mu^z$ with respect to the auxiliary variable $\mathbf{z}$, so that only the primal variable $\mathbf{x}$ is left as decision variable for the subproblem. Indeed, the AL framework is in common, and these two approaches are just different ways to face the AL subproblems; recall the relationship between (1.2.5) and (1.2.12). The PAL strategy tends to generate smoother, regularized, smaller problems, while the ALP leads to larger, possibly more structured and sparse, problems. In contrast with the fact that cheap projections are often disregarded [87, 163], both approaches heavily rely on the projection onto the constraint set S.

Augmented Lagrangian proximal approach The problem of minimizing the AL function $\mathcal{L}_\mu^z$ given in (1.4.3), for some multiplier estimate $\mathbf{y}$ and penalty parameter $\mu > 0$, can be expressed as

$$\underset{\mathbf{x}\in\mathbb{R}^n,\ \mathbf{z}\in\mathbb{R}^m}{\text{minimize}} \quad f^z(\mathbf{x},\mathbf{z}) + g^z(\mathbf{x},\mathbf{z}), \tag{1.4.4}$$

where the smooth and the nonsmooth terms, respectively f^z and g^z are clear from (1.4.3). Function f^z inherits regularity from f and $\mathbf{c}$, and thus is continuously differentiable with Lipschitz continuous gradient, which after rearrangement reads

$$\nabla f^z(\mathbf{x},\mathbf{z}) = \begin{pmatrix} \nabla f(\mathbf{x}) + \nabla\mathbf{c}(\mathbf{x})^\top(\mathbf{y} + [\mathbf{c}(\mathbf{x}) - \mathbf{z}]/\mu) \\ -(\mathbf{y} + [\mathbf{c}(\mathbf{x}) - \mathbf{z}]/\mu) \end{pmatrix}. \tag{1.4.5}$$

Similarly to (1.2.12), the proximal operator of g^z boils down to

$$\operatorname{prox}_{\gamma g^z}(\mathbf{x},\mathbf{z}) = \begin{pmatrix} \operatorname{prox}_{\gamma g}(\mathbf{x}) \\ \Pi_S(\mathbf{z}) \end{pmatrix} \tag{1.4.6}$$

for any $\gamma > 0$. Although evaluating these quantities can be done efficiently, subproblem (1.4.4) possibly has many more decision variables than the original problem (1.1.1), as the minimization is over both $\mathbf{x}$ and $\mathbf{z}$. Nevertheless, as the oracles are fairly inexpensive, this may be a viable approach for solving the AL subproblem (1.4.3); depending on the problem, further exploitation may also be possible.

Proximal augmented Lagrangian approach Considering the minimization of $\mathcal{L}_\mu^z(\cdot,\cdot,\mathbf{y})$, one can formally solve for $\mathbf{z}$ for each fixed $\mathbf{x}$:

$$\mathbf{z}_\mu(\mathbf{x},\mathbf{y}) := \operatorname*{arg\,min}_{\mathbf{z}\in\mathbb{R}^m} \mathcal{L}_\mu^z(\mathbf{x},\mathbf{z},\mathbf{y}) \tag{1.4.7}$$

$$\begin{aligned} &= \operatorname*{arg\,min}_{\mathbf{z}\in\mathbb{R}^m} \chi_S(\mathbf{z}) + \frac{1}{2\mu}\|\mathbf{z} - [\mathbf{c}(\mathbf{x}) + \mu\mathbf{y}]\,\|^2 \\ &= \operatorname{prox}_{\mu\chi_S}(\mathbf{c}(\mathbf{x}) + \mu\mathbf{y}) \\ &= \Pi_S(\mathbf{c}(\mathbf{x}) + \mu\mathbf{y})\,. \end{aligned} \tag{1.4.8}$$

This reflects the spirit of the PAL strategy of [222] but also resembles the classical approach to handle inequality constraints in the AL framework [62, §3.1]. Injecting this back into (1.4.3), we obtain the PAL function

$$\begin{aligned}\mathcal{L}_\mu(\mathbf{x},\mathbf{y}) &:= \mathcal{L}^z_\mu(\mathbf{x},\mathbf{z}_\mu(\mathbf{x},\mathbf{y}),\mathbf{y}) && (1.4.9)\\ &= f(\mathbf{x}) + g(\mathbf{x}) + \chi_S^\mu\left(\mathbf{c}(\mathbf{x}) + \mu\mathbf{y}\right) - \frac{\mu}{2}\|\mathbf{y}\|^2 \\ &= f(\mathbf{x}) + g(\mathbf{x}) + \frac{1}{2\mu}\operatorname{dist}_S^2\left(\mathbf{c}(\mathbf{x}) + \mu\mathbf{y}\right) - \frac{\mu}{2}\|\mathbf{y}\|^2, && (1.4.10)\end{aligned}$$

where χ_S^μ denotes the Moreau envelope of χ_S with stepsize $\mu > 0$ [143, 215]. Thus, the AL subproblem is expressed in terms of $\mathbf{x}$ only. We notice that, comparing (1.3.4) and (1.4.10), the penalty parameter $\mu > 0$ plays the same role as $\eta^{-1} > 0$ and, for any given multiplier estimate $\mathbf{y} \in \mathbb{R}^m$, the two expressions differ by a constant term, which has no effect when minimizing with respect to variable $\mathbf{x}$. Although foreseeable, this correspondence emphasizes the remarkable relationships between augmented Lagrangian function, proximal operator, and shifted penalty method [27, 29, 94, 214].

1.4.1 Gradients and shift updates

Let the smooth part of the Lagrangian and the proximal augmented Lagrangian functions be denoted respectively by

$$\pi(\mathbf{x},\mathbf{y}) := f(\mathbf{x}) + \mathbf{c}(\mathbf{x})^\top\mathbf{y} \tag{1.4.11}$$

$$\pi_\mu(\mathbf{x},\mathbf{y}) := f(\mathbf{x}) + \frac{1}{2\mu}\operatorname{dist}_S^2\left(\mathbf{c}(\mathbf{x}) + \mu\mathbf{y}\right) - \frac{\mu}{2}\|\mathbf{y}\|^2. \tag{1.4.12}$$

Then, the point-wise necessary optimality condition for a solution $(\mathbf{x}^\star,\mathbf{y}^\star)$ to the original problem (1.1.1) reads

$$\mathbf{x}^\star \in \operatorname{prox}_{\gamma g}\left(\mathbf{x}^\star - \gamma\nabla_x\pi(\mathbf{x}^\star,\mathbf{y}^\star)\right) \tag{1.4.13}$$

for some $\gamma > 0$ sufficiently small; see Theorem 1.2.1. Given a dual estimate $\overline{\mathbf{y}}_k \in \mathbb{R}^m$ and a penalty parameter $\mu_k > 0$, the (primal) subproblem solution $\mathbf{x}_k$, namely a minimizer of $\mathcal{L}_{\mu_k}(\cdot,\overline{\mathbf{y}}_k)$, satisfies

$$\mathbf{x}_k \in \operatorname{prox}_{\gamma_k g}\left(\mathbf{x}_k - \gamma_k\nabla_x\pi_{\mu_k}(\mathbf{x}_k,\overline{\mathbf{y}}_k)\right) \tag{1.4.14}$$

for some $\gamma_k > 0$ sufficiently small. We can find an update rule for the dual estimate, that is, a way to correct the estimate $\overline{\mathbf{y}}_k$ given $\mathbf{x}_k$, by comparing and matching (1.4.13) and (1.4.14). A different, more pragmatic, yet equivalent derivation is the one in [163, §4.1]. Let us denote $\hat{\mathbf{y}}$ a generic updated dual estimate such that the condition

$$\nabla_x\pi(\mathbf{x}_k,\hat{\mathbf{y}}) = \nabla_x\pi_{\mu_k}(\mathbf{x}_k,\overline{\mathbf{y}}_k) \tag{1.4.15}$$

is satisfied. Direct calculation based on (1.4.11) and (1.4.12) gives

$$\nabla f(\mathbf{x}_k) + \nabla\mathbf{c}(\mathbf{x}_k)^\top\hat{\mathbf{y}} = \nabla f(\mathbf{x}_k) + \nabla\mathbf{c}(\mathbf{x}_k)^\top\left[\overline{\mathbf{y}}_k + \frac{\mathbf{c}(\mathbf{x}_k) - \Pi_S(\mathbf{c}(\mathbf{x}_k) + \mu_k\overline{\mathbf{y}}_k)}{\mu_k}\right],$$

which guarantees the existence of such an update $\hat{\mathbf{y}}$, but also shows its uniqueness depends on the rank of $\nabla\mathbf{c}(\mathbf{x}_k)$. We choose

$$\hat{\mathbf{y}}_k = \hat{\mathbf{y}}(\mathbf{x}_k,\overline{\mathbf{y}}_k,\mu_k) := \overline{\mathbf{y}}_k + \frac{\mathbf{c}(\mathbf{x}_k) - \Pi_S(\mathbf{c}(\mathbf{x}_k) + \mu_k\overline{\mathbf{y}}_k)}{\mu_k}, \tag{1.4.16}$$

which is valid for any $\nabla\mathbf{c}(\mathbf{x}_k)$. This generalizes the classical first-order multiplier update to constraints expressed as in (1.1.1), cf. [62, 163]. However, since the projection onto S is possibly set-valued, so is the update $\hat{\mathbf{y}}_k$. Indeed, the expression above can be equivalently rewritten, by exposing its set-valuedness, as

$$\mathbf{z}_k \in \Pi_S(\mathbf{c}(\mathbf{x}_k) + \mu_k\overline{\mathbf{y}}_k), \qquad \hat{\mathbf{y}}_k = \overline{\mathbf{y}}_k + \frac{\mathbf{c}(\mathbf{x}_k) - \mathbf{z}_k}{\mu_k}.$$

These results can also be obtained by following the ALP approach and applying analogous arguments. It is clear that $\hat{\mathbf{y}}_k$ depends on the (arbitrary) choice of $\mathbf{z}_k$.

A by-product of the update rule presented above is that, by construction, the gradient of the (smooth) Lagrangian function (1.4.11) at the updated primal-dual pair coincides with the gradient of the (smooth) augmented Lagrangian function (1.4.12) at the subproblem solution, that is, (1.4.15) holds. As highlighted in [242], this means that the subproblem termination condition, based on the approximate satisfaction of (1.4.14), can be directly adopted for checking (1.4.13) for the outer, original problem.

1.4.2 Primal–dual approaches

The generalized augmented Lagrangian function has been proposed in [114, 149]. Several methods draw upon its primal-dual instantiation, which naturally leads to regularized and stabilized methods [158, 177, 183, 195, 238]. In the context of PAL methods, it has been employed as a primal-dual merit function in [194, 222] for convex structured optimization.

Following [114], one can replace the (primal) AL function $\mathcal{L}_\mu^z$ in (1.4.3) with the corresponding primal-dual AL function $\mathcal{M}_{\mu,\rho}^z$, defined by

$$\mathcal{M}_{\mu,\rho}^z(\mathbf{x},\mathbf{z},\mathbf{y},\overline{\mathbf{y}}) := \mathcal{L}_\mu^z(\mathbf{x},\mathbf{z},\mathbf{y}) + \frac{1}{2\rho}\|\mathbf{z}-\mathbf{c}(\mathbf{x})+\rho(\mathbf{y}-\overline{\mathbf{y}})\|^2.$$

for some $\rho > 0$. Furthermore, one can consider an additional proximal regularization for the primal variable, given an estimate $\overline{\mathbf{x}} \in \mathbb{R}^n$ and a penalty parameter $\sigma \geq 0$, which yields

$$\begin{aligned}\mathcal{M}_{\mu,\rho,\sigma}^z(\mathbf{x},\mathbf{z},\mathbf{y},\overline{\mathbf{x}},\overline{\mathbf{y}}) &:= \mathcal{M}_{\mu,\rho}^z(\mathbf{x},\mathbf{z},\mathbf{y},\overline{\mathbf{y}}) + \frac{\sigma}{2}\|\mathbf{x}-\overline{\mathbf{x}}\|^2 \\ &= f(\mathbf{x}) + g(\mathbf{x}) + \chi_S(\mathbf{z}) + \frac{\sigma}{2}\|\mathbf{x}-\overline{\mathbf{x}}\|^2 + \frac{1}{2\mu}\|\mathbf{z}-\mathbf{c}(\mathbf{x})-\mu\overline{\mathbf{y}}\|^2 \\ &\quad - \frac{\mu}{2}\|\overline{\mathbf{y}}\|^2 + \frac{1}{2\rho}\|\mathbf{z}-\mathbf{c}(\mathbf{x})+\rho(\mathbf{y}-\overline{\mathbf{y}})\|^2. \end{aligned} \tag{1.4.17}$$

As its name suggests, the primal-dual AL function is to be minimized, in each subproblem, with respect to both primal and dual variables. Therefore, parameters σ and ρ allow to control the deviation from the current primal and dual estimates, respectively, via a (proximal) quadratic penalty. Notice that $\mathcal{M}_{\mu,\rho,\sigma}^z$ falls back to $\mathcal{L}_\mu^z$ with $\rho \to +\infty$ and $\sigma = 0$.

Both the ALP and the PAL approaches are still valid, and similar pondering applies. In particular, the explicit minimization with respect to the auxiliary variable $\mathbf{z}$ can be performed and a dual estimate update rule can then be derived. These steps are covered in the rest of this section.

In order to expose the structure of (1.4.17) with respect to $\mathbf{z}$, we use the following identity, valid for $\alpha + \beta \neq 0$,

$$\alpha\|\mathbf{z}-\mathbf{a}\|^2 + \beta\|\mathbf{z}-\mathbf{b}\|^2 = (\alpha+\beta)\left\|\mathbf{z} - \frac{\alpha\mathbf{a}+\beta\mathbf{b}}{\alpha+\beta}\right\|^2 + \frac{\alpha\beta}{\alpha+\beta}\|\mathbf{a}-\mathbf{b}\|^2.$$

With $\alpha = 1/(2\mu)$, $\beta = 1/(2\rho)$, $\mathbf{a} = \mathbf{c}(\mathbf{x}) + \mu\overline{\mathbf{y}}$, and $\mathbf{b} = \mathbf{c}(\mathbf{x}) + \rho(\overline{\mathbf{y}} - \mathbf{y})$, this yields

$$\begin{aligned}\mathcal{M}_{\mu,\rho,\sigma}^z(\mathbf{x},\mathbf{z},\mathbf{y},\overline{\mathbf{x}},\overline{\mathbf{y}}) &= f(\mathbf{x}) + g(\mathbf{x}) + \chi_S(\mathbf{z}) - \frac{\mu}{2}\|\overline{\mathbf{y}}\|^2 + \frac{\sigma}{2}\|\mathbf{x}-\overline{\mathbf{x}}\|^2 \\ &\quad + \frac{\rho+\mu}{2\mu\rho}\left\|\mathbf{z}-\mathbf{c}(\mathbf{x}) - \frac{2\mu\rho}{\rho+\mu}(\overline{\mathbf{y}}-\mathbf{y}/2)\right\|^2 + \frac{1}{2(\rho+\mu)}\|(\mu-\rho)\overline{\mathbf{y}}+\rho\mathbf{y}\|^2.\end{aligned}$$

With this expression at hand, it is easy to obtain the minimizer of $\mathcal{M}_{\mu,\rho,\sigma}^z$ with respect to $\mathbf{z}$:

$$\begin{aligned}\mathbf{z}_{\mu,\rho}^{\mathcal{M}}(\mathbf{x},\mathbf{y},\overline{\mathbf{y}}) &:= \operatorname*{arg\,min}_{\mathbf{z}\in\mathbb{R}^m} \mathcal{M}_{\mu,\rho,\sigma}^z(\mathbf{x},\mathbf{z},\mathbf{y},\overline{\mathbf{x}},\overline{\mathbf{y}}) \\ &= \operatorname{prox}_{\frac{\mu\rho}{\rho+\mu}\chi_S}\left(\mathbf{c}(\mathbf{x}) + \frac{2\mu\rho}{\rho+\mu}(\overline{\mathbf{y}}-\mathbf{y}/2)\right) \\ &= \Pi_S\left(\mathbf{c}(\mathbf{x}) + \frac{2\mu\rho}{\rho+\mu}(\overline{\mathbf{y}}-\mathbf{y}/2)\right)\end{aligned}$$

Notably, $\mathbf{z}^{\mathcal{M}}_{\mu,\rho}$ does not depend on the primal estimate $\overline{\mathbf{x}}$ nor on the penalty parameter σ, and, although not identical, its expression closely resemble (1.4.8). Injecting $\mathbf{z}^{\mathcal{M}}_{\mu,\rho}$ into $\mathcal{M}^{z}_{\mu,\rho,\sigma}$ gives the counterpart of $\mathcal{L}_\mu$ in (1.4.10), namely the primal-dual PAL function $\mathcal{M}_{\mu,\rho,\sigma}$:

$$\begin{aligned}\mathcal{M}_{\mu,\rho,\sigma}(\mathbf{x},\mathbf{y},\overline{\mathbf{x}},\overline{\mathbf{y}}) &:= \mathcal{M}^{z}_{\mu,\rho,\sigma}\left(\mathbf{x}, \mathbf{z}^{\mathcal{M}}_{\mu,\rho}(\mathbf{x},\mathbf{y},\overline{\mathbf{y}}), \mathbf{y}, \overline{\mathbf{x}}, \overline{\mathbf{y}}\right)\\ &= f(\mathbf{x}) + g(\mathbf{x}) + \frac{\rho+\mu}{2\mu\rho}\operatorname{dist}^2_S\left(\mathbf{c}(\mathbf{x}) + \frac{2\mu\rho}{\rho+\mu}(\overline{\mathbf{y}} - \mathbf{y}/2)\right)\\ &\quad + \frac{\sigma}{2}\|\mathbf{x}-\overline{\mathbf{x}}\|^2 - \frac{\mu}{2}\|\overline{\mathbf{y}}\|^2 + \frac{1}{2(\rho+\mu)}\|(\mu-\rho)\overline{\mathbf{y}} + \rho\mathbf{y}\|^2\end{aligned}$$

By choosing $\rho = \mu$ as in [114, §3.7], these expressions simplify into

$$\mathbf{z}^{\mathcal{M}}_{\mu}(\mathbf{x},\mathbf{y},\overline{\mathbf{y}}) = \Pi_S\left(\mathbf{c}(\mathbf{x}) + \mu\,(\overline{\mathbf{y}} - \mathbf{y}/2)\right) \tag{1.4.18}$$

and

$$\begin{aligned}\mathcal{M}_{\mu,\sigma}(\mathbf{x},\mathbf{y},\overline{\mathbf{x}},\overline{\mathbf{y}}) = f(\mathbf{x}) + g(\mathbf{x}) + \frac{1}{\mu}\operatorname{dist}^2_S\left(\mathbf{c}(\mathbf{x}) + \mu\,(\overline{\mathbf{y}} - \mathbf{y}/2)\right)\\ + \frac{\sigma}{2}\|\mathbf{x}-\overline{\mathbf{x}}\|^2 - \frac{\mu}{2}\|\overline{\mathbf{y}}\|^2 + \frac{\mu}{4}\|\mathbf{y}\|^2.\end{aligned} \tag{1.4.19}$$

Comparing (1.4.19) to (1.4.10), we notice that the coefficient in front of the squared distance is doubled, for a given parameter $\mu > 0$. However, the primal PAL function $\mathcal{L}_\mu$ in (1.4.10) is easily recovered by setting $\mathbf{y} = \overline{\mathbf{y}}$ and adjusting μ.

Let us focus on the dual estimate update rule, which we expect to differ from (1.4.16) for the classical AL approach, since the dual solution $\mathbf{y}_k$ may play also a role. A solution $(\mathbf{x}_k, \mathbf{y}_k)$ of the primal-dual PAL subproblem necessarily satisfies a condition analogous to (1.4.14), which simplifies into

$$\mathbf{x}_k \in \operatorname{prox}_{\gamma_k g}\left(\mathbf{x}_k - \gamma_k \nabla_x \kappa_{\mu_k,\sigma_k}(\mathbf{x}_k,\mathbf{y}_k,\overline{\mathbf{x}}_k,\overline{\mathbf{y}}_k)\right) \tag{1.4.20a}$$

$$\mathbf{0} = \nabla_y \kappa_{\mu_k,\sigma_k}(\mathbf{x}_k,\mathbf{y}_k,\overline{\mathbf{x}}_k,\overline{\mathbf{y}}_k) \tag{1.4.20b}$$

with $\kappa_{\mu,\sigma}(\mathbf{x},\mathbf{y},\overline{\mathbf{x}},\overline{\mathbf{y}}) := \mathcal{M}_{\mu,\sigma}(\mathbf{x},\mathbf{y},\overline{\mathbf{x}},\overline{\mathbf{y}}) - g(\mathbf{x})$ the smooth part of $\mathcal{M}_{\mu,\sigma}$. Comparing and matching the first one with (1.4.13), we may insist on

$$\nabla_x \pi(\mathbf{x}_k, \check{\mathbf{y}}) = \nabla_x \kappa_{\mu_k,\sigma_k}(\mathbf{x}_k,\mathbf{y}_k,\overline{\mathbf{x}}_k,\overline{\mathbf{y}}_k)$$

to obtain a (primal-dual) dual estimate update $\check{\mathbf{y}}$, consistently with (1.4.15). For the sake of brevity, let us denote $\check{\mathbf{w}}_k := \mathbf{c}(\mathbf{x}_k) + \mu_k(\overline{\mathbf{y}}_k - \mathbf{y}_k/2)$ and $\check{\mathbf{z}}_k \in \Pi_S(\check{\mathbf{w}}_k)$. Then, direct calculation yields

$$\nabla f(\mathbf{x}_k) + \nabla \mathbf{c}(\mathbf{x}_k)^\top \check{\mathbf{y}} = \nabla f(\mathbf{x}_k) + \frac{2}{\mu_k}\nabla \mathbf{c}(\mathbf{x}_k)^\top\left[\check{\mathbf{w}}_k - \check{\mathbf{z}}_k\right] + \sigma_k(\mathbf{x}_k - \overline{\mathbf{x}}_k).$$

Neglecting the primal proximal regularization term, which should vanish on its own, there exists a (possibly nonunique) dual update $\check{\mathbf{y}}$ that satisfies the equality, independent on $\nabla\mathbf{c}(\mathbf{x}_k)$, namely

$$\begin{aligned}\check{\mathbf{y}}_k = \check{\mathbf{y}}(\mathbf{x}_k,\mathbf{y}_k,\overline{\mathbf{y}}_k,\mu_k) &:= \frac{2}{\mu_k}\left[\mathbf{c}(\mathbf{x}_k) + \mu_k(\overline{\mathbf{y}}_k - \mathbf{y}_k/2) - \check{\mathbf{z}}_k\right]\\ &= 2\left[\overline{\mathbf{y}}_k + \frac{\mathbf{c}(\mathbf{x}_k) - \check{\mathbf{z}}_k}{\mu_k}\right] - \mathbf{y}_k.\end{aligned} \tag{1.4.21}$$

From the second optimality condition above, namely

$$\mathbf{0} = -\left[\check{\mathbf{w}}_k - \check{\mathbf{z}}_k\right] + \mu_k\mathbf{y}_k/2 = \check{\mathbf{z}}_k - \mathbf{c}(\mathbf{x}_k) + \mu_k(\mathbf{y}_k - \overline{\mathbf{y}}_k),$$

one can formally solve for $\mathbf{y}_k$, since $\mu_k > 0$, obtaining

$$\mathbf{y}_k = \overline{\mathbf{y}}_k + \frac{\mathbf{c}(\mathbf{x}_k) - \check{\mathbf{z}}_k}{\mu_k}. \tag{1.4.22}$$

Since $\mathbf{y}_k$ coincides with the term in brackets, by (1.4.21) the primal-dual first-order multiplier estimate is just $\check{\mathbf{y}}_k = \mathbf{y}_k$. Although a result analogous to (1.4.21) was obtained in [114, p. 81] (for equality constraints only), it seems the last development was overlooked, based on the coupling with the second optimality condition. Nevertheless, it must be said that these relations are valid in the case the subproblems are solved exactly, and it is not clear the effect of solving them only approximately, as is usual practice.

1.4.3 Embedding simple constraints

Often some constraints are simple, in the sense that they can be easily satisfied, and hard, in the sense that they cannot be violated nor relaxed [163]. In such cases, the relaxation introduced by adopting a penalty function may induce an unacceptable violation of such constraints. Therefore, thanks to the (rich and flexible) structure of (1.1.1), these simple and hard constraints are better embedded into the optimization problem by considering a characteristic function in the objective. General lower-level constraints [87, 115] may help dealing with the greediness phenomenon and unbounded subproblem in the AL framework [131]. Simple, hard constraints should be embedded into the nonsmooth term g, in order to cope with them in a straightforward manner. This is illustrated in §1.6 with some examples.

1.5 Algorithm and Convergence

In this section we propose an algorithmic framework for solving NCSPs and assess its convergence properties. The development follows the classical AL framework, as in [163], and extends it to accommodate the general formulation in (1.1.1).

Algorithm 1 is a basic AL-type algorithm for solving NCSPs. It proceeds by minimizing the AL function $\mathcal{L}_\mu$ at each iteration, for fixed dual estimate and penalty parameter, and updating Lagrange multipliers and penalty parameters between iterations. Despite its generality and vagueness, it is possible to analyze several aspects under different conditions. Algorithm 1 is adapted from [163, Alg. 4.1], but similar, additional, or related features can be found in [49, 114, 145, 173, 196, 257].

Algorithm 1 Abstract Augmented Lagrangian Proximal algorithm

Input: $\mathbf{x}_0 \in \mathbb{R}^n$, $\mathbf{y}_0 \in \mathbb{R}^m$, $Y \subseteq \mathbb{R}^m$ compact, $\theta \in (0,1)$
  set $k \leftarrow 0$ and select $\mu_k > 0$
  while true **do**
    select $\overline{\mathbf{y}}_k \in Y$ ▷ *dual estimate*
    select $\epsilon_k > 0$ such that $\lim_{k\to+\infty} \epsilon_k = 0$
    find an ϵ_k-approximate minimizer $\mathbf{x}_k$ of $\mathcal{L}_{\mu_k}(\cdot, \overline{\mathbf{y}}_k)$, given in (1.4.10) ▷ *subproblem*
    select $\mathbf{z}_k \in \Pi_S\left(\mathbf{c}(\mathbf{x}_k) + \mu_k \overline{\mathbf{y}}_k\right)$
    set $\mathbf{y}_k \leftarrow \overline{\mathbf{y}}_k + [\mathbf{c}(\mathbf{x}_k) - \mathbf{z}_k]/\mu_k$
    set $C_k \leftarrow \|\mathbf{c}(\mathbf{x}_k) - \mathbf{z}_k\|$
    if $k = 0$, or $C_k \leq \theta C_{k-1}$ **then**
      set $\mu_{k+1} \leftarrow \mu_k$
    else
      select $\mu_{k+1} \in (0, \mu_k)$ such that $\lim_{k\to+\infty} \mu_k = 0$ ▷ *penalty update*
    end if
    update $k \leftarrow k + 1$
  end while

Some comments are in order. The dual estimate $\overline{\mathbf{y}}_k$ is selected from a compact set $Y \subseteq \mathbb{R}^m$, making the method safeguarded, possibly based on the previous update $\mathbf{y}_{k-1}$. Similarly, solving the AL subproblem may be warm-started at the previous solution $\mathbf{x}_{k-1}$. An element $\mathbf{z}_k$ of the set-valued projection is selected arbitrarily, but additional criteria could be considered. Then, the dual estimate update is found, according to §1.4.1, and the residual of the complementarity condition (1.2.3) in Theorem 1.2.1 is computed. This plays the role of test function, for assessing whether it is appropriate to strengthen the penalty term. Note also that special treatment is given to the case $k = 0$, since C_{-1} is undefined. However, this is not just for formal reasons: it allows to reset the penalty parameter, based on the first subproblem solution, which may improve the balance between objective and constraint violation. Moreover, following the analysis in [163, 211], we have left the term "ϵ_k-approximate minimizer" unspecified: in §1.5.1 and §1.5.2 we establish results in the case one seeks global minima or critical points for the subproblems, respectively.

Although many computational details are left ambiguous in Algorithm 1, it already provides a useful property concerning the Lagrange multiplier for inactive constraints.

Theorem 1.5.1. *Suppose the set S has nonempty interior. Let $\{\mathbf{x}_k\}$ be a sequence generated by Algorithm 1, $\mathbf{x}^\star$ be any limit point of $\{\mathbf{x}_k\}$ and $K \subset \mathbb{N}$ a subsequence such that $\lim_{k\in K} \mathbf{x}_k = \mathbf{x}^\star$. Then, if $\mathbf{c}(\mathbf{x}^\star) \in \operatorname{int} S$, it holds $\mathbf{y}_k = 0$ for all $k \in K$ sufficiently large.*

Proof. Let $\mathbf{c}(\mathbf{x}^\star) \in \operatorname{int} S \neq \emptyset$. Then, by continuity, for all sufficiently large $k \in K$ it holds $\mathbf{c}(\mathbf{x}_k) \in \operatorname{int} S$. There are two cases: either (i) the sequence $\{\mu_k\}$ tends to zero or (ii) it is bounded away from zero.

(i) Since $\mathbf{c}(\mathbf{x}_k) \in \operatorname{int} S$ for $k \in K$ sufficiently large and $\{\overline{\mathbf{y}}_k\}$ is bounded, we have $\mathbf{c}(\mathbf{x}_k)+\mu_k\overline{\mathbf{y}}_k \in \operatorname{int} S$ for sufficiently large $k \in K$.

(ii) Condition $C_k \leq \theta C_{k-1}$ holds for all $k \in K$ sufficiently large, and thus $\lim_{k\to\infty} C_k = 0$, since $\theta \in (0,1)$. This implies $\lim_{k\to\infty} \mu_k\overline{\mathbf{y}}_k = \mathbf{0}$, because it is $\mathbf{c}(\mathbf{x}_k) \in \operatorname{int} S$ for $k \in K$ sufficiently large, and the boundedness of $\{\mu_k\}$ away from zero gives $\lim_{k\to\infty} \overline{\mathbf{y}}_k = \mathbf{0}$. Hence, we have $\mathbf{c}(\mathbf{x}_k) + \mu_k\overline{\mathbf{y}}_k \in \operatorname{int} S$ for $k \in K$ sufficiently large.

Therefore, in both cases, for $k \in K$ sufficiently large it is $\mathbf{z}_k = \mathbf{c}(\mathbf{x}_k) + \mu_k\overline{\mathbf{y}}_k \in S$. Then, substituting and rearranging, it follows that $\mathbf{y}_k = \overline{\mathbf{y}}_k + [\mathbf{c}(\mathbf{x}_k) - \mathbf{z}_k]/\mu_k = \mathbf{0}$ for all $k \in K$ sufficiently large. □

The Lagrange multiplier vanishes for constraints that are inactive in the limit, independently of the feasibility of the limit point, whose existence is assumed. Notice that, if it is the case, Theorem 1.5.1 can be refined by exploiting the separable structure of S. For instance, considering the standard hyperbox $[\ell, \mathbf{u}] \subseteq \mathbb{R}^m$, the result applies componentwise, recovering the classical result [163, Thm. 4.1].

1.5.1 Global minimization of subproblems

This section investigates the convergence properties of Algorithm 1 considering the subproblems solved at global optimality. The discussion is based on [163, Ch. 5]. Let us work under the following assumption:

Assumption 1.5.2. *Let $\{\epsilon_k\}$ be a bounded sequence of nonnegative tolerances. For all $k \in \mathbb{N}$, it holds $\mathcal{L}_{\mu_k}(\mathbf{x}_k, \overline{\mathbf{y}}_k) \leq \mathcal{L}_{\mu_k}(\mathbf{x}, \overline{\mathbf{y}}_k) + \epsilon_k$ for all $\mathbf{x} \in \mathbb{R}^n$.*

Notice that the bounded tolerances need not be small in principle. The following result shows that limit points of sequences generated by Algorithm 1 tend to minimize infeasibility, namely the constraint violation. Therefore, if problem (1.1.1) is feasible, minimal infeasibility corresponds to feasibility, and thus limit points are feasible.

Theorem 1.5.3. *Let $\{\mathbf{x}_k\}$ be a sequence generated by Algorithm 1 under Assumption 1.5.2. Let $\mathbf{x}^\star$ be a limit point of $\{\mathbf{x}_k\}$ and $K \subset \mathbb{N}$ a subsequence such that $\lim_{k\in K} \mathbf{x}_k = \mathbf{x}^\star$. Then, it holds $\operatorname{dist}_S^2(\mathbf{c}(\mathbf{x}^\star)) \leq \operatorname{dist}_S^2(\mathbf{c}(\mathbf{x}))$ for all $\mathbf{x} \in \mathbb{R}^n$.*

Proof. There are two cases: either (i) the sequence $\{\mu_k\}$ is bounded away from zero or (ii) it tends to zero.

(i) Condition $C_k \leq \theta C_{k-1}$ must hold for all sufficiently large k. Thus, it is $\lim_{k\to\infty} C_k = 0$, since $\theta \in (0,1)$. This implies that $\mathbf{c}(\mathbf{x}_k)$ approaches the set S, since $\mathbf{z}_k \in S$ for all k. Then, the limit point $\mathbf{x}^\star$ is feasible. By nonnegativity of the distance and the fact that $\operatorname{dist}_S(\mathbf{c}(\mathbf{x}^\star)) = 0$, the result is obtained.

(ii) Let us assume there exists $\mathbf{x} \in \mathbb{R}^n$ such that $\operatorname{dist}_S^2(\mathbf{c}(\mathbf{x}^\star)) > \operatorname{dist}_S^2(\mathbf{c}(\mathbf{x}))$. In such case, by continuity of $\mathbf{c}$ and dist_S^2, boundedness of $\{\overline{\mathbf{y}}_k\}$, and $\{\mu_k\} \to 0$, there exists $\xi > 0$ such that $\operatorname{dist}_S^2\left(\mathbf{c}(\mathbf{x}_k) + \mu_k\overline{\mathbf{y}}_k\right) >$

$\operatorname{dist}_S^2\left(\mathbf{c}(\mathbf{x}) + \mu_k\overline{\mathbf{y}}_k\right) + \xi$ for all sufficiently large $k \in K$. This implies that

$$\begin{aligned}
\mathcal{L}_{\mu_k}(\mathbf{x}_k, \overline{\mathbf{y}}_k) &= \varphi(\mathbf{x}_k) + \frac{1}{2\mu_k}\operatorname{dist}_S^2\left(\mathbf{c}(\mathbf{x}_k) + \mu_k\overline{\mathbf{y}}_k\right) - \frac{\mu_k}{2}\|\overline{\mathbf{y}}_k\|^2 \\
&> \varphi(\mathbf{x}_k) + \frac{1}{2\mu_k}\operatorname{dist}_S^2\left(\mathbf{c}(\mathbf{x}) + \mu_k\overline{\mathbf{y}}_k\right) - \frac{\mu_k}{2}\|\overline{\mathbf{y}}_k\|^2 + \frac{1}{2\mu_k}\xi \\
&= \mathcal{L}_{\mu_k}(\mathbf{x}, \overline{\mathbf{y}}_k) + \frac{1}{2\mu_k}\xi + \varphi(\mathbf{x}_k) - \varphi(\mathbf{x}) \\
&> \mathcal{L}_{\mu_k}(\mathbf{x}, \overline{\mathbf{y}}_k) + \epsilon_k
\end{aligned}$$

for all sufficiently large $k \in K$, since $\{\mathbf{x}_k\} \to \mathbf{x}^\star$, φ is continuous, $\xi > 0$, $\{\mu_k\} \to 0$, and $\{\epsilon_k\}$ is bounded. This contrasts with Assumption 1.5.2, proving the result by contradiction. □

The following results shows that feasible limit points are global minimizers, exposing a powerful feature of the AL framework. If the unconstrained (or simply constrained) subproblems are solved to global optimality, a global minimizer is to be expected for the original constrained (feasible) problem.

Theorem 1.5.4. *Let $\{\mathbf{x}_k\}$ be a sequence generated by Algorithm 1 under Assumption 1.5.2 with $\lim_{k\to\infty}\epsilon_k = 0$. Suppose that problem (1.1.1) is feasible. Let $\mathbf{x}^\star$ be a limit point of $\{\mathbf{x}_k\}$ and $K \subset \mathbb{N}$ a subsequence such that $\lim_{k\in K}\mathbf{x}_k = \mathbf{x}^\star$. Then, $\mathbf{x}^\star$ is a global minimizer.*

Proof. By Theorem 1.5.3, feasibility of the problem implies that $\mathbf{x}^\star$ is feasible. Let $\mathbf{x} \in \mathbb{R}^n$ be an arbitrary feasible point. By Assumption 1.5.2, for all $k \in \mathbb{N}$ it holds $\mathcal{L}_{\mu_k}(\mathbf{x}_k, \overline{\mathbf{y}}_k) \leq \mathcal{L}_{\mu_k}(\mathbf{x}, \overline{\mathbf{y}}_k) + \epsilon_k$, which implies (the second line of)

$$\begin{aligned}
\varphi(\mathbf{x}_k) &\leq \varphi(\mathbf{x}_k) + \frac{1}{2\mu_k}\operatorname{dist}_S^2\left(\mathbf{c}(\mathbf{x}_k) + \mu_k\overline{\mathbf{y}}_k\right) && (1.5.1)\\
&\leq \varphi(\mathbf{x}) + \frac{1}{2\mu_k}\operatorname{dist}_S^2\left(\mathbf{c}(\mathbf{x}) + \mu_k\overline{\mathbf{y}}_k\right) + \epsilon_k \\
&\leq \varphi(\mathbf{x}) + \frac{\mu_k}{2}\|\overline{\mathbf{y}}_k\|^2 + \epsilon_k, && (1.5.2)
\end{aligned}$$

where the first line is due to the nonnegativity of dist_S^2 and the last line comes from the following inequality, which holds for any feasible $\mathbf{x} \in \mathbb{R}^n$ and any $\mathbf{v} \in \mathbb{R}^m$:

$$\begin{aligned}
\operatorname{dist}_S^2\left(\mathbf{c}(\mathbf{x}) + \mathbf{v}\right) &= \|\mathbf{c}(\mathbf{x}) + \mathbf{v} - \Pi_S\left(\mathbf{c}(\mathbf{x}) + \mathbf{v}\right)\|^2 \\
&\leq \|\mathbf{c}(\mathbf{x}) + \mathbf{v} - \mathbf{c}(\mathbf{x})\|^2 = \|\mathbf{v}\|^2. && (1.5.3)
\end{aligned}$$

Herein, the inequality is due to the minimal distance of the projection. There are two cases: either (i) the sequence $\{\mu_k\}$ tends to zero or (ii) it is bounded away from zero.

(i) By continuity of φ, boundedness of $\{\overline{\mathbf{y}}_k\}$, $\{\mu_k\} \to 0$, and $\{\epsilon_k\} \to 0$, taking the limit for $k \in K$, we get

$$\varphi(\mathbf{x}^\star) \leq \varphi(\mathbf{x}), \tag{1.5.4}$$

which proves the result, since $\mathbf{x}$ is an arbitrary feasible point.

(ii) Condition $C_k \leq \theta C_{k-1}$ holds for all $k \in K$ sufficiently large, and thus $\lim_{k\to\infty} C_k = 0$, since $\theta \in (0,1)$. This yields $\lim_{k\to\infty}\mathbf{c}(\mathbf{x}_k) - \Pi_S\left(\mathbf{c}(\mathbf{x}_k) + \mu_k\overline{\mathbf{y}}_k\right) = \mathbf{0}$ and, by continuity, we have

$$\begin{aligned}
\lim_{k\to\infty}\operatorname{dist}_S^2\left(\mathbf{c}(\mathbf{x}_k) + \mu_k\overline{\mathbf{y}}_k\right) &= \lim_{k\to\infty}\|\mathbf{c}(\mathbf{x}_k) + \mu_k\overline{\mathbf{y}}_k - \Pi_S\left(\mathbf{c}(\mathbf{x}_k) + \mu_k\overline{\mathbf{y}}_k\right)\|^2 \\
&= \lim_{k\to\infty}\|\mu_k\overline{\mathbf{y}}_k\|^2 && (1.5.5)
\end{aligned}$$

Finally, comparing (1.5.1) and (1.5.2), taking the limit for $k \in K$, substituting (1.5.5) and rearranging, we get (1.5.4), concluding the proof. □

It is interesting to investigate the behaviour of Algorithm 1 for infeasible problems. In such cases, Theorem 1.5.3 guarantees that sequences generated by Algorithm 1 converge to (global) minimizers of the infeasibility. However, as Theorem 1.5.4 does not apply, we are left with no guarantees on the quality of the limit points in terms of the objective function.

Let us consider the classical penalty method, introduced in §1.3, with the subproblems solved at global optimality. Then, intuitively, we expect the objective to be minimized once the minimal infeasibility is reached, since this minimizes the merit function which combines objective function and constraint violation. Indeed, this holds true, as shown and exploited in the following. On the other hand, the introduction of nonzero shifts, as in §1.3.1, modifies the merit function, and thus may perturb the limit points. This suggests it is recommendable to avoid shifting the constraints when the problem is, or is detected to be, infeasible. We further argue to support the benefits of resetting the dual estimate. By Algorithm 1, infeasible problems lead to sequence $\{C_k\}$ being bounded away from zero and thus $\{\mu_k\} \to 0$. By boundedness of $\{\overline{\mathbf{y}}_k\}$, the multiplier update (1.4.16) is expected to return extremely large values for $\mathbf{y}_k$ as μ_k approaches zero. In contrast, for difficult yet feasible problems, as $\mu_k \to 0$ we expect $C_k \to 0$ and $\mathbf{y}_k$ to remain bounded. Therefore, the extreme growth of the multiplier update $\mathbf{y}_k$ can be adopted as a proxy to detect problem infeasibility. If this is the case, shifting constraints can be deactivated and the algorithm is expected to fall back to the classical penalty method, which returns a minimizer of the objective function subject to minimal infeasibility [173]. On the other hand, as noticed in §1.3, with a strong penalty term it barely makes sense to shift the constraints (unless the exact one is known, perhaps). These motivate the idea of resetting the dual estimate, possibly temporarily, as established in the following assumption.

Assumption 1.5.5. *For all $k \in \mathbb{N}$, if $\mathbf{y}_k \notin Y$, then $\overline{\mathbf{y}}_{k+1} = \mathbf{0}$.*

Furthermore, in the case $\{\mu_k\} \to 0$, one can interpret Assumption 1.5.5 as a mechanism to achieve finite convergence of $\mu_k \overline{\mathbf{y}}_k$ to zero, leading to a safeguarded method and recovering the classical penalty method [196, 173]. The following result is adapted from [163, Thm. 5.3].

Theorem 1.5.6. *Let $\{\mathbf{x}_k\}$ be a sequence generated by Algorithm 1 under Assumptions 1.5.2 and 1.5.5 with $\lim_{k\to\infty} \epsilon_k = 0$. Let $\mathbf{x}^\star$ be a limit point of $\{\mathbf{x}_k\}$ and $K \subset \mathbb{N}$ a subsequence such that $\lim_{k\in K} \mathbf{x}_k = \mathbf{x}^\star$. Then, $\varphi(\mathbf{x}') \geq \varphi(\mathbf{x}^\star)$ for all $\mathbf{x}' \in \mathbb{R}^n$ such that $\operatorname{dist}_S^2(\mathbf{c}(\mathbf{x}')) = \operatorname{dist}_S^2(\mathbf{c}(\mathbf{x}^\star))$.*

Proof. By Theorem 1.5.3, the limit point $\mathbf{x}^\star$ is a (global) minimizer of the infeasibility, therefore it is $\operatorname{dist}_S^2(\mathbf{c}(\mathbf{x}_k)) \geq \operatorname{dist}_S^2(\mathbf{c}(\mathbf{x}^\star))$ for all $k \in \mathbb{N}$. If the limit point $\mathbf{x}^\star$ is feasible, the result follows immediately from Theorem 1.5.4.

Let us focus on the infeasible case, *i.e.*, when $\operatorname{dist}_S^2(\mathbf{c}(\mathbf{x}^\star)) > 0$. This implies that $\{C_k\}$ is bounded away from zero and, consequently, it must be $\lim_{k\to\infty} \mu_k = 0$. Let $\mathbf{x} \in \mathbb{R}^n$ be arbitrary. Rearranging the inequality given in Assumption 1.5.2, it holds

$$\varphi(\mathbf{x}_k) \leq \varphi(\mathbf{x}) + \epsilon_k + \frac{1}{2\mu_k}\left[\operatorname{dist}_S^2\left(\mathbf{c}(\mathbf{x}) + \mu_k\overline{\mathbf{y}}_k\right) - \operatorname{dist}_S^2\left(\mathbf{c}(\mathbf{x}_k) + \mu_k\overline{\mathbf{y}}_k\right)\right] \tag{1.5.6}$$

for all $k \in \mathbb{N}$. By boundedness of $\{\overline{\mathbf{y}}_k\}$, boundedness of $\{C_k\}$ away from zero, and the fact that $\{\mu_k\} \to 0$, the update (1.4.16) generates $\mathbf{y}_{k+1} \notin Y$ for sufficiently large $k \in \mathbb{N}$. Hence, by Assumption 1.5.5, it is $\overline{\mathbf{y}}_k = \mathbf{0}$ for sufficiently large $k \in \mathbb{N}$. This is sufficient, along with the fact that $\operatorname{dist}_S^2(\mathbf{c}(\mathbf{x}')) = \operatorname{dist}_S^2(\mathbf{c}(\mathbf{x}^\star)) \leq \operatorname{dist}_S^2(\mathbf{c}(\mathbf{x}_k))$ for all $k \in \mathbb{N}$, for (1.5.6) to imply that

$$\varphi(\mathbf{x}_k) \leq \varphi(\mathbf{x}') + \epsilon_k$$

for sufficiently large $k \in \mathbb{N}$. Taking the limit for $k \in K$, this gives $\varphi(\mathbf{x}^\star) \leq \varphi(\mathbf{x}')$, and the result follows from the arbitrariness of $\mathbf{x}'$. □

1.5.2 Affordable minimization of subproblems

Let us focus on the AL framework with the subproblems solved via affordable methods, which are designed to converge to mere stationary (or critical) points, not necessarily global minima [163, Ch. 6]. In this context, we consider the following assumption [202].

Assumption 1.5.7. *Suppose g is prox-bounded with threshold γ_g. For all $k \in \mathbb{N}$, let L_k denote the Lipschitz constant of the gradient of $\pi_{\mu_k}(\cdot, \bar{\mathbf{y}}_k)$, see (1.4.12). Let $\{\epsilon_k\}$ be a bounded sequence of nonnegative tolerances. Then, for all $k \in \mathbb{N}$, it holds $\|\mathbf{x}_k - \mathbf{u}_k\|/\gamma_k \leq \epsilon_k$ for some $\mathbf{u}_k \in \operatorname{prox}_{\gamma_k g}\left(\mathbf{x}_k - \gamma_k \nabla_x \pi_{\mu_k}(\mathbf{x}_k, \bar{\mathbf{y}}_k)\right)$ and $\gamma_k \in (0, \min\{1/L_k, \gamma_g\})$.*

Remark 1.5.8. The condition given in Assumption 1.5.7 can be employed as a stopping criterion in structured optimization, and it deserves some comments. In fact, structured optimization solvers usually return an element of the proximal mapping, namely $\mathbf{u}_k$, as solution, instead of $\mathbf{x}_k$. The reason for this is twofold: (i) if the nonsmooth term represents some constraints, then $\mathbf{u}_k$ is guaranteed to be feasible, because projected onto the feasible set; (ii) for suitable values of the stepsize γ_k, every proximal gradient step yields some sufficient decrease in the objective function [200, 214], hence $\mathbf{u}_k$ improves the objective over $\mathbf{x}_k$. For purely smooth problems, that is, omitting $g = 0$, condition $\|\mathbf{x}_k - \mathbf{u}_k\|/\gamma_k \leq \epsilon_k$ boils down to $\|\nabla_x \mathcal{L}_{\mu_k}(\mathbf{x}_k, \bar{\mathbf{y}}_k)\| \leq \epsilon_k$, which is a classical termination condition for unconstrained smooth problems. This, however, evaluates the gradient at $\mathbf{x}_k$, and not at $\mathbf{u}_k$, as one would expect. Adopted in [257], the condition

$$\left\| \frac{\mathbf{x}_k - \mathbf{u}_k}{\gamma_k} + \nabla_x \pi_{\mu_k}(\mathbf{u}_k, \bar{\mathbf{y}}_k) - \nabla_x \pi_{\mu_k}(\mathbf{x}_k, \bar{\mathbf{y}}_k) \right\| \leq \epsilon_k$$

overcomes this mismatch. Nonetheless, although this seems a more robust and consistent condition, its theoretical properties are not discussed in [257]. Indeed, its practical consequences are still unclear.

The next theorem shows that, considering sequences generated by Algorithm 1, feasible limit points satisfy the (sequential) optimality conditions given in Theorem 1.2.1. This result is strongly related to the AKKT conditions [136]; see also [163, Thm. 6.2].

Theorem 1.5.9. *Let $\{\mathbf{x}_k\}$ be a sequence generated by Algorithm 1 under Assumption 1.5.7 with $\lim_{k\to\infty} \epsilon_k = 0$. Let $\mathbf{x}^\star$ be a feasible limit point of $\{\mathbf{x}_k\}$ and $K \subset \mathbb{N}$ a subsequence such that $\lim_{k\in K} \mathbf{x}_k = \mathbf{x}^\star$. Then, $\mathbf{x}^\star$ satisfies the (sequential) necessary optimality conditions given in Theorem 1.2.1.*

Proof. Condition (1.2.1) holds by assumption. Feasibility of $\mathbf{x}^\star$ and continuity of $\mathbf{c}$ yield (1.2.2). Considering Assumption 1.5.7, injecting (1.4.15), and taking the limit yields

$$\lim_{k\in K} \left\| \frac{\mathbf{x}_k - \operatorname{prox}_{\gamma_k g}\left(\mathbf{x}_k - \gamma_k \nabla_x \pi(\mathbf{x}_k, \mathbf{y}_k)\right)}{\gamma_k} \right\| = 0 \tag{1.5.7}$$

since $\{\epsilon_k\} \to 0$. This, by boundedness of $\{\gamma_k\}$, gives (1.2.4). Let us focus on (1.2.3). The particular case $\mathbf{c}(\mathbf{x}^\star) \in \operatorname{int} S$ is covered by Theorem 1.5.1. It remains to show that it holds also in the case $\mathbf{c}(\mathbf{x}^\star) \in \partial S$. Denoting $\mathbf{p}_k \in \Pi_S(\mathbf{c}(\mathbf{x}_k))$, $\mathbf{w}_k = \mathbf{c}(\mathbf{x}_k) + \mu_k \bar{\mathbf{y}}_k$, and $\mathbf{z}_k \in \Pi_S(\mathbf{w}_k)$, the dual update rule (1.4.16) reads $\mathbf{y}_k = (\mathbf{w}_k - \mathbf{z}_k)/\mu_k$. There are two cases: either (i) the sequence $\{\mu_k\}$ is bounded away from zero or (ii) it tends to zero.

(i) By boundedness of $\{\mu_k\}$ away from zero, condition $C_k \leq \theta C_{k-1}$ must hold for all sufficiently large $k \in K$, and thus $\lim_{k\in K} C_k = 0$, since $\theta \in (0, 1)$. Therefore, it is $\lim_{k\in K} \|\mathbf{c}(\mathbf{x}_k) - \mathbf{z}_k\| = 0$ and, by properties of norms, $\lim_{k\in K} \mathbf{c}(\mathbf{x}_k) - \mathbf{z}_k = \mathbf{0}$. Thus,

$$\begin{aligned} \lim_{k\in K} \|\mathbf{c}(\mathbf{x}_k) - \Pi_S\left(\mathbf{c}(\mathbf{x}_k) + \mu_k \mathbf{y}_k\right)\| &= \lim_{k\in K} \|\mathbf{c}(\mathbf{x}_k) - \Pi_S\left(\mathbf{c}(\mathbf{x}_k) + \mathbf{w}_k - \mathbf{z}_k\right)\| \\ &= \lim_{k\in K} \|\mathbf{c}(\mathbf{x}_k) - \Pi_S\left(\mathbf{w}_k\right)\| \\ &= \lim_{k\in K} \|\mathbf{c}(\mathbf{x}_k) - \mathbf{z}_k\| = 0. \end{aligned}$$

(ii) Since $\{\mu_k\} \to 0$, by boundedness of $\{\bar{\mathbf{y}}_k\}$, it is $\lim_{k\in K} \mathbf{c}(\mathbf{x}_k) - \mathbf{w}_k = \mathbf{0}$ and, consequently,

$\lim_{k\in K} \mathbf{z}_k - \mathbf{p}_k = \mathbf{0}$. By feasibility of $\mathbf{x}^\star$, it is $\lim_{k\to\infty} \mathbf{c}(\mathbf{x}_k) - \mathbf{p}_k = \mathbf{0}$. Thus,

$$\begin{aligned}\lim_{k\in K} \|\mathbf{c}(\mathbf{x}_k) - \Pi_S\left(\mathbf{c}(\mathbf{x}_k) + \mu_k \mathbf{y}_k\right)\| &= \lim_{k\in K} \|\mathbf{c}(\mathbf{x}_k) - \Pi_S\left(\mathbf{c}(\mathbf{x}_k) + \mathbf{w}_k - \mathbf{z}_k\right)\| \\ &= \lim_{k\in K} \|\mathbf{c}(\mathbf{x}_k) - \Pi_S\left(\mathbf{c}(\mathbf{x}_k) + \mathbf{c}(\mathbf{x}_k) - \mathbf{p}_k\right)\| \\ &= \lim_{k\in K} \|\mathbf{c}(\mathbf{x}_k) - \Pi_S\left(\mathbf{c}(\mathbf{x}_k)\right)\| \\ &= \lim_{k\in K} \|\mathbf{c}(\mathbf{x}_k) - \mathbf{p}_k\| = 0.\end{aligned}$$

In both cases, since $K \subset \mathbb{N}$ is an arbitrary subsequence, (1.2.3) is satisfied and the proof is complete. □

Dropping the feasibility assumption, a similar, yet local, counterpart of Theorem 1.5.3 can be obtained. The following theorem establishes that infeasible limit points are (local) minimizers of the infeasibility. Interestingly, there is no need to assume that $\lim_{k\to\infty} \epsilon_k = 0$.

Theorem 1.5.10. *Let $\{\mathbf{x}_k\}$ be a sequence generated by Algorithm 1 under Assumption 1.5.7. Let $\mathbf{x}^\star$ be a limit point of $\{\mathbf{x}_k\}$ and $K \subset \mathbb{N}$ a subsequence such that $\lim_{k\in K} \mathbf{x}_k = \mathbf{x}^\star$. Then, $\mathbf{x}^\star$ satisfies the (sequential) necessary optimality conditions, given in Theorem 1.2.1, for the problem*

$$\operatorname*{minimize}_{\mathbf{x}\in\mathbb{R}^n} \quad \Phi(\mathbf{x}) := \frac{1}{2}\operatorname{dist}_S^2(\mathbf{c}(\mathbf{x})).$$

Proof. There are two cases: (i) the sequence $\{\mu_k\}$ is bounded away from zero or (ii) it tends to zero.

(i) Condition $C_k \leq \theta C_{k-1}$ must hold for all sufficiently large k, and thus $\lim_{k\to\infty} C_k = 0$, since $\theta \in (0,1)$. Therefore, the limit point $\mathbf{x}^\star$ is feasible, since $\mathbf{z}_k \in S$ for all k. By nonnegativity of the distance and the fact that $\operatorname{dist}_S(\mathbf{c}(\mathbf{x}^\star)) = 0$, $\mathbf{x}^\star$ is a (global) minimizer of Φ and necessarily satisfies the conditions given in Theorem 1.2.1.

(ii) Let us denote, for all $k \in \mathbb{N}$, the residual

$$\mathbf{r}_k := \frac{\mathbf{x}_k - \operatorname{prox}_{\gamma_k g}\left(\mathbf{x}_k - \gamma_k \nabla_x \pi_{\mu_k}\left(\mathbf{x}_k, \overline{\mathbf{y}}_k\right)\right)}{\gamma_k}.$$

By Assumption 1.5.7 and boundedness of $\{\epsilon_k\}$, the residual norm $\|\mathbf{r}_k\|$ is bounded too. Hence, $\{\mu_k\} \to 0$ implies that $\{\mu_k \mathbf{r}_k\} \to \mathbf{0}$. Since $\gamma_k \in (0, \min\{\gamma_g, 1/L_k\})$ from Assumption 1.5.7, it is $\gamma_k = O(\mu_k)$ as $\mu_k \to 0$. Therefore, $\{\gamma_k\} \to 0$ and, by continuity of g and definition of the proximal operator, $\mathbf{r}_k = O\left(\nabla_x \pi_{\mu_k}\left(\mathbf{x}_k, \overline{\mathbf{y}}_k\right)\right)$ as $\mu_k \to 0$. Consequently, we have $\lim_{k\in K} \mu_k \nabla_x \pi_{\mu_k}\left(\mathbf{x}_k, \overline{\mathbf{y}}_k\right) = \lim_{k\in K} \mu_k \mathbf{r}_k = \mathbf{0}$. By boundedness of $\{\nabla f(\mathbf{x}_k)\}$, $\{\nabla \mathbf{c}(\mathbf{x}_k)\}$, and $\{\overline{\mathbf{y}}_k\}$, and (1.4.12), after trivial simplifications, it is

$$\mathbf{0} = \lim_{k\in K} \nabla \mathbf{c}(\mathbf{x}_k)^\top \left[\mathbf{c}(\mathbf{x}_k) - \Pi_S(\mathbf{c}(\mathbf{x}_k))\right] = \lim_{k\in K} \nabla \Phi(\mathbf{x}_k).$$

Since K is an arbitrary subsequence and the feasibility problem is unconstrained, this proves the result. □

The following result, not provided in [163], extends Theorem 1.5.6 to affordable algorithms. This means that, with a resetting safeguard on the Lagrange multiplier, limit points are critical points subject to minimal infeasibility.

Theorem 1.5.11. *Let $\{\mathbf{x}_k\}$ be a sequence generated by Algorithm 1 under Assumptions 1.5.5 and 1.5.7 with $\lim_{k\to\infty} \epsilon_k = 0$. Let $\mathbf{x}^\star$ be a limit point of $\{\mathbf{x}_k\}$ and $K \subset \mathbb{N}$ a subsequence such that $\lim_{k\in K} \mathbf{x}_k = \mathbf{x}^\star$. Then, $\mathbf{x}^\star$ satisfies the (sequential) necessary optimality conditions of the problem*

$$\operatorname*{minimize}_{\mathbf{x}\in\mathbb{R}^n} \quad f(\mathbf{x}) + g(\mathbf{x}) \qquad \text{subject to} \quad \operatorname{dist}_S^2(\mathbf{c}(\mathbf{x})) = \operatorname{dist}_S^2(\mathbf{c}(\mathbf{x}^\star)).$$

Proof. By Theorem 1.5.10, any limit point $\mathbf{x}^\star$ is a stationary point of the infeasibility. Therefore, for any feasible limit point $\mathbf{x}^\star$, the result follows immediately from Theorem 1.5.9.

Let us focus on an infeasible limit point $\mathbf{x}^\star$. Since $\operatorname{dist}_S^2(\mathbf{c}(\mathbf{x}^\star)) > 0$, $\{C_k\}$ is bounded away from zero and, consequently, $\{\mu_k\}$ tends to zero. By boundedness of $\{\overline{\mathbf{y}}_k\}$, boundedness of $\{C_k\}$ away from zero, and the fact that $\{\mu_k\} \to 0$, the dual update rule (1.4.16) generates $\mathbf{y}_{k+1} \notin Y$ for $k \in K$ sufficiently large. Thus, by Assumption 1.5.5, it is $\overline{\mathbf{y}}_k = \mathbf{0}$ for sufficiently large $k \in K$. Let us define, for all $\mathbf{x} \in \mathbb{R}^n$, the smooth function $\xi(\mathbf{x}) := \operatorname{dist}_S^2(\mathbf{c}(\mathbf{x})) - \operatorname{dist}_S^2(\mathbf{c}(\mathbf{x}^\star))$ and denote $\xi(\mathbf{x}) = 0$ the scalar constraint in the problem above. Then, $\nabla\xi(\mathbf{x}) = 2\nabla\mathbf{c}(\mathbf{x})^\top[\mathbf{c}(\mathbf{x}) - \Pi_S(\mathbf{c}(\mathbf{x}))]$ and

$$\begin{aligned}\nabla_x \pi_{\mu_k}(\mathbf{x}_k, \mathbf{0}) &= \nabla f(\mathbf{x}_k) + \mu_k^{-1}\nabla\mathbf{c}(\mathbf{x}_k)^\top[\mathbf{c}(\mathbf{x}_k) - \Pi_S(\mathbf{c}(\mathbf{x}_k))] \\ &= \nabla f(\mathbf{x}_k) + \eta_k \nabla\xi(\mathbf{x}_k),\end{aligned}$$

where $\eta_k := 1/(2\mu_k) > 0$. Therefore, by Assumption 1.5.7 and $\{\epsilon_k\} \to 0$, it is

$$\lim_{k\in K} \left\| \frac{\mathbf{x}_k - \operatorname{prox}_{\gamma_k g}\left(\mathbf{x}_k - \gamma_k\left[\nabla f(\mathbf{x}_k) + \eta_k\nabla\xi(\mathbf{x}_k)\right]\right)}{\gamma_k} \right\| = 0$$

for some $\gamma_k > 0$ sufficiently small, which proves (1.2.4). Since $S = \{0\}$, (1.2.3) boils down to $\lim_{k\to\infty} \|\xi(\mathbf{x}_k)\| = 0$. Consequently, by continuity of ξ and $\xi(\mathbf{x}^\star) = 0$, both (1.2.2) and (1.2.3) are satisfied, concluding the proof. □

1.5.3 Boundedness of the penalty parameter

The theoretical results presented so far are valid for any (safeguarded) choice of the dual estimate $\overline{\mathbf{y}}_k$. This comprises even the case that the Lagrange multiplier is safeguarded by the trivial choice $\overline{\mathbf{y}}_k = \mathbf{0}$ for all $k \in \mathbb{N}$, which corresponds to the (classical, external) penalty method [20], in which the constraints are not shifted. However, stronger results are available in NLP when the dual estimate is effectively updated employing the strategy derived in §1.4.1, possibly subject to safeguards [49, 62, 163]. Similarly, we expect additional properties to hold in NCSP under similar assumptions, that is, for feasible, sufficiently regular limit points. Employing the following dual update rule, there exists a limit point for the sequence of dual variable $\{\mathbf{y}_k\}$.

Assumption 1.5.12. *For all $k \in \mathbb{N}$, if $\mathbf{y}_k \in Y$, then $\overline{\mathbf{y}}_{k+1} = \mathbf{y}_k$.*

Proposition 1.5.13. *Let $\{\mathbf{x}_k\}$ be a sequence generated by Algorithm 1 under Assumptions 1.5.5 and 1.5.12. Let $\mathbf{x}^\star$ be a limit point of $\{\mathbf{x}_k\}$ and $K \subset \mathbb{N}$ a subsequence such that $\lim_{k\in K} \mathbf{x}_k = \mathbf{x}^\star$. Then, if $\{\mu_k\}$ is bounded away from zero, the sequence $\{\mathbf{y}_k\}$ admits a limit point $\mathbf{y}^\star$.*

Proof. By boundedness of $\{\mu_k\}$ away from zero, it is $\lim_{k\in K} C_k = 0$. Therefore, by properties of norms, $\lim_{k\in K} \mathbf{c}(\mathbf{x}_k) - \mathbf{z}_k = \mathbf{0}$. Thus, by (1.4.16), $\lim_{k\in K} \mathbf{y}_k - \overline{\mathbf{y}}_k = \mathbf{0}$. By Assumptions 1.5.5 and 1.5.12, compactness of Y yields the result. □

Furthermore, we argue, it is reasonable to expect the boundedness of $\{\mu_k\}$ away from zero, possibly under additional assumptions. This is in fact a desirable feature since the difficulty of solving subproblems increases with the penalty parameter approaching zero. Although the penalty term introduces a benign ill-conditioning [94], gradient-based methods may struggle, and Newton-type methods are needed to overcome this issue. Quasi-Newton methods for structured optimization, such as `ZeroFPR` [215] and `PANOC` [202], may also be able to effectively cope with severely ill-conditioned problems.

The Augmented Lagrangian method has the property of converging to the correct solution maintaining bounded penalty parameters, under mild assumption [163, §7.8]. Thus, in practice, this means that the occurrence of extremely large penalty parameters is a symptom of infeasibility.

1.5.4 Subproblem convergence

A fundamental step in Algorithm 1, and perhaps the most computationally demanding, is the (approximate) minimization of the AL function; of course, this plays a key role in any declination of the AL framework. Independent on the approach chosen, either PAL or ALP, the arising subproblem is a structured optimization problem, with a proper, lower semi-continuous, nonconvex objective function; see §1.4. Quasi-Newton methods for such class of problems have been only recently developed, pioneered by the work of Stella, Themelis, and Patrinos [200, 214]. In the following we build upon `ZeroFPR` [215] and `PANOC` [202], two linesearch-based quasi-Newton methods for structured optimization. Although they can cope with fully nonconvex problems, they require only simple algebraic operations, have a small memory footprint, and can exploit second-order information to enable asymptotic superlinear rates under mild assumptions. The interested reader may refer to [202, 215, 229] for a detailed convergence analysis of such methods.

Algorithm 2 is designed for solving the structured problem

$$\operatorname*{minimize}_{\mathbf{x}\in\mathbb{R}^n} \varphi(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x}).$$

For the sake of a concise notation, we denote by $\overline{\mathbf{x}}_\gamma$ a proximal-gradient step, by $\mathbf{r}_\gamma$ the forward-backward residual (related to Assumption 1.5.7), by $\check{f}_\gamma$ the quadratic model of the smooth term f, defined by

$$\overline{\mathbf{x}}_\gamma(\mathbf{x}) \in \operatorname{prox}_{\gamma g}(\mathbf{x} - \gamma\nabla f(\mathbf{x})), \tag{1.5.8}$$

$$\mathbf{r}_\gamma(\mathbf{x}) := \mathbf{x} - \overline{\mathbf{x}}_\gamma(\mathbf{x}), \tag{1.5.9}$$

$$\check{f}_\gamma(\mathbf{x}) := f(\mathbf{x}) - \nabla f(\mathbf{x})^\top \mathbf{r}_\gamma(\mathbf{x}) + \frac{1}{2\gamma}\|\mathbf{r}_\gamma(\mathbf{x})\|^2, \tag{1.5.10}$$

respectively. The FBE φ_γ of $\varphi := f + g$, defined in (0.2.4), can be expressed as

$$\begin{aligned}
\varphi_\gamma(\mathbf{x}) &:= \inf_{\mathbf{z}\in\mathbb{R}^n} \left\{ f(\mathbf{x}) + \nabla f(\mathbf{x})^\top(\mathbf{z}-\mathbf{x}) + \frac{1}{2\gamma}\|\mathbf{z}-\mathbf{x}\|^2 + g(\mathbf{z}) \right\} \\
&= \inf_{\mathbf{z}\in\mathbb{R}^n} \left\{ f(\mathbf{x}) - \frac{\gamma}{2}\|\nabla f(\mathbf{x})\|^2 + \frac{1}{2\gamma}\|\mathbf{z}-\mathbf{x}+\gamma\nabla f(\mathbf{x})\|^2 + g(\mathbf{z}) \right\} \\
&= f(\mathbf{x}) + \nabla f(\mathbf{x})^\top(\overline{\mathbf{x}}_\gamma(\mathbf{x})-\mathbf{x}) + \frac{1}{2\gamma}\|\overline{\mathbf{x}}_\gamma(\mathbf{x})-\mathbf{x}\|^2 + g(\overline{\mathbf{x}}_\gamma(\mathbf{x})) \\
&= \check{f}_\gamma(\mathbf{x}) + g\left(\overline{\mathbf{x}}_\gamma(\mathbf{x})\right).
\end{aligned} \tag{1.5.11}$$

Finally, we define the sufficient decrease parameter as

$$\sigma_\gamma := \alpha_\sigma(1-\alpha_\gamma)/(2\gamma). \tag{1.5.12}$$

Although Algorithm 2 closely resembles `PANOC`, originally proposed in [202], it should be complemented with comments regarding two aspects: the possibly set-valued gradient of the smooth objective, and the conditions on the forward-backward stepsize γ.

Following the PAL strategy, for a nonconvex set S, the gradient (of the smooth part) of the AL function $\mathcal{L}_\mu(\cdot, \overline{\mathbf{y}})$ is set-valued, due to the projection Π_S. Whenever an iterate exhibits set-valued gradient, however, one could switch to the ALP formulation, which yields a continuously differentiable smooth term, and then proceed. In fact, this is possible because the two approaches stem from, and aim at solving, the same subproblem. Even better, on the vein of (1.2.5) and (1.2.12), it suffices to pick any element of the set-valued projection and proceed without further concern. In fact, this is formally equivalent to switching strategy and taking a step for the ALP subproblem.

The stepsize γ plays a crucial role in forward-backward splitting (FBS) algorithms and, similarly to gradient methods, it depends on the Lipschitz constant of the gradient. However, no prior knowledge of such Lipschitz constant is required in practice, as a simple backtracking procedure can select suitable

Algorithm 2 PANOC: Proximal averaged Newton-type method, based on [202]

Input: $\mathbf{x}_0 \in \mathbb{R}^n$, $\gamma > 0$, $\alpha_\gamma, \alpha_\sigma, \beta_\gamma, \beta_\tau \in (0,1)$, $\epsilon > 0$.
Output: $\mathbf{x}^\star$

```
for k = 0, 1, ... do
    while f(x̄_γ(x_k)) ≥ f̆_γ(x_k) do
        update γ ← β_γ γ                                   ▷ γ-backtracking
    end while
    if ‖r_γ(x_k)‖ ≤ γε then
        return x̄_γ(x_k)
    end if
    select d_k ∈ ℝ^n and set τ ← 1                         ▷ search direction
    while true do
        set x_{k+1} ← (1 − τ)x̄_γ(x_k) + τ(x_k + d_k)       ▷ tentative update
        if f(x̄_γ(x_{k+1})) < f̆_γ(x_{k+1}) and φ_γ(x_{k+1}) ≤ φ_γ(x_k) − σ_γ‖r_k‖² then
            break
        end if
        update τ ← β_τ τ                                  ▷ τ-backtracking
    end while
end for
```

values for γ in finitely many iterations [202, Rem. III.4]. In fact, this is the goal of the γ-backtracking loop in Algorithm 2: the value of γ is repeatedly decreased until the quadratic upper bound is satisfied. Diverging from [202, 255, 257], this condition is additionally checked within the linesearch procedure. The rationale behind this feature is related to increased robustness. Firstly, it is worth noticing that the value of the FBE $\varphi_\gamma(\mathbf{x}_{k+1})$ at the tentative update $\mathbf{x}_{k+1}$ depends on the current stepsize γ. Therefore, the sufficient decrease condition alone is unreliable as a termination criterion for the linesearch. This is particularly relevant for ill-condition problems, where the gradient can greatly vary between different regions, and robust to very large search directions. On the other hand, having these two termination criteria does not affect the convergence guarantees: by Lipschitz continuity of the gradient of the smooth term, suitability of γ at the current iterate, sufficient decrease given by the proximal gradient step, and continuity of the FBE, there always exists a linesearch stepsize $\tau > 0$ yielding sufficient decrease within the range of validity of the quadratic upper bound, cf. [202, 215].

1.5.5 Algorithm

This section takes a step toward an implementable algorithm for NCSP, based on Algorithm 1. Although not trivially derived from it, Algorithm 3 maintains its essential structure and retains its convergence guarantees, while offering additional features which make it more practical and improve its efficiency.

Let us comment on Algorithm 3 and compare it to Algorithm 1.

- Convergence to an approximate solution is assessed at every iteration, by checking the conditions given in Theorem 1.5.14 below. This result provides theoretical support for stopping Algorithm 3 and declaring convergence, based on Theorem 1.2.1.

- Although not necessary, the inner solver should be warm-started from the previous solution. Providing $\mathbf{x}_{k-1}$ as an initial guess for the k-th subproblem can greatly improve performance and robustness of the method.

- The dual safeguard Y needs not be constant, as in Algorithm 1, nor bounded; see [114, Alg. 4.2.1] for expanding safeguards. Nonetheless, in Algorithm 3, Y_k is assumed compact for all k, for the sake of simplicity. Moreover, the dual resetting mechanism is included, based on Assumption 1.5.5, so to benefit from Theorem 1.5.11.

Algorithm 3 ALPX: Augmented Lagrangian ProXimal method

Input: $\mathbf{x}_0 \in \mathbb{R}^n$, $\mathbf{y}_0 \in \mathbb{R}^m$, $\epsilon_{\text{opt}}, \epsilon_{\text{feas}} > 0$, $\{Y_k\} \subseteq \mathbb{R}^m$ compact, $\theta \in (0,1)$.
Output: $\mathbf{x}^\star$, $\mathbf{y}^\star$

```
select ȳ_1 ∈ Y_1 and μ_1 > 0
for k = 1, 2, ... do
    select ε_k > 0 such that ε_k ≤ ε_opt for k sufficiently large
    find an ε_k-approximate minimizer x_k of L_{μ_k}(·, ȳ_k), given in (1.4.10)      ▷ subproblem
    select z_k ∈ Π_S (c(x_k) + μ_k ȳ_k)
    set y_k ← ȳ_k + [c(x_k) − z_k] / μ_k
    if (x_k, y_k) is a (ε_opt, ε_feas)-approximate solution then
        return x_k, y_k
    end if
    if y_k ∈ Y_k then set ȳ_{k+1} ← y_k, else set ȳ_{k+1} ← 0, end if              ▷ dual estimate
    select z_k^+ ∈ Π_S (c(x_k) + μ_k y_k)
    set C_k ← ‖c(x_k) − z_k‖, F_k ← dist_S(c(x_k)), and V_k ← ‖c(x_k) − z_k^+‖
    select μ°_{k+1} > 0 such that lim_{k→+∞} μ°_k = 0
    if k = 1 or C_k ≤ θ C_{k−1} or max{V_k, F_k} ≤ ε_feas then
        select μ_{k+1} > 0
    else
        select μ_{k+1} ∈ (0, μ°_{k+1}]                                            ▷ penalty update
    end if
end for
```

- In contrast to Algorithm 1, iterations are deemed *successful* [49, 114, 145] considering also an alternative condition, which enlarges their scope. Sufficient progress, namely $C_k \leq \theta C_{k-1}$, is no longer necessarily required for iterates characterized by approximate feasibility and complementarity, namely $\max\{V_k, F_k\} \leq \epsilon_{\text{feas}}$. This strategy is recommended in [145, Alg. 2.2] and reminiscent of the switching condition in [113, §2.3]. Its rationale is that, in the latter case, there is no need to tighten the penalty term, since the current iterate is satisfactory in terms of primal feasibility and, indeed, it may be difficult to get further progress. This possibly limits the ill-conditioning of subproblems, as the penalty parameter is decreased fewer times.

- The penalty parameter update is non-monotone, as suggested in [145]. By relaxing the penalization term after successful iterations, this yields subproblems with better scaling, which the inner solver is expected to solve more easily. Yet, the (slowly) decreasing sequence $\{\mu_k^\circ\}$ guarantees that, if needed, the sequence $\{\mu_k\}$ decays to zero, preventing the method from cycling.

- The requirement that $\lim_{k\to\infty} \epsilon_k = 0$ in Algorithm 1 is relaxed, by considering the optimality tolerance $\epsilon_{\text{opt}} > 0$. This is possible thanks to the identity (1.4.15) given by the dual update rule (1.4.16).

Borrowing terminology from [145], we say the k-th iteration is *incomplete* if the subsolver returns a solution $\mathbf{x}_k$ which does not satisfy the condition given in Assumption 1.5.7, namely $\mathbf{x}_k$ is not an ϵ_k-approximate solution to the k-th subproblem. It should be noticed that, in exact arithmetic, reasonable solvers are guaranteed to find a critical point, and so to yield a complete iteration, unless the subproblem is unbounded from below. This case represents a pathology inherent in the AL framework [131], and effective strategies to detect and handle it should be implemented. However, in practical computations, incomplete iterations may appear due to difficult or slow progress, especially due to poor scaling when the penalty parameter approaches zero. Incomplete iterations may deserve particular care in practice. An interesting approach is discussed in [145, Alg. 2.3], that explicitly addresses possible failures of the subproblem minimization solver. Different strategies could be adopted depending on the failure cause, as well as on the outer loop history.

Theorem 1.5.14 below provides suitable termination criteria for Algorithm 3 and their theoretical

grounds, based on Theorem 1.2.1: *convergence* is declared as soon as these conditions hold, see [145, Thm. 2.3]. Notice that, despite no assumption regarding the problem feasibility is made, we emphasize it is itself a necessary condition and, in particular, (primal) infeasibility prevents the second condition to hold. On the contrary, convergence cannot be declared if infinitely many incomplete iterations take place, the penalty parameter vanishes, or both. This points out the need for additional termination criteria in practical implementations, such as, *e.g.*, maximum number of iterations, maximum elapsed time, detection of (local) infeasibility and unboundedness [113, 163].

Theorem 1.5.14. *Let $\epsilon_{opt}, \epsilon_{feas} > 0$ be arbitrary and $\{\mathbf{x}_k\}$, $\{\mathbf{y}_k\}$ denote sequences generated by Algorithm 3.*

(i) If a finite number of iterations is incomplete, then for all $k \in \mathbb{N}$ sufficiently large it holds

$$\|\mathbf{x}_k - \operatorname{prox}_{\gamma_k g}\left(\mathbf{x}_k - \gamma_k \left[\nabla f(\mathbf{x}_k) + \nabla \mathbf{c}(\mathbf{x}_k)^\top \mathbf{y}_k\right]\right)\| \leq \gamma_k \epsilon_{opt}$$

for some suitable $\gamma_k > 0$.

(ii) If $\{\mu_k\}$ is bounded away from zero, then it holds

$$\begin{aligned} \operatorname{dist}_S(\mathbf{c}(\mathbf{x}_k)) &\leq \epsilon_{feas} \\ \|\mathbf{c}(\mathbf{x}_k) - \Pi_S(\mathbf{c}(\mathbf{x}_k) + \mu_k \mathbf{y}_k)\| &\leq \epsilon_{feas} \end{aligned}$$

for all $k \in \mathbb{N}$ sufficiently large.

Proof. (i) Since at most a finite number of iterations is incomplete, there exists $k_0 \in \mathbb{N}$ such that the condition in Assumption 1.5.7 holds for all $k \geq k_0$. Then, by using the dual update (1.4.16), the equality (1.4.15), the gradient of $\pi(\cdot, \mathbf{y}_k)$ in (1.4.11), and the fact that eventually $\epsilon_k \leq \epsilon_{\text{opt}}$, the result follows.

(ii) Since $\{\mu_k\}$ is bounded away from zero, there exists $k_0 \in \mathbb{N}$ such that, for all $k \geq k_0$, either $C_k \leq \theta C_{k-1}$ or $\max\{V_k, F_k\} \leq \epsilon_{\text{feas}}$ holds. It is sufficient to consider the two cases separately. The latter case gives trivially the result, by definition of V_k and F_k. In the former case, it is $\{C_k\} \to 0$, since $\theta \in (0,1)$, and thus $\{\mathbf{c}(\mathbf{x}_k) - \mathbf{z}_k\} \to \mathbf{0}$. Since $\mathbf{z}_k \in S$ for all k and, by the dual update (1.4.16), $\{\mathbf{y}_k - \overline{\mathbf{y}}_k\} \to \mathbf{0}$, by continuity both conditions are satisfied for k sufficiently large. □

Some further comments are in order. A peculiar feature of the AL framework is that, in its outer layer, it is matrix-free. This property is becoming of particular interest to cope with huge-scale problems [132, 179, 205, 249]. Employing quasi-Newton directions in `PANOC` and `ZeroFPR` [202, 215], *e.g.*, via L-BFGS [38], Algorithm 3 can be readily made matrix-free.

Infeasibility detection is a valuable feature for numerical solvers [113, 165, 242, 232]. A simple yet effective approach is to consider a criterion related to the convergence to an infeasible point. Given some tolerances $\epsilon_{\text{feas}}^{\text{inf}}, \epsilon_{\text{opt}}^{\text{inf}} > 0$, one may consider the stationarity of the infeasibility measure $\Phi(\mathbf{x}) := \operatorname{dist}_S^2(\mathbf{c}(\mathbf{x}))$. If it holds

$$\|\mathbf{x}_k - \Pi_\Omega(\mathbf{x}_k - \nabla \Phi(\mathbf{x}_k))\| \leq \epsilon_{\text{opt}}^{\text{inf}} \qquad \text{and} \qquad \operatorname{dist}_S(\mathbf{c}(\mathbf{x})) \geq \epsilon_{\text{feas}}^{\text{inf}},$$

then the iterative method may stall at the infeasible point $\mathbf{x}_k$, based on Theorems 1.5.10 and 1.5.11. However, as discussed in [163, §10.2.3], it is not obvious whether this condition should be included or not as a termination criterion. If a problem is indeed infeasible, the execution should be stopped, as soon as possible, by enabling this check. On the other hand, if the problem is instead feasible, successive iterates may move away and eventually find a solution in the sense of Theorem 1.2.1, that is, a point $\mathbf{x}_k$ that satisfies the conditions given in Theorem 1.5.14.

1.5.6 Parameter selection

Penalty and AL methods require selecting values for some parameter and forcing sequences, *e.g.*, μ_0, θ, and $\{\epsilon_k\}$ in Algorithm 3. Great effort has been devoted to the design of tuning-free algorithms,

which inspired the use of trust regions [48, 83], filters [89], flexible penalties [118], and trust funnels [133], among others. Within this work, however, we focus on the simple, yet effective, Algorithm 3. Therefore, we specify how the forcing sequences are generated; cf. [46, 114, 145, 163].

The sequence $\{\epsilon_k\}$ of inner tolerances plays an important role, in that it balances the number of inner and outer iterations. Let $\epsilon_{\text{opt}}, \epsilon_{\text{feas}} \in (0,1)$ be arbitrary and given. Following [145], we take $\epsilon_1 = \sqrt{\epsilon_{\text{opt}}}$ and generate successive values depending on the iterates. If the k-th iteration is complete and $\max\{V_k, F_k\} \leq \sqrt{\epsilon_{\text{feas}}}$, then

$$\epsilon_{k+1} = \max\left\{\alpha_\epsilon \epsilon_{\text{opt}}, \min\left\{\beta_\epsilon \epsilon_k, \beta_r \|\mathbf{r}_k\|_\infty\right\}\right\},$$

otherwise we set $\epsilon_{k+1} = \epsilon_k$. Here, $\mathbf{r}_k := [\mathbf{x}_k - \overline{\mathbf{x}}_{\gamma_k}(\mathbf{x}_k)]/\gamma_k$ denotes the subproblem residual (see §1.5.4 and Assumption 1.5.7), and $\alpha_\epsilon \in (0,1]$, $\beta_\epsilon, \beta_r \in (0,1)$ are given parameters.

As mentioned above, multiple penalty parameters are considered, for better scaling of the constraint violations. For some $k \in \mathbb{N}$, let $\mathbf{x}_k \in \Omega$, $\mathbf{c}_k = \mathbf{c}(\mathbf{x}_k)$, $\mathbf{p}_k \in \Pi_S(\mathbf{c}_k)$, and $\varphi_k = f(\mathbf{x}_k) + g(\mathbf{x}_k)$. Based on balancing objective and constraint violation [163, Eq. 12.1], we take

$$\mu_k^i := \mu_{\text{est}} \frac{\max\left\{1, (\mathbf{c}_{k-1}^i - \mathbf{p}_{k-1}^i)^2/2\right\}}{\max\{1, |\varphi_{k-1}|\}}$$

for some $\mu_{\text{est}} > 0$. Smaller values of μ_{est} emphasize constraint violation and may yield faster convergence, whereas larger values result in easier, possibly better scaled, subproblems. If approximate feasibility and complementarity hold, namely $\max\{V_k, F_k\} \leq \epsilon_{\text{feas}}$, we relax the penalization by setting $\mu_{k+1} \leftarrow \min\{\alpha_+ \mu_k, \mu_{\max}\}$, with $\alpha_+ \geq 1$ and $\mu_{\max} > 0$. If there is only sufficient progress, namely $C_k \leq \theta C_{k-1}$, the penalization is left untouched, *i.e.* $\mu_{k+1} \leftarrow \mu_k$. For unsuccessful iterations, we simply take $\mu_{k+1} = \min\{\alpha_- \mu_k, \mu_{k+1}^\circ\}$, with $\alpha_- \in (0,1)$. The zero sequence $\{\mu_k^\circ\}$ is generated by having $\mu_k^\circ := \alpha_\circ^k \mu_{\max}$, with $\alpha_\circ \in (0,1)$.

It was recently highlighted in [228, §3.4] that having independent penalty parameters for the different constraints may lead to poorer convergence guarantees. Nevertheless, we believe the penalization of each constraint should be balanced, *i.e.*, provide a trade-off between objective function and constraint violation. In this perspective, the use of multiple penalty parameters, one attached to each constraint, can effectively scale the constraint violations. The relative magnitude is estimated a few times following [163, §12.4], *e.g.*, $k \in \{1, 2\}$, and then scaled all accordingly. Therefore, we can rely on the convergence analysis based on a single penalty parameter, without further modifications.

Finally, let us consider the dual safeguards Y_k, which play a role in controlling the growth of the dual variable, see [114, 196]. For the sake of simplicity, we construct each Y_k to be a bounded hyperbox in $\mathbb{R}^m$ containing the origin $\{\mathbf{0}\}$. However, we allow this hyperbox to expand, inspired by [114, Alg. 4.2.1]. If a dual update falls outside Y_k, the bounds corresponding to the overstepped faces are increased, by a factor $\beta_y \geq 1$ and up to a given maximum size $Y_{\max}$. This procedure is applied componentwise, as well as the reset of the dual estimate.

1.6 Numerical Results

This section presents details of our implementation of Algorithm 3, named `ALPX`, and reports on numerical evaluations. Some examples involving vanishing and disjunctive constraints are discussed, highlighting the advantages of formulating these problems as NCSPs. Then, `ALPX` is benchmarked against the open-source interior-point solver `IPOPT` [113] on a suite of NLPs.

Implementation We implemented Algorithm 3 in `Julia` during the writing of this thesis, see §0.1. The code is freely available within the open-source `Bazinga` package, which collects `ALPX` and other tools for constrained, structured optimization, such as `PANOC` and `ZeroFPR`, among others. Another package, called `OptiMo`, was developed as a modelling tool for NCSPs that generates problem formulations in a format suitable for `Bazinga`.

In the spirit of matrix-free methods [179, 249], `ALPX` does not require to explicitly form the constraint Jacobian $\nabla \mathbf{c}$, but only the transposed Jacobian-vector product is accessed, as an oracle. In particular,

Table 1.1: Default parameters of ALPX, PANOC, and ZeroFPR.

Parameter	Value	Parameter	Value
ALPX			
ϵ_{opt}	10^{-8}	ϵ_{feas}	10^{-8}
θ	0.5	ϵ_0	$\sqrt{\epsilon_{\text{opt}}}$
α_ϵ	1	β_ϵ	0.1
β_r	0.5	$\mu_{\max}$	10^6
μ_{est}	0.1	α_-	0.25
α_+	10	$\alpha_\circ$	0.99
Y_0	$[-10^3, 10^3]$	$Y_{\max}$	$[-10^6, 10^6]$
β_y	1.1	max iterations	50
PANOC and ZeroFPR			
α_γ	0.95	α_σ	0.5
β_γ	0.5	β_τ	0.5
$\gamma_{\min}$	10^{-16}	$\tau_{\min}$	10^{-8}
L-BFGS memory	8	max iterations	1000

this concerns the gradient evaluation of the augmented Lagrangian (1.4.12): the constraint Jacobian $\nabla \mathbf{c}$ should not be formed if m is large and if the transposed Jacobian-vector product can be computed more efficiently. Moreover, the subproblems solvers, PANOC and ZeroFPR, are equipped with search directions based on L-BFGS, a limited-memory quasi-Newton method, with the two-loop recursion [38]. Table 1.1 reports the default settings adopted in the numerical evaluations. Notice that parameters $\gamma_{\min}$ and $\tau_{\min}$ are employed in our implementation: the solvers exit the linesearch procedure with $\tau = 0$, *i.e.*, taking a proximal-gradient step, when the linesearch stepsize τ is decreased below $\tau_{\min}$; also, they abort as soon as $\gamma < \gamma_{\min}$, as progressing becomes difficult.

1.6.1 Illustrative examples

Disjunctive programming problems [206] can be cast into the form of (1.1.1). We now consider a few exemplary problems to illustrate some of the advantages offered by modeling problems in the form of (1.1.1). We give detailed reformulations of two examples of mathematical program with vanishing constraints (MPVCs) and a program with either-or constraints.

Academic MPVC Let us consider a two dimensional problem arising in truss topology optimization [64], a classical example in the context of MPVCs [124, 240]. The variables $\mathbf{x} \in \mathbb{R}^2$ represent cross sectional areas of two different groups of truss bars and the meaning of the objective function is the weight of the structure. The problem reads

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^2}{\text{minimize}} \quad & 4x_1 + 2x_2 \\ \text{subject to} \quad & x_1 \geq 0, \quad x_2 \geq 0 \\ & (x_1 + x_2 - 5\sqrt{2})x_1 \geq 0 \\ & (x_1 + x_2 - 5)x_2 \geq 0. \end{aligned} \tag{1.6.1}$$

The origin $\mathbf{x}^\circ = \mathbf{0}$ is the unique global minimizer of the problem, and $\mathbf{x}^\star = (0, 5)$ is a local minimizer. However, numerical evidence have shown that, due to lack of constraint qualification, solvers may end up also at $\mathbf{x}^+ = (0, 5\sqrt{2})$, which is not a local minimizer. This happens because of the geometry of the feasible set; see Figure 1.1. We reformulate (1.6.1) as a NCSP by considering (1.1.1) with the following

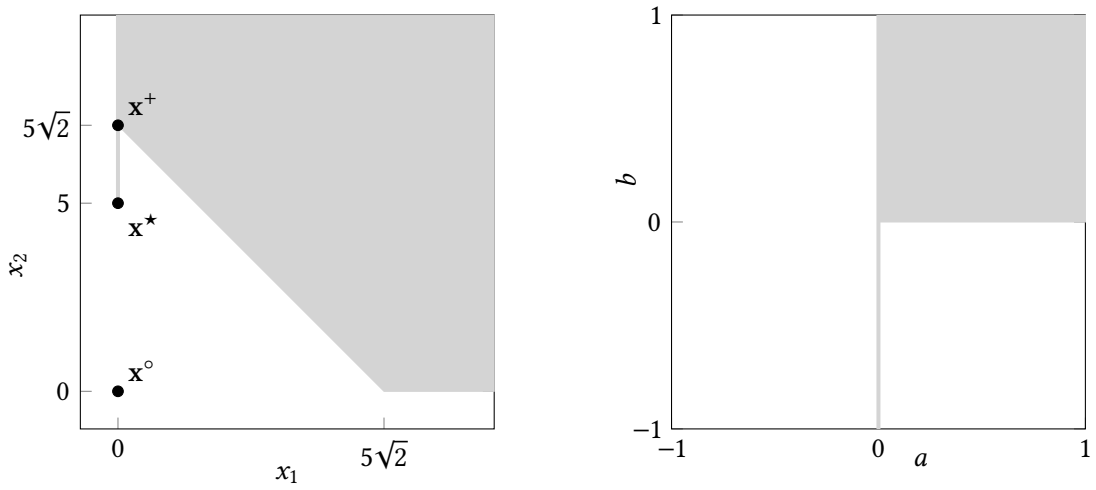


Figure 1.1: Feasible set of problem (1.6.1) (left) and (projection of) set S in (1.6.2) (right).

terms:

$$f(\mathbf{x}) := 4x_1 + 2x_2 \qquad g(\mathbf{x}) := \chi_{\mathbb{R}_+}(x_1) + \chi_{\mathbb{R}_+}(x_2) \tag{1.6.2}$$

$$\mathbf{c}(\mathbf{x}) := \begin{pmatrix} x_1 \\ x_2 \\ x_1 + x_2 - 5\sqrt{2} \\ x_1 + x_2 - 5 \end{pmatrix} \qquad S := \{(\mathbf{a}, \mathbf{b}) \in \mathbb{R}^4 \mid \forall i \in [1; 2]\ a_i = 0 \vee (a_i \geq 0 \wedge b_i \geq 0)\}.$$

This seems to be a fairly unusual formulation. In fact, MPVCs are often solved adopting smoothing, regularization, and continuation techniques [124, 152, 240]. Conversely, the reformulation associated with (1.6.2) has no regularization nor relaxation. Instead, the set S is nonconvex and its projection operator Π_S is not single-valued, but can be evaluated efficiently; see Figure 1.1.

We run ALPX, with the default settings, starting from a uniform grid of 2601 initial points in $[-5, 20]^2$. Computational results are summarized in Table 1.2 and displayed in Figure 1.2, where initial points are marked according to the solution found (within 10^{-3}). ALPX always returns either the global minimizer $\mathbf{x}^\circ$ or the local minimizer $\mathbf{x}^\star$, depending on the initial guess. In [240, §4.1], the Authors compare four different regularization methods, obtaining the global minimizer only 13–15% of the times (with 676 initial points).

Recalling the practical background of problem (1.6.1), we artificially exclude the point $\mathbf{x}^\circ = \mathbf{0}$ from the feasible set, as trusses must have a positive cross sectional area. Following [124, §9.5.1], we add the linear constraint $x_1 + x_2 \geq 3$ appending a smooth constraint, namely

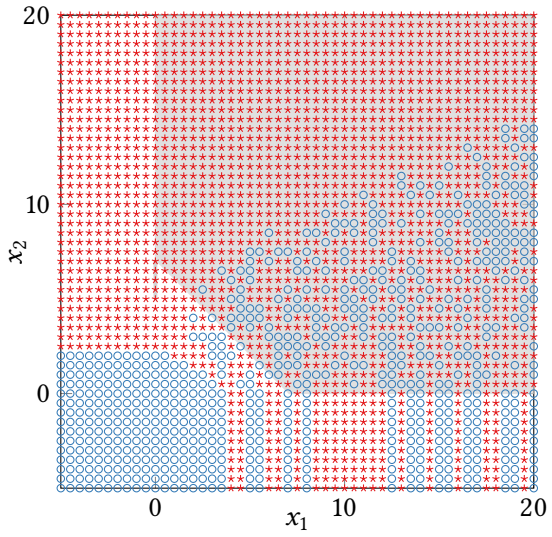
$$c_5(\mathbf{x}) := x_1 + x_2, \qquad S_5 := [3, +\infty). \tag{1.6.3}$$

As shown in Figure 1.2b, ALPX is not able to return a feasible solution for some initial guesses. Nevertheless, for all other cases, the (global, now) minimizer $\mathbf{x}^\star$ is found. In fact, with this problem formulation, the penalty-based method is not able to escape the infeasible region, for some initial points, and thus remains stuck with $x_1 + x_2 < 3$.

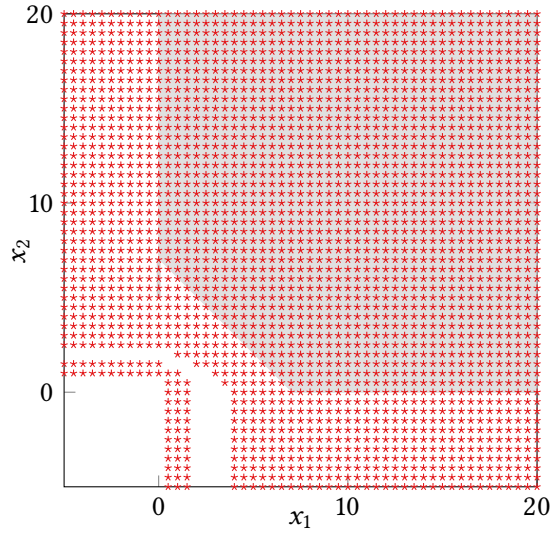
However, another formulation is possible, thanks to the rich structure of (1.1.1). In fact, the linear constraint can be enforced exactly by replacing g in (1.6.2) with

$$g(\mathbf{x}) := \chi_{\mathbb{R}_+}(x_1) + \chi_{\mathbb{R}_+}(x_2) + \chi_{\mathbb{R}_+}(x_1 + x_2 - 3), \tag{1.6.4}$$

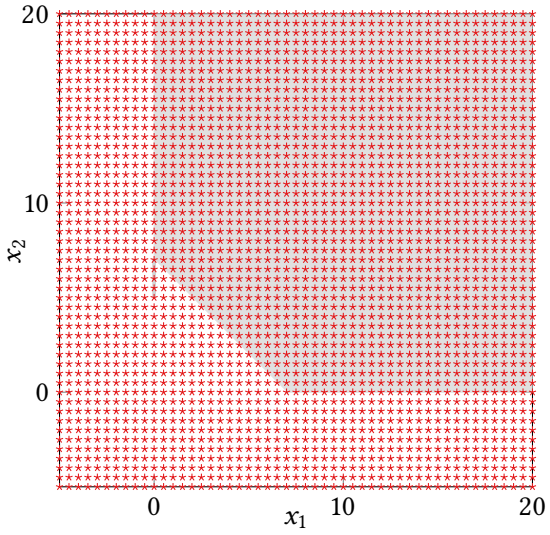
whose proximal operator can be easily evaluated, due to its convex, simple structure. In this case, the additional constraint $x_1 + x_2 \geq 3$ is satisfied at each and every iteration, as well as the nonnegativity of $\mathbf{x}$. Thus, the shifted penalty method is left with the vanishing constraints only. Numerical evidence shows that ALPX always finds the desired solution $\mathbf{x}^\star = (0, 5)$; see Figure 1.2c.



(a) MPVC (1.6.2), with $\mathbf{x}^\circ$ included.



(b) MPVC (1.6.3), with $\mathbf{x}^\circ$ excluded.



(c) MPVC (1.6.4), with $\mathbf{x}^\circ$ excluded.

Figure 1.2: Results for the academic MPVC example. Initial points are marked if the solution found is close to $\mathbf{x}^\circ = (0, 0)$ (blue circle) or $\mathbf{x}^\star = (0, 5)$ (red star).

Table 1.2: Run time (median) and success rate of ALPX on the academic MPVC problem.

MPVC	Time [ms] (median)	$\mathbf{x}^\circ = (0,0)$		$\mathbf{x}^\star = (0,5)$	
(1.6.2)	0.6	707	(27.2%)	1894	(72.8%)
(1.6.3)	0.7	0	(0.0%)	2404	(92.4%)
(1.6.4)	0.7	0	(0.0%)	2601	(100.0%)

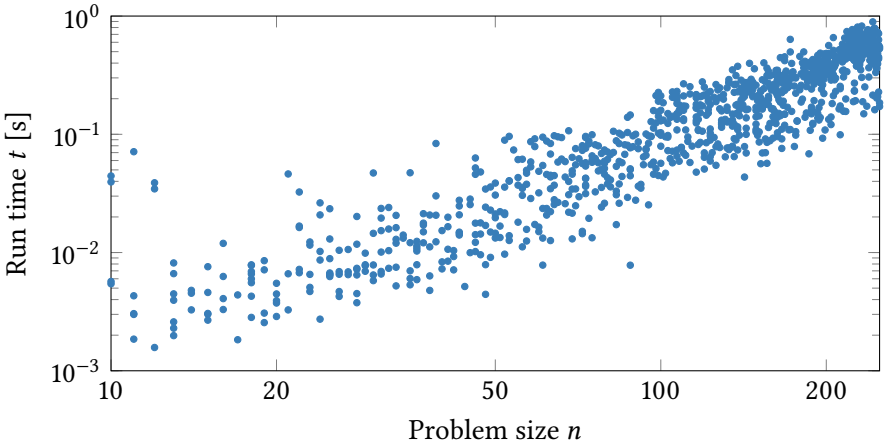


Figure 1.3: Results for the QPVC example. Problem size n versus run time t.

Quadratic programs with vanishing constraints MPVCs arise also in applications with combinatorial or logic constraints. The adoption of SQP-type methods for such problems leads to quadratic programs with vanishing constraints (QPVCs) [159]. These problems have a feasible set which is structurally combinatorial, hence nonconvex. Without loss of generality, we focus on the following QPVC:

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad & f(\mathbf{x}) := \frac{1}{2}\mathbf{x}^\top \boldsymbol{Q}\mathbf{x} + \mathbf{q}^\top \mathbf{x} \\ \text{subject to} \quad & (\boldsymbol{G}_i\mathbf{x} - \mathbf{g}_i)\mathbf{x}_i \geq 0, \quad \mathbf{x}_i \geq 0, \qquad i \in [1;N] \end{aligned} \tag{1.6.5}$$

where $\boldsymbol{Q} \in \mathrm{Sym}_{++}(\mathbb{R}^n)$, $\boldsymbol{G} \in \mathbb{R}^{N\times n}$, $\mathbf{q} \in \mathbb{R}^n$, $\mathbf{g} \in \mathbb{R}^N$, and $N \in \mathbb{N}$ denotes the number of vanishing constraints. This problem formulation can always be obtained by introducing slack variables and suitable rearrangements [159, §2.5]. We consider (1.1.1) with the following terms:

$$g(\mathbf{x}) := \chi_{\mathbb{R}^N_+}(\mathbf{x}_{1:N}) \qquad \mathbf{c}(\mathbf{x}) := \begin{pmatrix} \mathbf{x}_1 \\ [\boldsymbol{G}\mathbf{x} - \mathbf{g}]_1 \\ \vdots \\ \mathbf{x}_N \\ [\boldsymbol{G}\mathbf{x} - \mathbf{g}]_N \end{pmatrix} \qquad S := VC^N.$$

Herein the nonconvex set VC is adopted to model the vanishing constraint, and its projection operator Π_{VC} can be easily evaluated.

$$VC := \{(a,b) \in \mathbb{R}^2 : a = 0 \vee (a \geq 0 \wedge b \geq 0)\}$$

Whenever the projection $\Pi_{VC}((a,b))$ is set-valued, namely $a > 0$ and $a + b = 0$, we select the element $(a, 0)$. Analogous results were obtained choosing $(0, b)$ instead.

We consider 1000 problems with $n \in [10; 250]$, $N = \lceil n/5 \rceil$, and randomly generated problem data. We set $\boldsymbol{Q} = \boldsymbol{P}^\top \boldsymbol{P}$, where $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{P}_{ij} \sim \mathcal{N}(0,1)$, $\mathbf{q}_i \sim \mathcal{N}(0,1)$, $\boldsymbol{G}_{ij} \sim \mathcal{N}(0,1)$, and $\mathbf{g}_i \sim \mathcal{N}(0,1)$. We run ALPX, with the default settings, starting from the initial guess $\mathbf{x}_0 = \mathbf{0}$, with dual estimate $\mathbf{y}_0 = \mathbf{0}$. ALPX is able to solve all the problem instances, requiring a modest number of (outer) iterations, 12 on median value, ranging from 4 to 25. Computation times are reported in Figure 1.3.

Either-or constraints We now consider an optimization problem with either-or constraints [225, §5.2.1]. The problem reads

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^2}{\text{minimize}} \quad & (x_1 - 8)^2 + (x_2 + 3)^2 \\ \text{subject to} \quad & x_1 - 2x_2 \leq -4 \quad \vee \quad x_1 \leq 2 \\ & x_1^2 \leq 4x_2 \quad \vee \quad (x_1 - 3)^2 + (x_2 - 1)^2 \leq 10. \end{aligned} \tag{1.6.6}$$

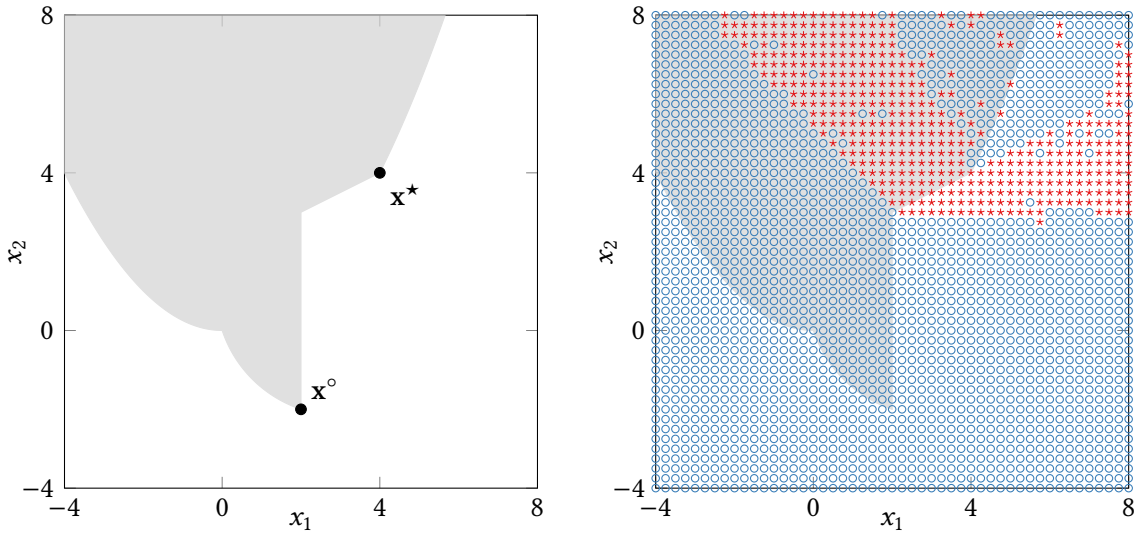


Figure 1.4: Feasible set (left) and results for problem (1.6.7). Initial points are marked if the solution found is close to $\mathbf{x}^\circ = (2, -2)$ (blue circle) or $\mathbf{x}^\star = (4, 4)$ (red star).

It admits a unique global minimizer $\mathbf{x}^\circ = (2, -2)$, a local minimizer $\mathbf{x}^\star = (4, 4)$, and its feasible set is depicted in Figure 1.4. We reformulate (1.6.6) as a NCSP by considering (1.1.1) with the following terms:

$$f(\mathbf{x}) := (x_1 - 8)^2 + (x_2 + 3)^2 \qquad g(\mathbf{x}) := 0 \qquad (1.6.7)$$
$$\mathbf{c}(\mathbf{x}) := \begin{pmatrix} x_1 - 2x_2 + 4 \\ x_1 - 2 \\ x_1^2 - 4x_2 \\ (x_1 - 3)^2 + (x_2 - 1)^2 - 10 \end{pmatrix} \qquad S := EO^2.$$

Herein the nonconvex set EO is adopted to model the either-or constraint:

$$EO := \{(a, b) \in \mathbb{R}^2 : a \leq 0 \vee b \leq 0\}.$$

Thus, in contrast with the approach followed in [225, §5.2.1], we need not introduce additional variables. Moreover, despite EO being nonconvex, computing an element of the projection Π_{EO} is trivial; ditto for Π_S. Based on the fact that

$$\Pi_{EO}((a, b)) = \begin{cases} (a, 0) & \text{if } a > b > 0, \\ \{(a, 0), (0, b)\} & \text{if } a = b > 0, \\ (0, b) & \text{if } b > a > 0, \\ (a, b) & \text{otherwise,} \end{cases} \qquad (1.6.8)$$

we solved the problem selecting always the element $(a, 0)$ whenever the case $a = b > 0$ applied.

We run ALPX, with the default settings, starting from a grid of 2401 initial points in $[-4, 8]^2$. Figure 1.4 depicts the results, with the same procedure followed in Figure 1.2. The median run time was 1.5 ms. ALPX returns a point within 10^{-3} from $\mathbf{x}^\circ$ and $\mathbf{x}^\star$ respectively 1954 (81.4%) and 447 times (18.6%). Thus, the solver always converges to a feasible minimizer; Figure 1.4 shows the basin of attraction of $\mathbf{x}^\circ$ and $\mathbf{x}^\star$ for the adopted formulation and settings. Analogous results were obtained by choosing the element $(0, b) \ni \Pi_{EO}((a, b))$ whenever $a = b > 0$. This led to $\mathbf{x}^\circ$ and $\mathbf{x}^\star$ respectively 1972 (82.1%) and 429 times (17.9%).

Nonconvex optimization Let us consider the example given in [257, §4.1], entailing the constrained optimization of the Rosenbrock function. It reads

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^5}{\text{minimize}} \quad & f(\mathbf{x}) := \sum_{i=1}^{4} p_2(x_{i+1} - x_i^2)^2 + (p_1 - x_i)^2 \\ \text{subject to} \quad & \|\mathbf{x}\| \le p_4 \\ & p_3 \sin x_1 = \cos(x_2 + x_3) \\ & x_3 + x_4 \le p_5 \end{aligned} \tag{1.6.9}$$

where $\mathbf{p} = (1, 50, 1.5, 0.73, 0.2)$ are given parameters. Problem (1.6.9) can be written in multiple ways in the form of (1.1.1). Following the second formulation given in [257], we choose the terms

$$g(\mathbf{x}) := \chi_{NC}(\mathbf{x}) \qquad NC := \{\mathbf{x} \in \mathbb{R}^5 : \|\mathbf{x}\| \le p_4\}$$
$$\mathbf{c}(\mathbf{x}) := \begin{pmatrix} p_3 \sin x_1 - \cos(x_2 + x_3) \\ x_3 + x_4 - p_5 \end{pmatrix} \qquad S := \{0\} \times (-\infty, 0].$$

Herein NC denotes the set of vectors whose norm is less than or equal to p_4, and it is adopted to enforce the norm constraint. The proximal operator of $g := \chi_{NC}$ coincides with the projection Π_{NC}, which is single-valued and computationally inexpensive.

In [257, §4.1], the Authors state that "the formulation based on the augmented Lagrangian method ran in 1.4 ms after 5 outer and 175 total inner iterations." Adopting the same settings as in [257], namely $\epsilon_{\text{opt}} = 10^{-5}$, $\epsilon_{\text{feas}} = 10^{-4}$, $\epsilon_0 = 10^{-4}$, $\mu_0 = 10^{-3}$, and $\alpha_- = 0.2$, `ALPX` takes approximately 3.1 ms, 8 outer and 95 inner iterations to solve the problem. With these looser tolerances but its own default settings, `ALPX` takes 4.0 ms, 10 outer and 97 inner iterations. Finally, `ALPX` takes 9.9 ms, 23 outer and 282 inner iterations to solve the problem with high accuracy, namely with $\epsilon_{\text{opt}} = \epsilon_{\text{feas}} = 10^{-8}$. Considering that `OpEn` generates code for a problem-specific solver, the performance shown by `ALPX` seems satisfactorily and, indeed, promising.

1.6.2 Nonlinear programming

Nonlinear programming is a fundamental topic in mathematical continuous optimization [112, 180]. Many methods and algorithms have been designed and analyzed during the last decades. Concurrently, developing efficient and robust numerical solvers has attracted growing interest and effort. Some state-of-the-art solvers are `IPOPT` [113], `WORHP` [157, 212], `LANCELOT` [49, 51], `SNOPT` [91, 103], `Knitro` [109], `MINOS` [33, 39], `LOQO` [93], `PENNON` [95], `ALGENCAN` [87, 115, 163], and `IPFILTER` [99], among others.

General nonlinear programs (NLPs) can be represented in the form

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{x}_{\text{l}} \le \mathbf{x} \le \mathbf{x}_{\text{u}} \\ & \mathbf{c}_{\text{l}} \le \mathbf{c}(\mathbf{x}) \le \mathbf{c}_{\text{u}}, \end{aligned} \tag{1.6.10}$$

where the constant vectors $\mathbf{x}_{\text{l}}, \mathbf{x}_{\text{u}}, \mathbf{c}_{\text{l}}, \mathbf{c}_{\text{u}}$ satisfy $\mathbf{x}_{\text{l}} \le \mathbf{x}_{\text{u}}$, $\mathbf{c}_{\text{l}} \le \mathbf{c}_{\text{u}}$, and the inequalities are understood componentwise. A simple reformulation of (1.6.10) to recover the form of (1.1.1) is obtained by setting

$$g := \chi_{[\mathbf{x}_{\text{l}}, \mathbf{x}_{\text{u}}]} \qquad \text{and} \qquad S := [\mathbf{c}_{\text{l}}, \mathbf{c}_{\text{u}}].$$

Since both $[\mathbf{x}_{\text{l}}, \mathbf{x}_{\text{u}}]$ and $[\mathbf{c}_{\text{l}}, \mathbf{c}_{\text{u}}]$ are convex sets, this gives smooth, bound-constrained subproblems in Algorithm 3, which boils down to those in [49, 163]. Therefore, one can, and likely should, use methods tailored for such problems, *e.g.*, `LBFGS-B` [59, 73], `TRON` [78], and `GENCAN` [87, 163], instead of methods for structured optimization, such as `ZeroFPR` and `PANOC`, that do not exploit the properties of smooth problems. Aware of this drawback, we are interested in testing the performance of `ALPX` and comparing it to a state-of-the-art NLP solver. However, testing and benchmarking optimization codes deserves significant effort and full-fledged solvers; this analysis does not dare to achieve this goal.

Metrics Let S, P, and $t_{s,p}$ denote the set of solvers, the set of problems, and the time required for solver $s \in S$ to return a solution for problem $p \in P$. The shifted geometric mean (sgm) $\widehat{t}_s$ of the run times for solver $s \in S$ on P is defined by

$$\widehat{t}_s := \exp\left(\frac{1}{|P|}\sum_{p\in P} \ln\left(t_{s,p} + t_{\text{shift}}\right)\right) - t_{\text{shift}}$$

with the shift $t_{\text{shift}} = 1$ s [259]. Here, when solver s fails to solve problem p, the term $t_{s,p}$ is set to the time limit. We also adopt the performance profiles [88] to compare the solver timings. These plot the function $f_s^{\text{r}} : \mathbb{R} \to [0,1]$, $s \in S$, defined by

$$f_s^{\text{r}}(\tau) := \frac{|\{p \in P : t_{s,p} \le \tau\, t_p^{\min}\}|}{|P|}, \quad t_p^{\min} := \min_{s\in S} t_{s,p}.$$

Considering $t_{s,p} = +\infty$ when solver s fails on problem p, $f_s^{\text{r}}(\tau)$ is the fraction of problems solved by solver s within τ times the best timing. Since performance profiles may be misleading when more than two solvers are compared [184], we will compare them pair-wise.

Furthermore, performance profiles do not provide the percentage of problems that can be solved (for some given tolerance ϵ) within a given time t. Thus, on the vein of data profiles [125, §2.2], we plot the function $f_s^{\text{a}} : \mathbb{R} \to [0,1]$, $s \in S$, defined by

$$f_s^{\text{a}}(t) := \frac{|\{p \in P : t_{s,p} \le t\}|}{|P|}.$$

Considering $t_{s,p} = +\infty$ when solver s fails on problem p, $f_s^{\text{a}}(t)$ is the fraction of problems solved by solver s within the time t. Note that, in contrast to f_s^{r}, the time profile $t \mapsto f_s^{\text{a}}(t)$ is independent from other solvers and displayed with the actual timings of solver s.

Setup We consider the subset of the CUTEst benchmark problems [175] with at least one variable, one nonlinear constraint, and at most 100 variables and 100 nonlinear constraints; this selection yields 446 problems. We also consider `IPOPT`, a well-established code implementing a primal-dual filter-linesearch interior-point method [107, 106, 113]; see https://coin-or.github.io/Ipopt/. A similar comparison of `IPOPT` against an augmented Lagrangian filter method can be found in [242].We access the problems and the solver through the infrastructure and the tools given by `CUTEst.jl` [246] and `NLPModelsIpopt.jl` [248], respectively. `OptiMo` offers a tool to import NLP models expressed with `NLPModels.jl` [247], and so to feed `ALPX` with CUTEst problems.

We run both `ALPX` and `IPOPT` with their default settings, besides the convergence tolerance $\epsilon = \epsilon_{\text{opt}} = \epsilon_{\text{feas}}$. A problem instance is considered solved by a solver if the output status is `first_order`, for `ALPX`, or `Solve_Succeeded`, for `IPOPT`; otherwise, it is a failure.

Results Computational results are reported in Table 1.3, with run times and failure rates, and depicted in Figure 1.5 with performance and time profiles, for different tolerance values. Although, at the moment of writing, the codes in the software package `Bazinga` are implemented for research purposes and are far from being production-ready, `ALPX` proves to be relatively robust. Many features which make `IPOPT` well-respected are not yet implemented, such as, *e.g.*, a robust restoration phase. For example, the optimizers in `Bazinga` stop as soon as a IEEE NaN or Inf is detected, without any tentative of recovery.

In terms of shifted geometric mean, `ALPX` is slower than `IPOPT` for all the tolerance values; see Table 1.3. However, inspecting the profiles in Figure 1.5, we can better appreciate the differences of the two methods. The time profiles demonstrate that `IPOPT`'s timings lies in a narrow interval and, in contrast, `ALPX` is very fast on some problems and takes long on others. An example of these problems is `MANCINONE` ($n = m = 100$), where `ALPX` stalls: with $\epsilon = 10^{-8}$, `IPOPT` succeeds in 4 iterations while `ALPX` finds a feasible point with dual residual $4.2 \cdot 10^{-8}$ in 6 outer and 97 inner iterations, but it is not able to progress further until the maximum number of iterations is reached. Indeed, `IPOPT` relies on

Table 1.3: Run time and failure rate on the CUTEst problems with $n, m \in [1; 100]$.

	Tolerance ϵ		10^{-4}	10^{-6}	10^{-8}
ALPX	Time (sgm)	[s]	3.010	3.532	4.242
	Failure rate	[%]	28.9	31.6	34.8
IPOPT	Time (sgm)	[s]	2.748	2.828	2.833
	Failure rate	[%]	28.5	28.9	28.9

a second-order Newton-type method, tailored to smooth NLPs, whereas ALPX does not exploit their structure. Nevertheless, on the considered test set, our ALPX seems competitive against IPOPT in terms of robustness and speed.

1.7 Summary

We presented the class of constrained structured optimization problems. By combining the augmented Lagrangian framework and proximal methods, we derived a numerical method and designed an algorithm for their approximate solution. Then, we investigated the convergence properties of the proposed algorithm and implemented it in the open-source Julia package Bazinga, available online. Numerical examples showed the benefits of formulating problems in the form of constrained structured programs. Finally, our implementation proved competitive with a state-of-the-art solver for small to medium-size nonlinear programming problems.

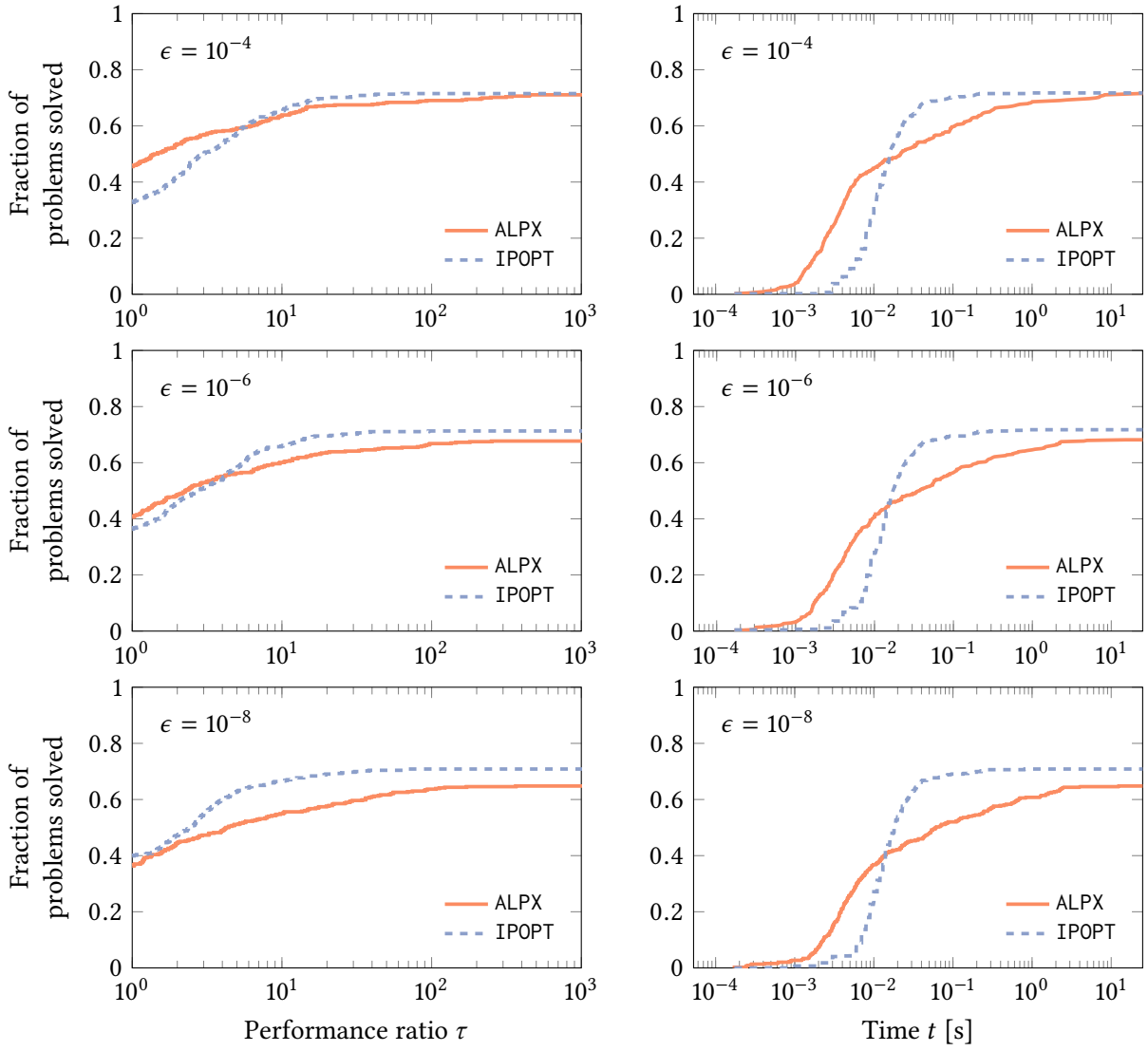


Figure 1.5: Results on the 446 CUTEst problems with $n, m \in [1; 100]$, for different tolerance values: performance profiles (left) and time profiles (right).

Chapter 2

Sparse Constrained Switching Time Optimization

All things being equal,
the simplest solution tends to be the best one.
—William of Ockham

This chapter considers mixed-integer optimal control problems with switching costs. A reformulation based on the cardinality function is proposed, which leads to constrained, structured problems.

The content of this chapter partially appears in [221, 237, 235].

2.1 Introduction

Mixed-integer optimal control problems offer many challenges, prominently posed by the discrete nature of the admissible control set [105, 134, 203]. The combinatorial structure of the problem makes approaches based on discrete optimization impractical [102]. This can be overcome by relaxing the integrality constraints, solving the relaxed problem with continuous optimization techniques, and then properly reconstruct the discrete-valued control. The combinatorial integral approximation (CIA) is a well-studied decomposition approach which takes advantage of the peculiar properties and structure of mixed-integer optimal control problems (MIOCPs) [144, 154, 241, 134]. Theoretical results provide error bounds to the exact solution [144, 241] and numerical evidence seems promising [126]. Other approaches stem from different reformulations, and comprise the control parametrization enhancing technique (CPET) [50, 67, 75, 77, 81], the variable time transformation (VTT) [111], and the switching time optimization (STO) [110, 178, 204].

Some MIOCPs exhibit solutions with chattering behavior, also known as Zeno's phenomenon, *i.e.*, the optimal solution may switch infinitely many times in a finite amount of time [58]. However, such solutions may be unrealistic and undesired in applications, and possibly just modeling artifacts. Moreover, one may wish to limit or penalize variations in the control inputs, even for problems with continuous-valued controls [160] or whose solution has finitely many switches. These observations lead to the formulation of MIOCPs with switching constraints and/or costs [134, 226]. Approaches for numerically solving such problems have been recently proposed, based on CIA and exploiting structure of the integer program for reconstructing the discrete-valued control [219]. These ideas introduce a trade-off between the approximation accuracy of the reconstruction and the costs induced by the optimized control function; see [230].

In this chapter we follow and discuss the approach proposed in [221] and extended in [237, 235]. Here we focus on problems with discrete-valued control inputs, so that the STO reformulation can be readily applied. In particular, we refer to the formulation adopted in [204], which proved effective but did not include switching costs. Strategies for dealing also with continuous-valued controls have been detailed in [221, 235] and can build upon, *e.g.*, the direct multiple shooting method [15, 41, 142]. Another

way would be considering indirect methods for OCPs [181], possibly by exploiting the minimum principle [16] and algorithms for solving generalized equations.

Some works which motivated and inspired this chapter are [111, 160, 156], whereas the approaches in [110, 162, 178, 174, 210, 230] differ in spirit but attempt to tackle similar problems from different perspectives. Although, at the moment of writing, it appears that both the time transformation in STO and the cardinality-based regularization yield unfortunate formulations, as recently underlined in [207, 251], the approach detailed in this chapter aims at providing and investigating a fresh viewpoint on switching costs.

Mixed-integer optimal control The problems considered in the remaining of this chapter belong to a broader class of constrained optimal control problems involving dynamic processes modeled by ordinary differential equations. These can be referred to as mixed-integer optimal control problems (MIOCPs) with control-volatility costs:

$$
\begin{aligned}
\underset{\mathbf{x}(\cdot),\mathbf{u}(\cdot),\mathbf{w}(\cdot)}{\text{minimize}} \quad & J(\mathbf{x}) + V(\mathbf{u},\mathbf{w}) && && (2.1.1)\\
\text{subject to} \quad & \dot{\mathbf{x}}(t) = \mathbf{f}(t,\mathbf{x}(t),\mathbf{u}(t),\mathbf{w}(t)), && t \in [t_\mathrm{i}, t_\mathrm{f}]\\
& \mathbf{0} = \mathbf{b}(\mathbf{x}(t_\mathrm{i}),\mathbf{x}(t_\mathrm{f})) &&\\
& \mathbf{0} \le \mathbf{r}(t,\mathbf{x}(t),\mathbf{u}(t),\mathbf{w}(t)), && t \in [t_\mathrm{i}, t_\mathrm{f}]\\
& \mathbf{u}(t) \in U, \mathbf{w}(t) \in W, && t \in [t_\mathrm{i}, t_\mathrm{f}].
\end{aligned}
$$

The state trajectory $\mathbf{x}(\cdot)$ of the dynamical system is affected by *continuous* controls $\mathbf{u}(\cdot)$ as well as *discrete* controls $\mathbf{w}(\cdot)$. The former attain values from the control set U, assumed to be a closed set with nonempty interior, whereas the latter attain values from a finite discrete set

$$
W := \{\mathbf{w}^1, \dots, \mathbf{w}^{n_W}\}, \quad \text{with cardinality } |W| = n_W < \infty.
$$

The functions J, $\mathbf{f}$, $\mathbf{b}$, $\mathbf{r}$ are supposed to be sufficiently smooth and, without loss of generality, it is assumed that the initial time t_i and the final time t_f are fixed. The objective function consists of two terms. The first term J depends on the state vector and represents the control objective; although common in practice, it is not our focus in this chapter. The second term V encodes volatiliy costs, that is, penalization terms for variations of the continuous controls [160, 244] and switchings of the discrete controls [219, 230]. For more details on numerical methods for MIOCPs, we refer to [102, 105, 111, 135], and [134, Ch. 2].

In this chapter we explore the switching time optimization (STO) approach for solving problems of the kind in (2.1.1). Without the term V of volatility costs, STO carries some drawbacks and challenges, such as the unknown number of switches, the nonregularity that occurs when intervals vanish, and the additional nonconvexities [135, 251]. Nevertheless, we believe STO offers some advantages to harness switching costs, namely to discourage frequent changes of the discrete controls. In fact, the STO approach takes the discrete controls $\mathbf{w}(\cdot)$ fixed on a grid and optimizes the switching times, that is, the grid points. Thus, it is based on a re-parametrization of the original problem, on the vein of the variable time transformation (VTT) [111] and the control parametrization enhancing technique (CPET) [50, 77]. All these approaches seek optimal switching times for a given discrete control grid in place of optimal discrete controls for a given time grid. Instead of seeking the optimal control $\mathbf{w}(\cdot) : [t_\mathrm{i}, t_\mathrm{f}] \to W$, we fix a number of intervals $N \in \mathbb{N}$, choose a discrete control sequence $\{\mathbf{w}_1, \dots, \mathbf{w}_N\} \subseteq W$ with N elements and define the N control functions

$$
\begin{gathered}
\tilde{\mathbf{w}}_k : [\tau_k, \tau_{k+1}] \to \mathbf{w}_k, \ k \in [1; N],\\
\text{with } t_\mathrm{i} = \tau_0 \le \tau_1 \le \cdots \le \tau_N \le \tau_{N+1} = t_\mathrm{f}.
\end{gathered}
$$

As we take interest in penalizing changes of the discrete controls, we assume that an optimal control function $\mathbf{w}(\cdot)$ switches only finitely many times. Hence, the original problem is equivalent to optimizing the number of intervals N and the vector of switching times $\boldsymbol{\tau}$, along with $\mathbf{x}(\cdot)$ and $\mathbf{u}(\cdot)$. STO arises

from fixing the number of intervals N and optimizing the switching times $\boldsymbol{\tau}$. We further re-parametrize the problem in terms of the set $\Delta \subseteq \mathbb{R}^N_+$ and the vector of switching intervals $\mathbf{d} \in \Delta$, given by

$$\Delta := \left\{ \mathbf{d} \in \mathbb{R}^N_+ : \mathbf{1}^\top \mathbf{d} = t_\mathrm{f} - t_\mathrm{i} \right\}, \qquad d_k := \tau_{k+1} - \tau_k, \qquad k \in [1; N].$$

2.2 Problem Formulation

Let us consider problem (2.1.1) without continuous controls, with autonomous dynamics and constraints, and with fixed given initial time $t_\mathrm{i} = 0$ and final time $t_\mathrm{f} = T > 0$. Following the STO approach, we consider N intervals, a fixed discrete control sequence $\{\mathbf{w}_1, \dots, \mathbf{w}_N\} \subseteq W$, a vector of switching times $\boldsymbol{\tau} \in \mathbb{R}^{N+1}$, with $\tau_1 = t_\mathrm{i}$ and $\tau_{N+1} = t_\mathrm{f}$, and the sequences of dynamics $\{\mathbf{f}_k\}$ and constraints $\{\mathbf{r}_k\}$ defined by

$$\mathbf{f}_k(\mathbf{x}) := \mathbf{f}(\mathbf{x}, \mathbf{w}_k), \qquad \mathbf{r}_k(\mathbf{x}) := \mathbf{r}(\mathbf{x}, \mathbf{w}_k), \qquad k \in [1; N].$$

In terms of switching intervals $\mathbf{d}$, the switching cost can be expressed as $s(\mathbf{d})$, with $s : \Delta \to \mathbb{R}$, and the problem is reformulated as

$$\begin{aligned}
\underset{\mathbf{x}(\cdot), \mathbf{d}}{\text{minimize}} \quad & J(\mathbf{x}) + s(\mathbf{d}) \\
\text{subject to} \quad & \dot{\mathbf{x}}(t) = \mathbf{f}_k(\mathbf{x}(t)), && t \in [\tau_k, \tau_{k+1}),\ k \in [1; N] \\
& \mathbf{0} = \mathbf{b}(\mathbf{x}(0), \mathbf{x}(T)) \\
& \mathbf{0} \leq \mathbf{r}_k(\mathbf{x}(t)), && t \in [\tau_k, \tau_{k+1}),\ k \in [1; N] \\
& \mathbf{d} \in \Delta.
\end{aligned}$$

The set Δ is the simplex of radius T and introduces an additional (linear) constraint which couples the switching intervals $\mathbf{d}$. For free final time problems, only the nonnegativity constraint would remain [221]. For the sake of focusing on the switching costs $s(\cdot)$, we consider further simplifications: an initial state $\mathbf{x}_0 \in \mathbb{R}^{n_x}$ is given, terminal conditions are neglected, and the state cost J is linear-quadratic; see (2.2.1). Moreover, we express the constraints as a function only of the switching intervals $\mathbf{d}$, in the form $\mathbf{c}(\mathbf{d}) \in S$, with given constraint function $\mathbf{c} : \Delta \to \mathbb{R}^{n_c}$ and constraint set $S \subseteq \mathbb{R}^{n_c}$. Then, the problem of interest reads

$$\begin{aligned}
\underset{\mathbf{x}(\cdot), \mathbf{d}}{\text{minimize}} \quad & \mathbf{x}(T)^\top \boldsymbol{E} \mathbf{x}(T) + \int_0^T \mathbf{x}(t)^\top \boldsymbol{Q} \mathbf{x}(t) \mathrm{d}t + s(\mathbf{d}) \\
\text{subject to} \quad & \dot{\mathbf{x}}(t) = \mathbf{f}_k(\mathbf{x}(t)), \qquad t \in [\tau_k, \tau_{k+1}), \qquad k \in [1; N] \\
& \mathbf{x}(0) = \mathbf{x}_0 \\
& \mathbf{c}(\mathbf{d}) \in S \\
& \mathbf{d} \in \Delta.
\end{aligned} \tag{2.2.1}$$

Herein, state cost matrices $\boldsymbol{Q} \in \mathrm{Sym}_+(\mathbb{R}^{n_x})$ and $\boldsymbol{E} \in \mathrm{Sym}_+(\mathbb{R}^{n_x})$ are given, functions $\mathbf{f}_k$, $k \in [1; N]$, and $\mathbf{c}$ are assumed sufficiently smooth and the feasible set nonempty, without further mention. Employing the direct single shooting approach [85, 141], the state evolution becomes implicit, and only the switching intervals $\mathbf{d}$ are left to optimization. Hence, following [204], problem (2.2.1) can be rewritten as

$$\begin{aligned}
\underset{\mathbf{d} \in \mathbb{R}^N_+}{\text{minimize}} \quad & \mathbf{x}_0^\top \boldsymbol{J}(\mathbf{d}) \mathbf{x}_0 + s(\mathbf{d}) \\
\text{subject to} \quad & \mathbf{c}(\mathbf{d}) \in S, \quad \mathbf{d} \in \Delta,
\end{aligned} \tag{2.2.2}$$

where $\boldsymbol{J} : \mathbb{R}^N_+ \to \mathrm{Sym}_+(\mathbb{R}^{n_x})$ is a matrix-valued function. As detailed in [188, 204], function $\boldsymbol{J}$ corresponds to the linearization of the dynamics around $\mathbf{d}$, the evaluation of the state transformation matrix, and the integration of the associated cost.

General MIOCPs can be reformulated as (2.2.2), and then as NCSPs in the form of (1.1.1), possibly introducing slack variables, augmenting the state dimension and system dynamics, or applying other standard transformations for OCPs [141, 235].

Note. Problem (2.2.2) is obtained with many assumptions and after several simplifications. These have been applied here to focus on the switching costs and their interplay with the STO approach. Nonetheless, the ideas detailed in this chapter can fit the broader class of MIOCPs: general nonlinear smooth objective terms, continuous controls, state and control constraints, boundary conditions, and non-autonomous dynamics can be included. The integration of all these elements could be accomplished via *first discretize, then optimize* approaches, such as the multiple shooting method [41].

In this context, we are particularly interested in the switching cost term s, which is introduced to discourage changes in the discrete control; see [134, 230]. We propose to define it as

$$s(\mathbf{d}) := \sigma \operatorname{nnz}(\mathbf{d}) \tag{2.2.3}$$

for some given scalar $\sigma \in \mathbb{R}_+$, where the cardinality-like function nnz counts the nonzero elements of a vector, namely

$$\operatorname{nnz}(\mathbf{d}) := |\{i \mid d_i \neq 0, i \in [1; N]\}|. \tag{2.2.4}$$

In fact, it is a (uniformly) weighted cardinality function. In the literature it is referred to as ℓ^0-norm, with slight abuse of terminology [186, 258, 252]. However, the switching cost term s, as defined in (2.2.3), is not continuous on $\mathbb{R}_+^N$, because nnz is discontinuous. Therefore, the method presented in Chapter 1, namely ALPX detailed in Algorithm 3, is not guaranteed to work since the underlying assumptions fail to hold. In particular, the cost term g fails to be continuous on its domain, which leaves us with no guarantees on the convergence of the (shifted) penalty method. This issue is discussed and surmounted in the next section. Although restricted to this context, the results established in §2.3 show that it is possible to relax the assumptions behind the method developed in Chapter 1 for tackling NCSPs.

2.3 Cardinality, Simplex and Proximal Operator

The nonsmooth function s in (2.2.3) is accessed by ALPX, *i.e.* Algorithm 3, only through its proximal mapping, as an oracle, when solving (2.2.2). In the following §2.3.1, we construct a function which is a continuous relaxation of the cardinality function nnz in (2.2.4) but whose proximal mapping is the same. Then, thanks to this indistinguishable oracle, we can argue ALPX retains its convergence guarantees for solving the STO problem (2.2.2). Subsequently, we consider in §2.3.2 the proximal mapping of s subject to a simplex constraint. This allows to satisfy the fixed final time constraint exactly, by design. Furthermore, for fixed final time problems without additional constraints, this allows to reformulate (2.2.2) as a structured optimization problem and adopt suitable solvers, avoiding the augmented Lagrangian outer loop at once.

Let us start with the proximal mapping of nnz, without any constraint. Thanks to the separable structure, it is sufficient to consider the scalar case. Despite its simplicity, it does not admit a unique solution in general, due to the intrinsic nonconvex nature. Let $\gamma > 0$ be arbitrary in the following.

$$\operatorname{prox}_{\gamma \operatorname{nnz}}(x) := \operatorname*{arg\,min}_{z \in \mathbb{R}} \left\{ \operatorname{nnz}(z) + \frac{1}{2\gamma}(z - x)^2 \right\}$$

For $z = 0$, the term in braces attains the value $x^2/(2\gamma)$, whereas, for $z \neq 0$, the first term is 1 and the second is minimum for $z = x$ at zero, yielding the value 1. The well-known result is recovered by comparing these two cases, namely

$$\operatorname{prox}_{\gamma \operatorname{nnz}}(x) = \begin{cases} 0 & \text{if } |x| < \sqrt{2\gamma} \\ \{0, x\} & \text{if } |x| = \sqrt{2\gamma} \\ x & \text{if } |x| > \sqrt{2\gamma}. \end{cases} \tag{2.3.1}$$

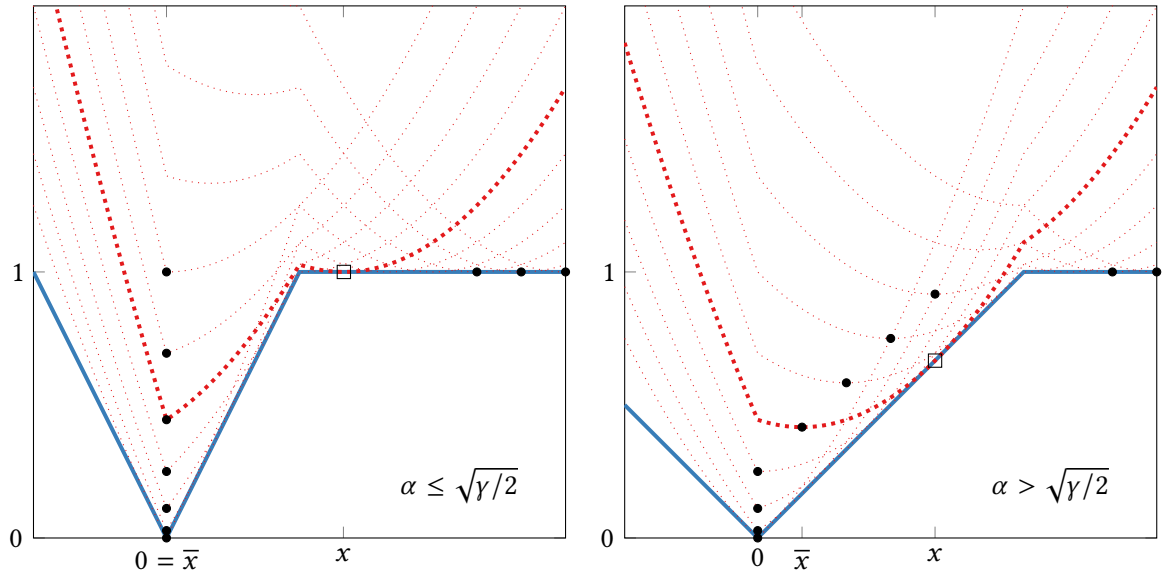


Figure 2.1: A proximal point $\overline{x}$ (black circles) of $\widetilde{\mathrm{nnz}}_\alpha$ (blue, solid line) at x is obtained by considering the minimizers of the corresponding proximal subproblem (red, dotted lines).

The nonnegativity constraint is useful in this context since switching intervals cannot attain negative values. Hence, we define the nonnegativity-constrained proximal mapping of nnz as

$$\mathrm{prox}_{\gamma\,\mathrm{nnz},\mathbb{R}_+}(x) := \operatorname*{arg\,min}_{z\in\mathbb{R}_+} \left\{ \mathrm{nnz}(z) + \frac{1}{2\gamma}(z-x)^2 \right\},$$

which can be evaluated following similar arguments:

$$\mathrm{prox}_{\gamma\,\mathrm{nnz},\mathbb{R}_+}(x) = \begin{cases} 0 & \text{if } x < \sqrt{2\gamma} \\ \{0, x\} & \text{if } x = \sqrt{2\gamma} \\ x & \text{if } x > \sqrt{2\gamma}. \end{cases} \tag{2.3.2}$$

2.3.1 Relaxed cardinality and proximal operator

In Chapter 1 it is assumed the possibly nonsmooth function g in (1.1.1) is continuous on its domain. Without this assumption, ALPX, which builds upon the (shifted) penalty method, is not guaranteed to work. The cardinality formulation given in (2.2.3) does not satisfy this assumption. Nonetheless, in this section we show that, for solving (2.2.2), ALPX is still a valuable tool. The core idea is that, since it accesses only the proximal mapping of the nonsmooth function and possibly evaluates it at the proximal point, it is sufficient to find a continuous function which acts as an equivalent, indistinguishable oracle. This is indeed the goal of this section.

Let us introduce the function $\widetilde{\mathrm{nnz}}_\alpha : \mathbb{R} \to \mathbb{R}$, defined by

$$\widetilde{\mathrm{nnz}}_\alpha(x) := \min\{1, |x|/\alpha\} \tag{2.3.3}$$

for some given $\alpha > 0$, which is a continuous function and gives a (symmetric, nonconvex, nonsmooth) approximation of nnz. The following result provides explicit expressions for the proximal mapping of $\widetilde{\mathrm{nnz}}_\alpha$, depending on the parameter α; see also Figure 2.1.

Proposition 2.3.1. *Let $\alpha, \gamma > 0$ and $x \in \mathbb{R}$ be arbitrary. If $\alpha \le \sqrt{\gamma/2}$, then it holds*

$$\mathrm{prox}_{\gamma\,\widetilde{\mathrm{nnz}}_\alpha}(x) = \begin{cases} 0 & \text{if } |x| < \sqrt{2\gamma} \\ \{0, x\} & \text{if } |x| = \sqrt{2\gamma} \\ x & \text{if } |x| > \sqrt{2\gamma}. \end{cases} \tag{2.3.4}$$

If $\alpha > \sqrt{\gamma/2}$, then it holds

$$\operatorname{prox}_{\gamma\,\widetilde{\operatorname{nnz}}_\alpha}(x) = \begin{cases} 0 & \text{if } |x| \le \gamma/\alpha \\ x - \operatorname{sign}(x)\gamma/\alpha & \text{if } |x| \in (\gamma/\alpha, \alpha + \gamma/(2\alpha)) \\ \{x - \operatorname{sign}(x)\gamma/\alpha, x\} & \text{if } |x| = \alpha + \gamma/(2\alpha) \\ x & \text{if } |x| > \alpha + \gamma/(2\alpha). \end{cases} \tag{2.3.5}$$

Proof. Let us denote the value $c_{\gamma\,\widetilde{\operatorname{nnz}}_\alpha}(z;x) := \min\{1, |z|/\alpha\} + (z-x)^2/(2\gamma)$ and the proximal point $\overline{x} \in \operatorname{prox}_{\gamma\,\widetilde{\operatorname{nnz}}_\alpha}(x)$. We can consider the two cases $\overline{x} \in [0, \alpha]$ and $\overline{x} \ge \alpha$, and then rely on symmetry. In the latter case, it must be

$$\overline{x} = \operatorname*{arg\,min}_{z\in\mathbb{R}} \left\{ 1 + \frac{1}{2\gamma}(z-x)^2 \right\} = x \ge \alpha.$$

In the former case, instead, it is

$$\overline{x} = \operatorname*{arg\,min}_{z\in\mathbb{R}} \left\{ \frac{z}{\alpha} + \frac{1}{2\gamma}(z-x)^2 \right\} = x - \frac{\gamma}{\alpha} \in [0, \alpha],$$

and therefore it is valid only for $x \in [\gamma/\alpha, \alpha+\gamma/\alpha]$. For $x \in [0, \gamma/\alpha]$, by symmetry and continuity of the value function, it is $\overline{x} = 0$. Hence, there are at most three, possibly overlapping, intervals, characterized by (i) $\overline{x} = 0$ for $x \in [0, \gamma/\alpha]$, (ii) $\overline{x} = x - \gamma/\alpha$ for $x \in [\gamma/\alpha, \alpha+\gamma/\alpha]$, and (iii) $\overline{x} = x$ for $\mathbf{x} \ge \alpha$. It remains to find the points where the corresponding values of $c_{\gamma\,\widetilde{\operatorname{nnz}}_\alpha}$ coincide. In particular, it is relevant where (iii) is positioned with respect to (i) and (ii). Algebraic manipulations give the breakeven point for (i) and (iii) at $x = \sqrt{2\gamma}$, valid if $\alpha \le \sqrt{\gamma/2}$, and for (ii) and (iii) at $x = \alpha + \gamma/(2\alpha)$, valid if $\alpha \ge \sqrt{\gamma/2}$. Therefore, if $\alpha \le \sqrt{\gamma/2}$, there exist only regions (i) and (iii), proving (2.3.4). Otherwise, the three regions coexist, as in (2.3.5). □

Remarkably, Proposition 2.3.1 establishes that selecting $\alpha > 0$ sufficiently small, in fact $\alpha \in (0, \sqrt{\gamma/2}]$, yields a proximal mapping for $\widetilde{\operatorname{nnz}}_\alpha$ with the same structure as for nnz; see (2.3.1) and (2.3.4). Indeed, the same value is attained at the proximal point:

$$\widetilde{\operatorname{nnz}}_\alpha(\overline{x}) = \min\{1, |\overline{x}|/\alpha\} = \operatorname{nnz}(\overline{x}).$$

Therefore, for sufficiently small $\alpha > 0$, the cardinality function nnz and the continuous relaxation $\widetilde{\operatorname{nnz}}_\alpha$ have indistinguishable oracles. We deduce it is possible to use the cardinality formulation while retaining the convergence properties and guarantees of ALPX, since it cannot *de facto* distinguish the two oracles.

2.3.2 Simplex-constrained proximal operator

This section is devoted to the simplex-constrained proximal mapping of nnz, which reads

$$\operatorname{prox}_{\gamma\,\operatorname{nnz},\Delta}(\mathbf{x}) := \operatorname*{arg\,min}_{\mathbf{z}\in\Delta} \left\{ \operatorname{nnz}(\mathbf{z}) + \frac{1}{2\gamma}\|\mathbf{z}-\mathbf{x}\|^2 \right\} \tag{2.3.6}$$

with $\gamma > 0$ and $\Delta := \{\mathbf{d} \in \mathbb{R}^n_+ \mid \mathbf{1}^\top \mathbf{d} = \beta\}$, $\beta \ge 0$. Some blanket assumptions are considered in the following.

Assumption 2.3.2. *(i)* $\beta > 0$. *(ii)* $\mathbf{x} \in \mathbb{R}^n$ *is sorted,* i.e., $x_1 \le x_2 \le \cdots \le x_n$.

In fact, these conditions give no loss of generality. (i) For $\beta = 0$, the feasible set collapses to the origin, *i.e.*, it is $\Delta = \{\mathbf{0}\}$, and thus the solution is trivially the origin. (ii) The cardinality function, and thus its proximal operator, are invariant under permutation of the input vector (and reverse-permutation of the output vector). Nonetheless, from a computational point-of-view, sorting may be inefficient for large values of n, which may ask for different approaches.

We notice that, for both the unconstrained and the nonnegative proximal mapping, the computation can be performed entrywise in that the underlying optimization problems are separable. Conversely, the simplex constraint couples the decision variable all together via a linear equality constraint. This leads us to follow a different approach, in particular the one proposed in [221]. Let us reformulate the simplex-constrained problem (2.3.6) in terms of the number of zero elements, denoted m, in the output vector $\overline{\mathbf{x}}$, namely $m := n - \mathrm{nnz}(\overline{\mathbf{x}})$. Introducing the integer variable m, problem (2.3.6) can be viewed as a constrained mixed-integer quadratic program, and can be solved as such. However, here we seek a tailored method exploiting its structure. In our approach, problem (2.3.6) is interpreted and tackled as a bilevel problem, aiming at optimizing the number of zero elements m, at the upper level, while accounting for the positive entries of $\overline{\mathbf{x}}$ at the lower level. Considering the problem size n to be relatively small in practice, say $n < 100$, we look for an efficient routine for solving the lower level problem and evaluating the associated upper level cost, so that a minimizer is then found by parsing the entire space of (feasible) values for m.

For any positive scalar β, the integer scalar m takes values in $\mathcal{M} := [0; n-1]$. Let us define the (nonconvex) set $\Delta[m]$ consisting of vectors in the simplex Δ with m zero entries, namely $\Delta[m] := \{\mathbf{d} \in \Delta \mid \mathrm{nnz}(\mathbf{d}) = n - m\}$. Then, the proximal problem (2.3.6) is equivalent to the mixed-integer program

$$(\overline{m},\ \overline{\mathbf{x}}[\overline{m}]) = \operatorname*{arg\,min}_{\substack{m\in\mathcal{M},\\ \mathbf{z}\in\Delta[m]}} \left\{ n - m + \frac{1}{2\gamma}\|\mathbf{z}-\mathbf{x}\|^2 \right\} \tag{2.3.7}$$

where $\overline{\mathbf{x}}[m]$ is a $\Delta[m]$-constrained proximal point. For any given $m \in \mathcal{M}$, this can be found by solving the lower level problem

$$\overline{\mathbf{x}}[m] = \operatorname*{arg\,min}_{\mathbf{z}\in\Delta[m]} \|\mathbf{z}-\mathbf{x}\|^2. \tag{2.3.8}$$

The following result provides a structural characterization of $\overline{\mathbf{x}}[m]$.

Proposition 2.3.3. *Let Assumption 2.3.2 hold. Then, there exists a vector $\overline{\mathbf{x}}[m]$ that solves problem (2.3.8) and has m initial zero entries followed by $(n-m)$ positive entries. If Assumption 2.3.2(ii) holds with strict inequalities, namely $x_1 < x_2 < \cdots < x_n$, then the solution vector $\overline{\mathbf{x}}[m]$ is unique.*

Proof. Existence follows from compactness of $\Delta[m]$ and continuity of the objective function. The solution structure follows from Assumption 2.3.2 and the simplex constraint. □

Let us denote L the Lagrange function for the lower level problem and λ the multiplier associated to the equality constraint induced by the simplex, namely $\mathbf{1}^\top\mathbf{z} = \beta$. Omitting nonnegativity and cardinality constraints, the Lagrange function L reads

$$L(\mathbf{z},\lambda) := \frac{1}{2}\|\mathbf{z}-\mathbf{x}\|^2 + \lambda(\beta - \mathbf{1}^\top\mathbf{z})\ . \tag{2.3.9}$$

The first-order necessary optimality condition $\nabla_z L(\mathbf{z},\lambda) = \mathbf{0}$, together with direct derivation of L in (2.3.9) and Proposition 2.3.3, yields

$$\overline{x}_i[m] = \begin{cases} 0 & \text{if } i \in [1;m] \\ x_i + \lambda[m] & \text{if } i \in [m+1;n]. \end{cases} \tag{2.3.10}$$

Substituting this into the equality constraint, one obtains an expression for the multiplier λ, after some rearrangements, as dependent solely on m and the problem data.

$$\lambda[m] = \frac{1}{n-m}\left(\beta - \sum_{i=m+1}^{n} x_i\right) \tag{2.3.11}$$

The nonnegativity constraints are not enforced so far. However, given a vector $\mathbf{x}$, for any $m \in \mathcal{M}$, the associated multiplier $\lambda[m]$ is fixed by (2.3.11) and thus one can check whether the vector $\mathbf{x}[m]$ from (2.3.10) has negative entries or not. Given m, it suffices to check $x_{m+1} + \lambda[m] > 0$, since $\mathbf{x}$ is sorted, by Assumption 2.3.2. If this condition is not met, the tentative value of m is actually invalid and should be discarded. Let us denote $\mathcal{F} \subseteq \mathcal{M}$ the set of feasible values for m.

Remark 2.3.4. There exists always at least a feasible value for m, *i.e.*, $\mathcal{F} \neq \emptyset$. Indeed, since $\lambda[n-1] = \beta - x_n$ and hence $\overline{x}_n[n-1] = x_n + \lambda[n-1] = \beta > 0$, it is always $n-1 \in \mathcal{F}$.

For any $m \in \mathcal{F}$, the lower level solution $\overline{\mathbf{x}}[m]$ reads as in (2.3.10), where the multiplier $\lambda[m]$ is given in (2.3.11). The associated cost $c[m]$ is given by

$$\begin{aligned} c[m] &= \mathrm{nnz}(\overline{\mathbf{x}}[m]) + \frac{1}{2\gamma}\|\overline{\mathbf{x}}[m] - \mathbf{x}\|^2 \\ &= (n-m) + \frac{1}{2\gamma}\sum_{i=1}^{m} x_i^2 + \frac{1}{2\gamma}(n-m)\lambda^2[m]. \end{aligned} \tag{2.3.12}$$

With this expression at hand, one can search the minimum of $c[m]$ for $m \in \mathcal{F}$, retrieve the minimizer $\overline{m}$, and then set $\overline{\mathbf{x}} = \overline{\mathbf{x}}[\overline{m}]$. Notice that the numerical evaluation of $\lambda[m]$ and $c[m]$ can be performed more efficiently by storing the cumulative sums in (2.3.11) and (2.3.12). The following observation can be exploited to prune infeasible values of $m \in \mathcal{M}$.

Lemma 2.3.5. *Let Assumption 2.3.2 hold. Then, the feasible set can be expressed as* $\mathcal{F} = \{m \mid m \geq m_\ell, m \in \mathcal{M}\}$, *where the value of* $m_\ell \in \mathcal{M}$ *depends on* $\mathbf{x}$ *and* β, *and corresponds to the lowest value* $m \in \mathcal{M}$ *such that* $x_{m+1} + \lambda(m) > 0$.

Proof. Let us denote $\zeta(m) := x_{m+1} + \lambda[m]$. Hence, by construction, it is $m \in \mathcal{F}$ if and only if $m \in \mathcal{M}$ and $\zeta(m) > 0$. Consider any $p \in \mathcal{F}$ and $q \in \mathcal{M}$ such that $p < q$. We seek a proof that $q \in \mathcal{F}$, that is, $\zeta(q) > 0$. Based on (2.3.11) and the definition of ζ, we have that

$$\begin{aligned} (n-q)\zeta(q) &= (n-q)x_{q+1} + \beta - \sum_{i=q+1}^{n} x_i \\ &= (n-q)x_{q+1} + (n-p)[\zeta(p) - x_{p+1}] + \sum_{i=p+1}^{q} x_i, \end{aligned}$$

after expanding and rearranging the terms to collect $\zeta(p)$. A lower bound for the last term is obtained by observing that $\sum_{i=p+1}^{q} x_i \geq (q-p)\, x_{p+1}$ by Assumption 2.3.2. Thus, by $\zeta(p) > 0$, it follows that

$$\begin{aligned} \zeta(q) &\geq x_{q+1} + \frac{n-p}{n-q}[\zeta(p) - x_{p+1}] + \frac{q-p}{n-q}x_{p+1} = x_{q+1} + \frac{n-p}{n-q}\zeta(p) - x_{p+1} \\ &\geq \frac{n-p}{n-q}\zeta(p) > \zeta(p) > 0, \end{aligned}$$

concluding the proof. □

Finally, we point out that, as Lemma 2.3.5 suggests, one can obtain a coarse lower bound for the number of zeros by inspecting the projection of $\mathbf{x}$ onto the simplex, namely $\overline{m} \geq n - \mathrm{nnz}(\Pi_\Delta(\mathbf{x}))$. In fact, the sparsity-inducing regularization can only reduce the number of nonzeros in $\overline{\mathbf{x}}$ with respect to $\Pi_\Delta(\mathbf{x})$.

2.4 Numerical Results

This section presents numerical results obtained with the proposed method. We implemented a modelling tool for STO problems with switching costs in the `Julia` package `ScSTO`, see §0.1. This builds upon `SwitchTimeOpt`, a module provided by [204]. `ScSTO` provides high-level language and routines, which allow the user to easily define problem instances and seamlessly interface with suitable solvers. Currently, `ScSTO` supports the solvers for constrained and unconstrained structured optimization provided by `Bazinga`, through `OptiMo` (cf. §1.6). All examples described in the following are available as accompanying demos of `ScSTO`.

We investigate potential and limitations of the proposed approach on three problems with switching costs: a fishing problem with switched Lotka–Volterra dynamics, with and without constraints, and a machine maintenance planning problem. All the examples are solved with the default solvers' options and initialized with equally spaced switching times between the initial and final times. The interested reader may find additional examples and benchmark problems in [153] and [221, 235, 237].

2.4.1 Fishing problems

Let us consider the optimal switching control problem of Lotka–Volterra dynamics [105, 204], with tracking and switching costs, fixed final time, and no further constraints. Besides the switching costs, this problem is a classical example in mixed-integer optimal control [153]. The dynamics can be described by

$$\mathbf{f}(\mathbf{x}, u) = \begin{pmatrix} x_1 - x_1 x_2 - 0.4 x_1 w \\ x_1 x_2 - x_2 - 0.2 x_2 w \end{pmatrix}, \qquad w \in \{w_-, w_+\},$$

and the two control inputs $w_- = 0$ and $w_+ = 1$. We consider the final time $T = 12$, the initial state $\mathbf{x}_0 = (0.5, 0.7)$, $N = 11$ switching intervals, with control input sequence $\{w_-, w_+, w_-, w_+, \dots\}$. The tracking-type cost term $\int_0^T \|\mathbf{x}(t) - \mathbf{1}\|^2 \mathrm{d}t$ is accounted for by augmenting the system state with two constant states; see [141, 204]. For gradient computations via sensitivity analysis, we adopt a fixed discretization grid with $n = 100$ time points. Without constraints $\mathbf{c}$ in (2.2.2), by exploiting the simplex-constrained proximal mapping from §2.3.2, we obtain an unconstrained structured problem formulation. This is solved via `ZeroFPR` (50 max iterations).

Figure 2.2 displays the results obtained with switching cost $\sigma \in \{0, 0.1, 1\}$. As changes in the dynamics are more penalized, fewer switching intervals are assigned with a positive duration. Despite the conflicting objectives of minimizing tracking error and switching cost, the overall state evolution seems to be only slightly affected. Without any switching cost, the control activates three times, each time for a shorter time period; an analogous solution is found with a small switching cost, whereas for a high value of the switching cost just one activation is left. This suggests that the activation at around $t \approx 2.5$ until $t \approx 4.2$ is (in some sense) a robust choice. Also, we can see the filtering action of switching costs. For $\sigma = 0.1$, the third activation disappears, meaning that its contribution to the smooth objective term is smaller than the switching cost. Therefore, unrealistic control inputs corresponding to chattering solutions can be easily avoided by introducing a positive σ.

It is worth mentioning that the solver, `ZeroFPR` in this case, returns an iterate which does not satisfy the optimality tolerance after the maximum number of iterations. This is due to the discretization and linearization procedure proposed in [204], and currently employed in `ScSTO`, in the spirit of direct single shooting methods. In fact, numerical tests have shown that, with finer discretization grids, say $n = 1000$, the solver succeeds. However, these have also highlighted that the solver, with the chosen settings ($n = 100$ and max 50 iterations), returns iterates very close to a solution.

Let us now introduce some constraints on the switching times. In particular, we seek

$$1 \le \tau_1 \le 2, \qquad 4 \le \tau_3 \le 5, \qquad \text{and} \qquad 7 \le \tau_5 \le 8.$$

These conditions are automatically reformulated by `ScSTO` in the form of (2.2.2). Inspecting the control w in Figure 2.2, we expect the first activation at $\tau_1 = 2.5$ to move backward to $t = 2$ and the second at $\tau_3 = 5.1$ to $t = 5$. We point out that, in contrast with the previous example, we now have a constrained structured problem, which can be solved by running `ALPX`. We consider maximum 10 outer iterations, subsolver `ZeroFPR` with maximum 10 iterations, and start from the unconstrained solution (and null dual variable).

Figure 2.3 depicts the results for $\sigma = 0.1$, comparing the constrained and unconstrained solutions. The constrained solution gives $\tau_1 = 2$, as expected, but, interestingly, the second activation takes place at $\tau_3 = 4$. This counter-intuitive behaviour likely stems from the nonconvexities present in the problem formulation.

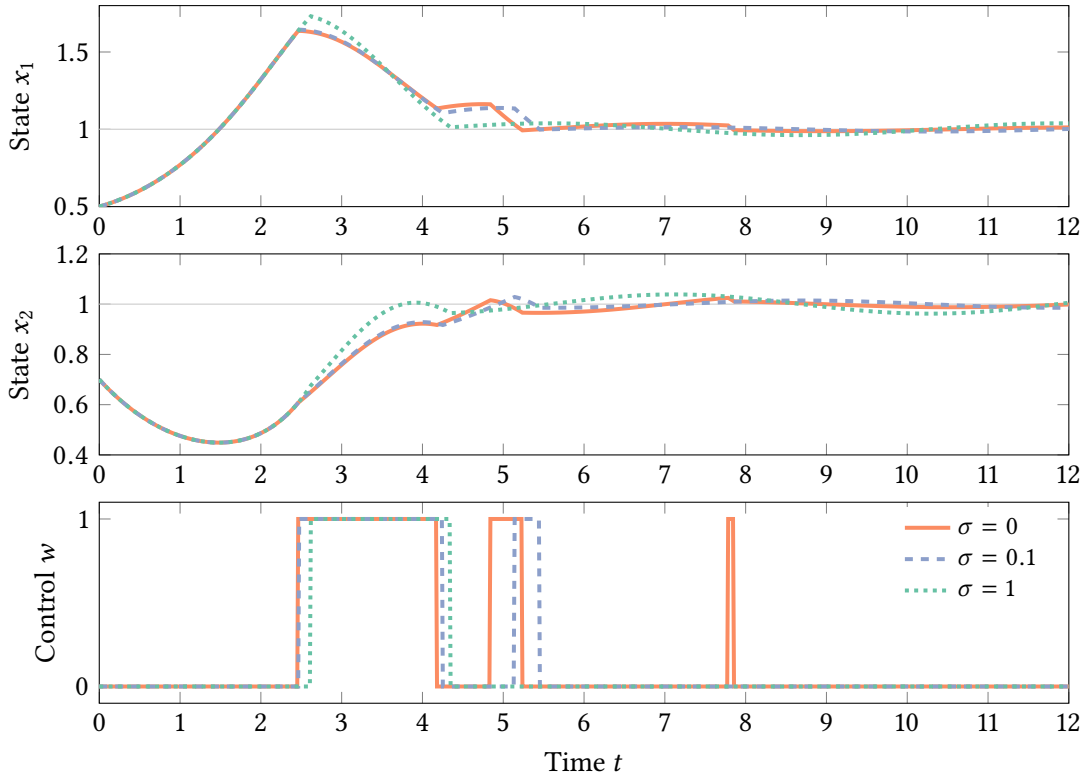


Figure 2.2: Unconstrained fishing problem: state and control trajectories for increasing switching cost σ.

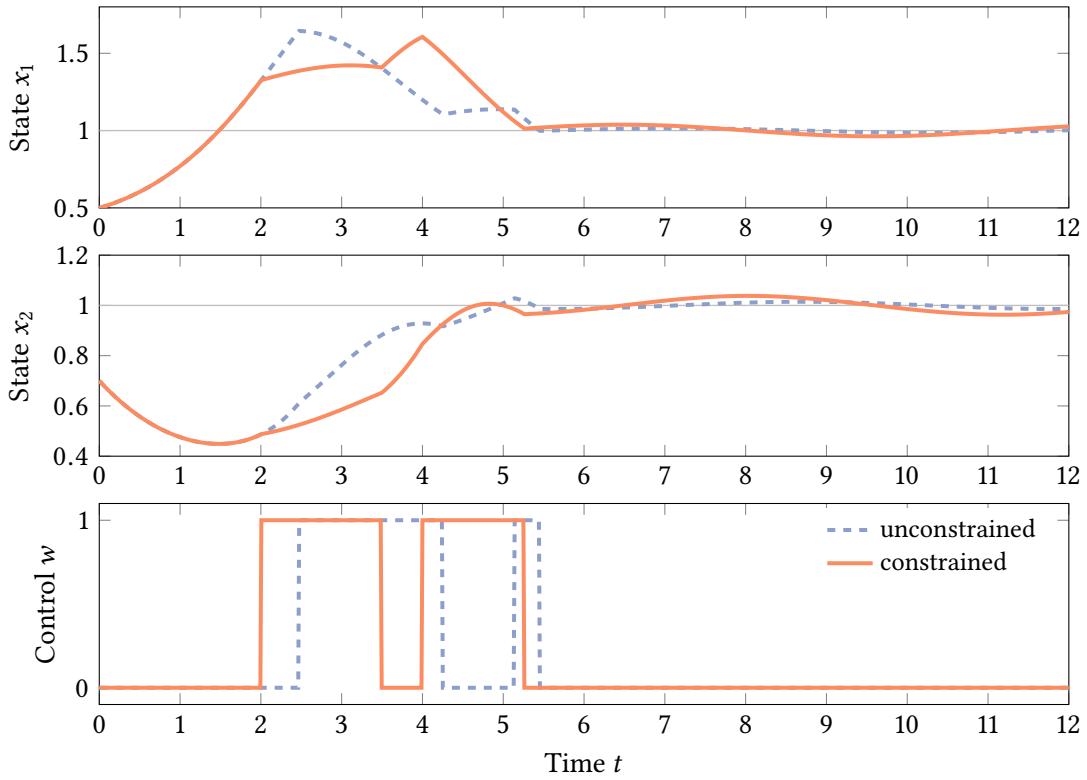


Figure 2.3: Constrained and unconstrained fishing problems: state and control trajectories with switching cost $\sigma = 0.1$.

2.4.2 Machine maintenance problem

The illustrative example presented in this section is concerned with the optimal planning of a machine maintenance throughout its life-cycle. This application was suggested by Eleonora Florian (University of Padua, 2019), kindly acknowledged. Indeed, this underlines the far reaching relevance of switching costs in real applications.

We construct a numerical example based on a continuous-time switched dynamical system; see [172] and references therein. Let us consider a machine (or a production plant, or anything of the kind) that requires some maintenance action. The machine has three operating modes: (0) full production, (1) minor maintenance, and (2) major maintenance. With (0), the machine is fully productive and subject to degradation. With (1), the production is slower, but not necessarily null, while the machine state is improved. With (2), the machine is not productive but its state is quickly restored. The problem under consideration is to plan the operating modes to minimize the total cost of operating and maintaining the machine.

Let us consider the time interval $[0, 1]$ and the system state $\mathbf{x} = (s, p)$. The machine state s and the profit p satisfy $s(t) \in [0, 1]$, with $s = 1$ being the perfect condition, and $p(t) \geq 0$ at all $t \in [0, 1]$. We set the initial state $\mathbf{x}_0 := (1, 0)$ and consider the dynamics governed by $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, w)$, with $w \in \{0, 1, 2\}$. The following simple model is proposed for the three operating modes w:

$$\mathbf{f}(\mathbf{x}, 0) = \begin{pmatrix} -k_1 s \\ k_2 s^2 - k_3 \end{pmatrix}, \qquad \mathbf{f}(\mathbf{x}, 1) = \begin{pmatrix} k_4(1 - s) \\ k_5 s^2 - k_6 \end{pmatrix}, \qquad \mathbf{f}(\mathbf{x}, 2) = \begin{pmatrix} k_7(1 - s^2) \\ -k_8 \end{pmatrix}, \tag{2.4.1}$$

with parameters $\mathbf{k} \in \mathbb{R}^8_+$. The degradation rate in (0) and the reconditioning rate with (1) are linear in the machine state s, while the generated profit is quadratic. Some constant offsets are associated with the production and reconditioning costs. On the other hand, during major maintenance (2), the machine state quickly approaches the unit, that is, the perfect condition, while the profit declines due to high maintenance costs. We found that the parameter values

$$\mathbf{k} = (2, 25, 1, 1, 8, 2, 50, 40)$$

produced a reasonable and interesting response of the model. Since the system dynamics have three operating modes, the sequence selected for the STO formulation may affect the solution, as pointed out in [111, 199, 204]. Owing to the fact that major maintenance is not expected to occur many times, we choose the following control sequence with $N = 17$ switching intervals:

$$\{w_k\} = \{0, 1, 0, 1, 0, 2, 0, 1, 0, 1, 0, 2, 0, 1, 0, 1, 0\}.$$

The objective function aims at maximizing the overall profit, *i.e.*, $p(1)$: we set $\boldsymbol{Q} = \mathbf{0}$ and $\boldsymbol{E} = \mathrm{diag}(0, -1)$. Finally, we set the fixed discretization grid with $n = 400$ time points. The resulting structured optimization problem is solved via `ZeroFPR`, with maximum 100 iterations.

Figure 2.4 reports the machine state, profit, and operating mode for the solution found without switching costs ($\sigma = 0$). Therein, the initial guess is also drawn for comparison. Although switches are not discouraged, the optimal control with $\sigma = 0$ activates only 6 out of $N = 17$ intervals. For the most time, the plant operates in full production mode, requiring the two major maintenance breaks to quickly recover and make the system productive. The initial minor maintenance phase allows to keep the plant in perfect condition before starting the bang-bang-like part of the solution.

Let us now consider the effect of switching cost $\sigma \in \{25, 30\}$. Concurrently with dynamics $\mathbf{f}$ and parameters $\mathbf{k}$, these values have been selected to obtain interesting results; in most cases, the solution found was relatively uninteresting. The solution for $\sigma = 0$ is given to `ZeroFPR` as initial guess. Figure 2.5 depicts the resulting state and control trajectories. With increasing switching cost, the number of activated intervals decreases, as one would expect. With $\sigma = 25$, minor maintenance is held longer and major maintenance is adopted only once. Even more, with $\sigma = 30$, minor maintenance is active most of the time whereas major maintenance is never invoked. These results show the considerable impact switching costs may have on scheduling and planning.

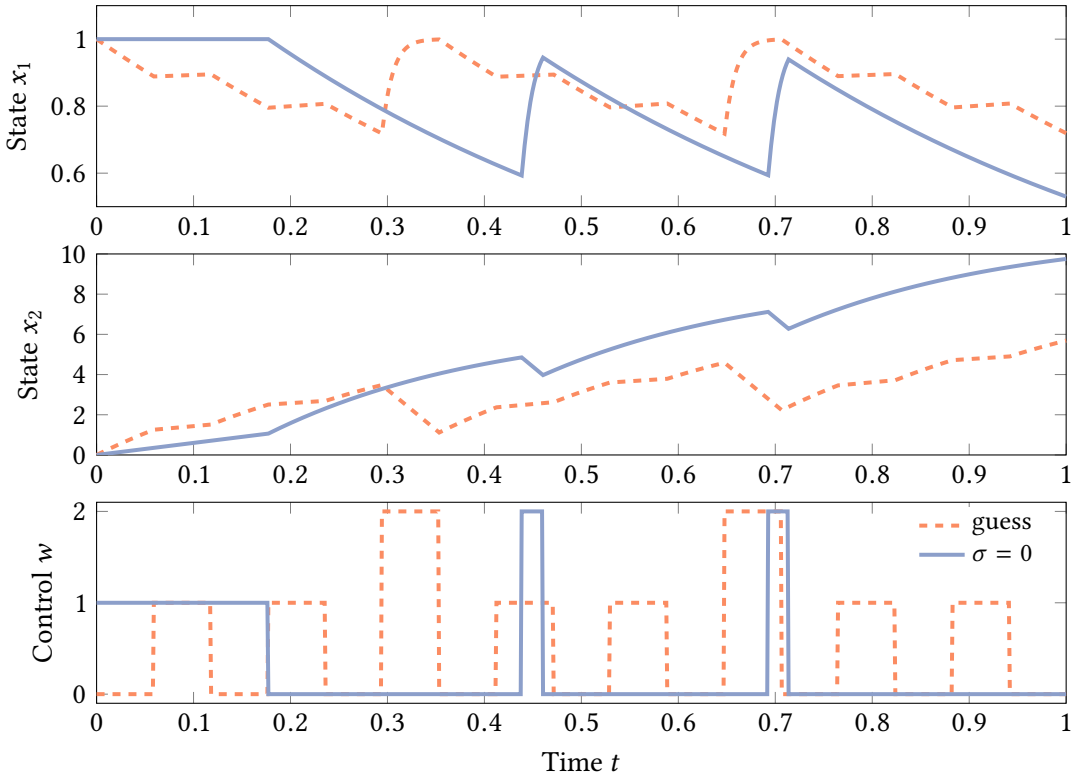


Figure 2.4: Maintenance problem: state and control trajectories without switching cost ($\sigma = 0$); initial guess and retrieved solution.

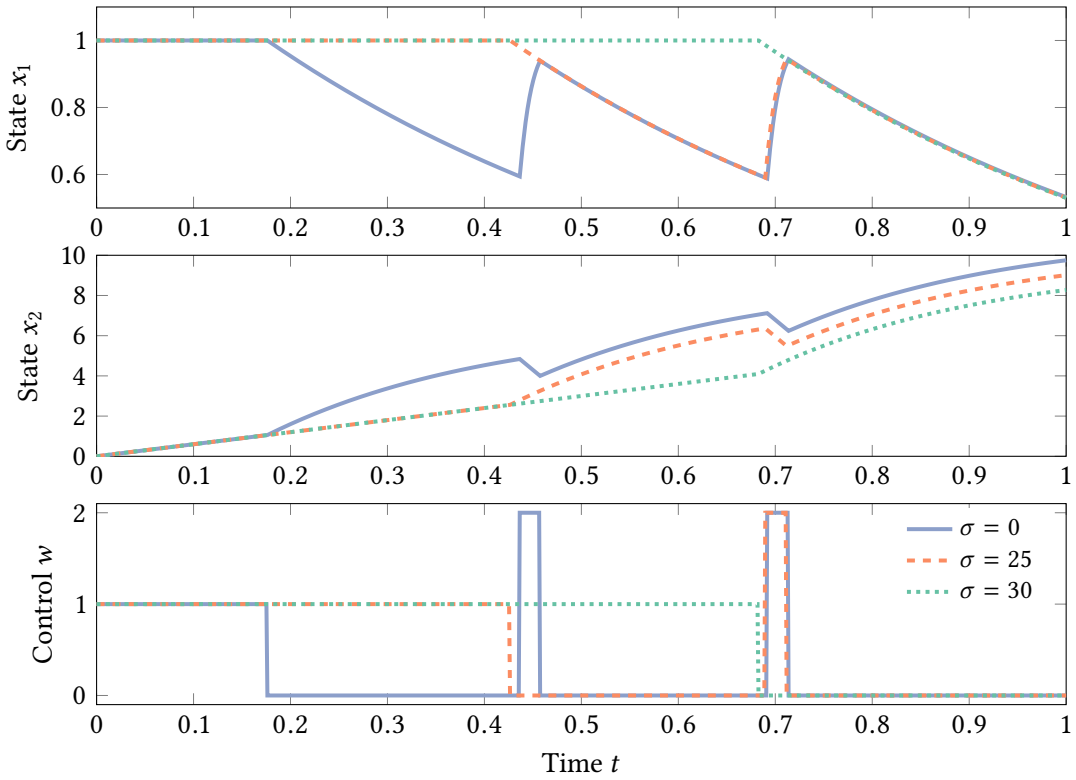


Figure 2.5: Maintenance problem: state and control trajectories for increasing switching cost σ.

2.5 Summary

We presented an approach for dealing with switching time optimization (STO) problems with constraints and switching costs. Borrowing ideas from sparse optimization and numerical optimal control, these are reformulated in the form of constrained structured programs, whose numerical solution has been investigated in Chapter 1. Although the proposed framework suffers from the limitations of STO, it can be extended to deal with nonlinear, constrained, multi-phase mixed-integer optimal control problems with switching costs. Furthermore, coupling our approach with rounding-based methods could greatly mitigate the drawbacks of the former, while improving accuracy of the latter. We implemented a modelling tool for constrained, sparse STO problems in the `Julia` package `ScSTO`. Numerical examples showed the potential of switching costs for extending the current modeling capabilities and their filtering action, ruling out undesired solutions.

Chapter 3

Convex Quadratic Programming

Linear Algebra and Optimization:
Together Forever!
—M. H. Wright [127]

This chapter details a primal-dual proximal Newton-type method for solving convex quadratic programs. The proximal point algorithm and a semismooth Newton's method are effectively weaved together via the primal-dual proximal augmented Lagrangian function.

The content of this chapter partially appears in [236].

3.1 Introduction

Convex programming appears in a variety of applications. Optimization problems of this form are of interest in engineering, statistics, finance and many other fields. In particular, convex quadratic programs (QPs) often arise within more general nonlinear optimization methods [97, 104, 112]. QPs cover many practical applications, greatly vary in terms of problem size and structure, and often have to be solved with limited computing resources and strict time constraints. Grown interest and effort has been recently devoted to solving convex QPs, embracing all these challenges. Methods differ in how they balance the number of iterations and the cost (*e.g.*, run time) per iteration. Interior point methods usually require few but rather demanding iterations [104, 112, 151]. Active set methods take more but cheaper iterations, as factorization updates can be used [166]. On the other hand, first-order methods take many but cheap iterations [185, 253, 87], and several schemes have been proposed to accelerate such methods; cf. [215, 171, 185] and [189, 227]. The augmented Lagrangian framework [49, 62, 112], semismooth Newton's methods [57, 121], and proximal techniques [30, 29, 170] are undergoing a revival, as their seamless combination exhibits valuable properties and provides useful features [194, 208, 224, 243]. Indeed, in this chapter we build upon these methods and introduce `QPDO`, a numerical solver for convex quadratic programming.

Most of the results in this chapter hold, possibly with minor modifications, also in the case of general convex programming. Nonetheless, we focus on convex QPs, for two compelling reasons: convex QPs play a key role in continuous optimization and exhibit plenty of structure. In particular, the latter tempts us to heavily exploit the underlying linear algebra.

Let us consider a general convex quadratic program (QP) in the form

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad & \frac{1}{2}\mathbf{x}^\top \boldsymbol{Q}\mathbf{x} + \mathbf{q}^\top \mathbf{x} \\ \text{subject to} \quad & \mathbf{l} \leq \boldsymbol{A}\mathbf{x} \leq \mathbf{u}. \end{aligned} \tag{3.1.1}$$

Here $\mathbf{x} \in \mathbb{R}^n$ is the decision variable, matrix $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ and vector $\mathbf{q} \in \mathbb{R}^n$ define the objective function, whereas the constraints are encoded by matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and vectors $\mathbf{l}, \mathbf{u} \in \overline{\mathbb{R}}^m$. In the rest of this chapter, without further mention, we assume the following requirements are satisfied; cf. [253, 224].

Assumption 3.1.1. *Matrix* $Q \in \mathrm{Sym}_+(\mathbb{R}^n)$ *and vectors* $\mathbf{l}$ *and* $\mathbf{u}$ *satisfy* $\mathbf{l} \leq \mathbf{u}$, $\mathbf{l} < +\infty$, *and* $\mathbf{u} > -\infty$ *component-wise.*

We will refer to the nonempty, closed and convex set

$$C := \{\mathbf{z} \in \mathbb{R}^m : \mathbf{l} \leq \mathbf{z} \leq \mathbf{u}\}$$

as the constraint set. Note that (3.1.1) represents a general convex QP, in that it can accommodate also equality constraints and bounds.

3.1.1 Background

Convex QPs have been studied since the 1950s [12] and several numerical methods have been developed since then. These differ in how they balance the number of iterations and the cost (*e.g.*, run time) per iteration.

Active-set methods for QPs originated from extending the simplex method for linear programs (LPs) [13]. These methods select a set of binding constraints and iteratively adapt it, seeking the set of active constraints at the solution. Active-set algorithms can be easily warm started and can lead to finite convergence. Moreover, by adding and dropping constraints from the set of binding ones, factorization updates can be adopted for solving successive linear systems. However, these methods may require many iterations to identify the correct set of active constraints. Modern solvers based on active-set methods are `qpOASES` [166] and `NASOQ` [233].

First-order methods iteratively compute an optimal solution using only first-order information about the cost function [171, 185]. As these methods consist of computationally cheap and simple steps, they are well suited to applications with limited computing resources [253]. However, first-order algorithms usually require many iterations to achieve accurate solutions and may suffer from ill-conditioning of the problem data. Several acceleration schemes have been proposed to improve their behaviour [189, 227]. The `OSQP` [253] solver offers an implementation based on ADMM [138].

Interior-point methods consider the problem constraints in the objective function via barrier functions and solve a sequence of parametric subproblems [97, Chap. 11], [112, §16.6]. Although not easily warm started, the polynomial complexity makes interior-point methods appealing for large scale problems [151]. They usually require few but rather demanding iterations [104, 112]. Recent developments are found in the regularized method `IP-PMM` [250].

Semismooth Newton's methods apply a nonsmooth version of Newton's method to the KKT conditions of the original problem [57, 69]. In the strictly convex case, *i.e.*, with $Q > 0$, this approach performs very well as long as the underlying linear systems are nonsingular. Regularized, or stabilized, semismooth Newton-type methods, such as `QPALM` [224, 239] and `FBstab` [243], overcome these drawbacks. Augmented Lagrangian [49, 62, 112] and proximal techniques [30, 170] introduce a regularizing outer layer that enhances numerical stability [208, 224, 243]. These ideas form the basis for our approach.

3.1.2 Approach

We present a numerical method for solving general convex QPs. The proposed algorithm is based on the proximal point algorithm and a semismooth Newton's method for solving the subproblems, which are always solvable for any choice of problem data. We therefore impose no restrictions such as strict convexity of the cost function or linear independence of the constraints. As such, our algorithm gathers together the benefits of fully regularized primal-dual methods and semismooth Newton's methods with active-set structure. Our algorithm can exploit warm starting to reduce the number of iterations, as well as factorization caching and multi-rank update techniques for efficiency, and it can obtain accurate solutions.

Our approach, dubbed `QPDO` from *Quadratic Primal-Dual Optimizer*, is inspired by and shares many characteristics with algorithms that have already been proposed, in particular with `QPALM` [224] and `FBstab` [243]. On the other hand, they differ on some key aspects. `QPALM` relates to the proximal method

of multipliers [224, Rem. 2], which in turn is associated to the classical (primal) augmented Lagrangian function [29]. Instead, FBstab and QPDO apply the proximal point method, yielding exact primal-dual regularization. However, FBstab reformulates the subproblem via the (penalized) Fischer-Burmeister NCP function [53, 82], and adopts the squared residual norm as a merit function for the inner iterative loop; this prevents the use of symmetric sparse linear solvers. Instead, QPDO adopts the minimum NCP function, which leads to symmetric linear systems with active-set structure. Then, we show the primal-dual proximal augmented Lagrangian function, introduced in [114, 149] and [222], is a suitable merit function for the proximal subproblem, which allows us to perform an exact linesearch in a fully primal-dual regularized context. Indeed, we believe, the main contribution of this work consists in recognizing this link, exploiting it to bridge the gap between previously proposed methods, and developing a robust and efficient algorithm that possesses their advantages but does not suffer from their inconveniences.

Outline We sketch our algorithmic framework in §3.2 and develop our method in details in §§ 3.3 and 3.4. In particular, in §3.4.1 we establish our key result, which relates the proximal operator and the primal-dual proximal augmented Lagrangian function. QPDO's convergence properties are analyzed in §3.5, and §3.6 juxtaposes QPDO with similar methods. We present details of our implementation in §3.7 and report on numerical experience and benchmarks in §3.8.

Nota bene In an earlier draft of this thesis, the method was presented from a different standpoint, following the framework developed in Chapter 1. For the sake of a simpler and more direct presentation, we reverted the perspective. Although now it may seem unrelated to previous chapters, we hope this gives a clearer interpretation of the method. Originally, we followed the primal-dual proximal augmented Lagrangian framework discussed in Chapter 1. By convexity, we were then able to show that solving the arising subproblems is equivalent to evaluating the proximal operator of the KKT conditions of the original QP in (3.1.1). This is comparable to going backward through §3.3. Establishing this link allowed us to relax the requirements on parameters and dual estimate updates to those of the proximal point algorithm, eventually obtaining QPDO.

Notation The algorithm is described with a nested structure, whose outer iterations are indexed by $k \in \mathbb{N}$. We denote $\mathbf{y}$ the dual variable associated to the constraints in problem (3.1.1). A primal-dual pair $(\mathbf{x}, \mathbf{y})$ will be denoted by $\mathbf{v}$, and we will refer interchangeably to it as a vector or to its components $\mathbf{x}$ and $\mathbf{y}$. An optimal solution to the problem (3.1.1) will be denoted as $(\mathbf{x}^\star, \mathbf{y}^\star)$, or $\mathbf{v}^\star$. Accordingly, $(\mathbf{x}_k^\star, \mathbf{y}_k^\star)$, or $\mathbf{v}_k^\star$, will denote the solution of the proximal subproblem corresponding to the k-th outer iteration.

3.2 Algorithm

In this section, we outline our Quadratic Primal-Dual Optimizer (QPDO), which weaves together the proximal point algorithm and a semismooth Newton's method. The proposed numerical scheme is sketched in Algorithms 4 and 5, highlighting the nested structure for clarity of presentation. We denote $\mathbf{y}$ a dual variable and $\mathbf{v}$ a primal-dual pair $(\mathbf{x}, \mathbf{y})$; $\mathbf{r}$ and $\mathbf{r}_k$ the outer and inner residuals defined in (3.3.3) and (3.4.5), respectively. In the following sections, we provide more details on the outer and inner procedures and investigate their global and local convergence properties.

Effectively, the proximal operator is evaluated by solving a subproblem via semismooth Newton's method. Thus, the latter constitutes a inner iterative procedure, embedded into the outer proximal point loop. Warm-starting and early termination of these subproblems yield a more efficient method, and deserve some comments.

- Warm-starting is good practice, motivated by the iterative nature of the numerical methods we are interested in. For example, when started close enough to a solution, pure Newton iterations can rapidly converge to it. In Algorithm 5, the inner loop sequence is constructed starting from

Algorithm 4 QPDO: Quadratic Primal-Dual Optimizer

input: Q, $\mathbf{q}$, A, $\mathbf{l}$, $\mathbf{u}$
parameters: $\epsilon > 0$, $\epsilon_0 \geq 0$, $\kappa_\epsilon \in [0, 1)$, $0 < \sigma_{\min} \leq \sigma_0$, $0 < \mu_{\min} \leq \mu_0$
guess: $\mathbf{x}_0 \in \mathbb{R}^n$, $\mathbf{y}_0 \in \mathbb{R}^m$
for $k = 0, 1, 2, \ldots$ **do**
  if $\|\mathbf{r}(\mathbf{v}_k)\|_\infty \leq \epsilon$ **then** ▷ *convergence check*
    return $\mathbf{v}_k$
  end if
  find $\mathbf{v}_{k+1}$ such that $\|\mathbf{r}_k(\mathbf{v}_{k+1})\|_\infty \leq \epsilon_k$ by invoking Algorithm 5 ▷ *subproblem*
  choose $\sigma_{k+1} \in [\sigma_{\min}, \sigma_k]$ and $\mu_{k+1} \in [\mu_{\min}, \mu_k]$ ▷ *parameters update*
  set $\epsilon_{k+1} \leftarrow \kappa_\epsilon \epsilon_k$
end for

Algorithm 5 QPDO's inner loop: semismooth Newton's method

$\mathbf{v} \leftarrow \mathbf{v}_k$ ▷ *warm start*
repeat
  get $\delta\mathbf{v}$ by solving the linear system (3.4.15) ▷ *search direction*
  get τ by solving the piecewise linear equation (3.4.17) ▷ *step size*
  set $\mathbf{v} \leftarrow \mathbf{v} + \tau\, \delta\mathbf{v}$
until $\|\mathbf{r}_k(\mathbf{v})\|_\infty \leq \epsilon_k$ ▷ *inner convergence check*
$\mathbf{v}_{k+1} \leftarrow \mathbf{v}$

the estimate $\mathbf{v}_k$. This initial guess turns out to improve at every outer iteration, due to the contraction properties of the overall method, investigated in §3.5.

- Early termination of the subproblems is convenient for nested algorithms such as Algorithm 4, mainly *because it makes no sense to spend a lot of time solving accurately a subproblem whose solution may be far from the solution of the original problem* [145, p. 955]. The specific stopping criterion stems from the discussion in §3.4.1.

3.3 Outer Loop: Inexact Proximal Point Method

Our method solves problem (3.1.1) using the proximal point algorithm, with inexact evaluation of the proximal operator. In Algorithm 4, this is evaluated by means of a semismooth Newton-type method, which constitutes a inner iterative procedure, further investigated in §3.4. This section focuses on the outer loop corresponding to the proximal point algorithm, which has been extensively studied in the literature [30]. We recall some important results and refer to [14, 29, 43, 169] for more details.

3.3.1 Optimality conditions

Problem (3.1.1) can be equivalently expressed as

$$\underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{x}) + g(A\mathbf{x}), \tag{3.3.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^m \to \overline{\mathbb{R}}$, given by

$$f(\mathbf{x}) := \frac{1}{2}\mathbf{x}^\top Q\mathbf{x} + \mathbf{q}^\top \mathbf{x} \qquad \text{and} \qquad g(\mathbf{z}) := \chi_C(\mathbf{z}),$$

are the objective function and the characteristic function of the constraint set C, respectively. The necessary and sufficient, first-order optimality conditions of problem (3.3.1), and hence problem (3.1.1), read

$$\mathbf{0} \in \mathcal{T}(\mathbf{v}) := \begin{pmatrix} Q\mathbf{x} + \mathbf{q} + A^\top \mathbf{y} \\ -A\mathbf{x} + \partial g^*(\mathbf{y}) \end{pmatrix}, \tag{3.3.2}$$

where ∂g^* denotes the conjugate subdifferential of g [169]. We will denote $\ell := n + m$ and refer to $\mathcal{T} : \mathbb{R}^\ell \rightrightarrows \mathbb{R}^\ell$ as the KKT operator for problem (3.1.1). These optimality conditions, in the form (3.3.2), involve the set-valued operator $\mathcal{T}$. However, noticing that, for any $\alpha > 0$, the conditions $\mathbf{v} = \Pi_C(\mathbf{v} + \alpha\mathbf{u})$ and $\mathbf{v} \in \partial g^*(\mathbf{u})$ are equivalent [70, §23], conditions in (3.3.2) can be equivalently rewritten. Choosing $\alpha = 1$, we can define the (outer) residual $\mathbf{r} : \mathbb{R}^\ell \to \mathbb{R}^\ell$ and express the KKT conditions for (3.1.1) as

$$\mathbf{0} = \mathbf{r}(\mathbf{v}) := \begin{pmatrix} \boldsymbol{Q}\mathbf{x} + \mathbf{q} + \boldsymbol{A}^\top \mathbf{y} \\ \boldsymbol{A}\mathbf{x} - \Pi_C(\boldsymbol{A}\mathbf{x} + \mathbf{y}) \end{pmatrix}. \tag{3.3.3}$$

This reformulation can be obtained also by employing the minimum NCP function [80] and rearranging to obtain the projection operator Π_C. The residual $\mathbf{r}$ is analogous to the natural residual function $\boldsymbol{\pi}$ investigated in [68]. Since it is an error bound for problem (3.1.1), in the sense that $\operatorname{dist}_{\mathcal{T}^{-1}(\mathbf{0})}(\mathbf{v}) = O(\|\mathbf{r}(\mathbf{v})\|)$ [68, Thm 18], the norm of $\mathbf{r}$ is a sensitive optimality measure and its value can be adopted as a stopping criterion.

3.3.2 Proximal point algorithm

The proximal point algorithm [30] finds zeros of maximal monotone operators by recursively applying their proximal operator. Since $\mathcal{T}$ is a maximal monotone operator [29, 169], the proximal point algorithm converges to an element $\mathbf{v}^\star$ of the set of primal-dual solutions $\mathcal{T}^{-1}(\mathbf{0})$, if any exists [14, 30]. Starting from an initial guess $\mathbf{v}_0$, it generates a sequence $\{\mathbf{v}_k\}$ of primal-dual pairs by recursively applying the proximal operator $\mathcal{P}_k$:

$$\mathbf{v}_{k+1} = \mathcal{P}_k(\mathbf{v}_k), \qquad \mathcal{P}_k := \left(\boldsymbol{I} + \Sigma_k^{-1}\mathcal{T}\right)^{-1}. \tag{3.3.4}$$

Here, $\{\Sigma_k\}$ is a sequence of non-increasing positive definite matrices, namely, $\Sigma_k \succ 0$ and $\Sigma_k - \Sigma_{k+1} \succeq 0$ for all $k \in \mathbb{N}$. The matrices Σ_k control the primal-dual proximal regularization and, similarly to exact penalty methods, these are not required to vanish [29, 30]. Since $\mathcal{T}$ is maximally monotone, the proximal operator $\mathcal{P}_k$ is well defined and single valued for all $\mathbf{v} \in \operatorname{dom}\mathcal{T} = \mathbb{R}^\ell$ [14]. Thus, from (3.3.4), evaluating the proximal operator $\mathcal{P}_k$ at $\mathbf{v}_k$ is equivalent to finding the unique $\mathbf{v} \in \mathbb{R}^\ell$ that satisfies

$$\mathbf{0} \in \mathcal{T}_k(\mathbf{v}) := \mathcal{T}(\mathbf{v}) + \Sigma_k(\mathbf{v} - \mathbf{v}_k). \tag{3.3.5}$$

This is guaranteed to have a unique solution and to satisfy certain useful regularity properties; see §3.4 below. As a result, we can construct a fast inner solver for these subproblems based on semismooth Newton's method.

3.3.3 Early termination

The proximal point algorithm tolerates errors, namely the inexact evaluation of the proximal operator $\mathcal{P}_k$ [30]. Criterion (A_r) in [43] provides conditions for the design of convergent inexact proximal point algorithms [43, Thm 2.1]. Let $\mathbf{v}_k^\star := \mathcal{P}_k(\mathbf{v}_k)$ denote the unique proximal subproblem solution and $\mathbf{v}_{k+1} \approx \mathbf{v}_k^\star$ the actual recurrence update. Then, the aforementioned criterion requires

$$\|\mathbf{v}_{k+1} - \mathbf{v}_k^\star\| \leq e_k \min\left(1, \|\mathbf{v}_{k+1} - \mathbf{v}_k\|^r\right),$$

where $r \geq 0$ and the sequence of inner tolerances $\{e_k\} \subseteq \mathbb{R}_+$ is summable, *i.e.*, $\sum_{k=0}^{\infty} e_k < +\infty$. However, since $\mathbf{v}_k^\star$ is effectively unknown, this criterion is impractical in its form. Instead, in Algorithm 4 it is required that $\mathbf{v}_{k+1}$ satisfies $\|\mathbf{r}_k(\mathbf{v}_{k+1})\|_\infty \leq \epsilon_k$. Here, $\mathbf{r}_k$ denotes the residual for the k-th subproblem, and is defined in (3.4.5). In §3.5 we will show that this criterion is a simple and viable substitute, which retains the significance of (A_r).

3.3.4 Warm starting

If a solution $\mathbf{v}^\star$ exists, the (outer) sequence $\{\mathbf{v}_k\}$ generated by (3.3.4) converges, by the global convergence of the proximal point algorithm [30]. Then, expectedly, $\mathcal{P}_k(\mathbf{v}_k)$ and $\mathbf{v}_k$ are arbitrarily close for

sufficiently large k [243, §4]. This supports the idea of warm starting the inner solver with the current outer estimate $\mathbf{v}_k$, that is, setting $\mathbf{v} \leftarrow \mathbf{v}_k$ in Algorithm 5. For large k, only one or few Newton-type inner iterations are needed to find an approximate subproblem solution $\mathbf{v}_{k+1}$.

3.4 Inner Loop: Semismooth Newton's Method

In this section we focus on solving subproblem (3.3.5) via a semismooth Newton's method. For the sake of clarity, and without loss of generality, we consider

$$\Sigma_k := \operatorname{blkdiag}(\sigma_k \boldsymbol{I}_n, \mu_k \boldsymbol{I}_m).$$

for some fixed parameters $\sigma_k, \mu_k \in \mathbb{R}_{++}$.

3.4.1 Merit function

We now derive the simple yet fundamental result that is the key to develop our method. This provides the NCP reformulation of the proximal subproblem with a suitable merit function. The former yields symmetric active-set linear systems, while the latter leads to exact linesearch.

Let us express the proximal subproblem (3.3.5) in the form

$$\mathbf{0} \in \begin{pmatrix} \boldsymbol{Q}\mathbf{x} + \mathbf{q} + \boldsymbol{A}^\top \mathbf{y} + \sigma_k(\mathbf{x} - \mathbf{x}_k) \\ -\boldsymbol{A}\mathbf{x} + \mu_k(\mathbf{y} - \mathbf{y}_k) + \partial g^*(\mathbf{y}) \end{pmatrix}. \tag{3.4.1}$$

Similarly to (3.3.3), for any given $\alpha > 0$, this can be rewritten as

$$\mathbf{0} = \begin{pmatrix} \boldsymbol{Q}\mathbf{x} + \mathbf{q} + \boldsymbol{A}^\top \mathbf{y} + \sigma_k(\mathbf{x} - \mathbf{x}_k) \\ \boldsymbol{A}\mathbf{x} + \mu_k(\mathbf{y}_k - \mathbf{y}) - \Pi_C(\mathbf{w}_k) \end{pmatrix}, \tag{3.4.2}$$

where we denote

$$\mathbf{w}_k := \boldsymbol{A}\mathbf{x} + \mu_k(\mathbf{y}_k - \mathbf{y}) + \alpha\mathbf{y}. \tag{3.4.3}$$

Then, for any positive $\alpha \neq \mu_k$, the conditions in (3.4.2) are equivalent to

$$\mathbf{0} = \begin{pmatrix} \boldsymbol{Q}\mathbf{x} + \mathbf{q} + \frac{1}{\alpha}\boldsymbol{A}^\top[\mathbf{w}_k - \Pi_C(\mathbf{w}_k)] + \sigma_k(\mathbf{x} - \mathbf{x}_k) \\ ((\alpha - \mu_k)/\alpha)[\mathbf{w}_k - \Pi_C(\mathbf{w}_k)] + (\mu_k - \alpha)\mathbf{y} \end{pmatrix}, \tag{3.4.4}$$

namely their unique solution coincides. Now, we observe that the right-hand side of (3.4.4) is the gradient of the function

$$f(\mathbf{x}) + \frac{1}{2\alpha}\operatorname{dist}_C^2(\mathbf{w}_k) + \frac{\sigma_k}{2}\|\mathbf{x} - \mathbf{x}_k\|^2 + \frac{\mu_k - \alpha}{2}\|\mathbf{y}\|^2.$$

By construction, this is a continuously differentiable function whose gradient vanishes at the unique solution of the proximal subproblem. Furthermore, for any $\alpha \in (0, \mu_k)$, it is strictly convex and hence admits a unique minimizer. This must coincide with the unique proximal point. Therefore, this function is a suitable merit function for the subproblem. The particular choice $\alpha := \mu_k/2$ inherits all these properties and leads to the inner optimality conditions

$$\mathbf{0} = \mathbf{r}_k(\mathbf{v}) := \begin{pmatrix} \boldsymbol{Q}\mathbf{x} + \mathbf{q} + \boldsymbol{A}^\top \mathbf{y} + \sigma_k(\mathbf{x} - \mathbf{x}_k) \\ \boldsymbol{A}\mathbf{x} + \mu_k(\mathbf{y}_k - \mathbf{y}) - \Pi_C(\boldsymbol{A}\mathbf{x} + \mu_k(\mathbf{y}_k - \mathbf{y}/2)) \end{pmatrix}, \tag{3.4.5}$$

with $\mathbf{r}_k : \mathbb{R}^\ell \to \mathbb{R}^\ell$ the inner residual, and the associated merit function

$$\mathcal{M}_k(\mathbf{v}) := f(\mathbf{x}) + \frac{1}{\mu_k}\operatorname{dist}_C^2(\boldsymbol{A}\mathbf{x} + \mu_k(\mathbf{y}_k - \mathbf{y}/2)) + \frac{\sigma_k}{2}\|\mathbf{x} - \mathbf{x}_k\|^2 + \frac{\mu_k}{4}\|\mathbf{y}\|^2. \tag{3.4.6}$$

In fact, $\mathcal{M}_k : \mathbb{R}^\ell \to \mathbb{R}$ is the primal-dual proximal augmented Lagrangian function in (1.4.19), up to the constant term $-\mu\|\overline{\mathbf{y}}\|^2/2$, with $\mathbf{c}(\mathbf{x}) := \boldsymbol{A}\mathbf{x}$ and $S := C$. This demonstrates the link to the framework

developed in Chapter 1. Furthermore, this underlines once again the strong relationship between the proximal point algorithm and the augmented Lagrangian framework, pioneered in [29]. On the one hand, by (3.4.6), the dual regularization parameter μ_k controls the constraint penalization [148, §3.2]. On the other hand, this provides an *interpretation of the augmented Lagrangian method as an adaptive constraint regularization process* [205, §2].

The inner residual $\mathbf{r}_k$ in (3.4.5) is piecewise affine, hence strongly semismooth on $\mathbb{R}^\ell$ [69, 167]. Effectively, it can be employed as stopping criterion in place of $\|\nabla \mathcal{M}_k(\cdot)\|$. In fact, given the unique, bounded, and nonsingular matrix $\boldsymbol{T}_k$ defined by

$$\boldsymbol{T}_k := \begin{bmatrix} \boldsymbol{I} & \frac{2}{\mu_k}\boldsymbol{A}^\top \\ \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix}, \tag{3.4.7}$$

we have the identity

$$\nabla \mathcal{M}_k(\mathbf{v}) = \boldsymbol{T}_k \mathbf{r}_k(\mathbf{v}) \quad \forall \mathbf{v} \in \mathbb{R}^\ell. \tag{3.4.8}$$

The availability of a suitable merit function allows us to adopt a damped Newton-type method and design a linesearch-based globalization strategy, in contrast with [96, 121, 243]. Since $\mathcal{M}_k$ is continuously differentiable and piecewise quadratic, an exact linesearch procedure can be carried out, which yields finite convergence [71].

The following result characterizes these subproblems, entailing the minimization of $\mathcal{M}_k$, and provides useful properties that hold for general convex programs.

Lemma 3.4.1. *Let any $\mathbf{v}_k \in \mathbb{R}^\ell$ and parameters $\sigma_k, \mu_k \in \mathbb{R}_{++}$ be given. Then,*

(i) $\nabla \mathcal{M}_k$ is Lipschitz continuous, i.e., *there exists $L_k \in \mathbb{R}_{++}$ such that*

$$\|\nabla \mathcal{M}_k(\mathbf{v}) - \nabla \mathcal{M}_k(\mathbf{u})\| \leq L_k \|\mathbf{v} - \mathbf{u}\| \qquad \forall \mathbf{v},\ \forall \mathbf{u} \in \mathbb{R}^\ell. \tag{3.4.9}$$

(ii) $\mathcal{M}_k$ is strictly convex, i.e., *there exists $\omega_k \in \mathbb{R}_{++}$ such that*

$$\boldsymbol{M}_k \geq \omega_k \boldsymbol{I} \qquad \forall \boldsymbol{M}_k \in \partial^2 \mathcal{M}_k(\mathbf{v}),\ \forall \mathbf{v} \in \mathbb{R}^\ell. \tag{3.4.10}$$

In particular, it is $\omega_k \geq \min(\sigma_k, \mu_k/2) > 0$.

Proof. Direct derivation gives that $\nabla \mathcal{M}_k$ is the composition of Lipschitz continuous terms, since the constraint set C is convex and the parameters $\sigma_k, \mu_k > 0$. By Assumption 3.1.1, $\mathcal{M}_k$ is the sum of convex terms; cf. (3.4.6). The lower bound on ω_k follows from the regularization terms $\sigma_k \|\mathbf{x} - \mathbf{x}_k\|^2/2$ and $\mu_k \|\mathbf{y}\|^2/4$. □

Finally, we highlight that the method asymptotically reduces to a sequence of regularized semismooth Newton's steps applied to the original, unperturbed optimality system, on the vein of [190]. This closely relates to the concept of exact regularization [120]. It turns out the proximal primal-dual regularization is in fact exact; see Proposition 3.4.2 and compare [205, Thm 1].

Proposition 3.4.2. *Let $k \in \mathbb{N}$ be arbitrary.*

(i) Suppose $\mathbf{v}_k^\star$ solves subproblem (3.4.5) for $\mathbf{v}_k := \mathbf{v}_k^\star$ and for some $\sigma_k \geq 0$ and $\mu_k > 0$. Then, $\mathbf{v}_k^\star$ solves the original problem (3.3.3).

(ii) Alternatively, suppose $\mathbf{v}_k^\star$ solves subproblem (3.4.5) for $\mathbf{y}_k := \mathbf{y}_k^\star$, $\sigma_k := 0$, and for some $\mu_k > 0$. Then, $\mathbf{v}_k^\star$ solves the original problem (3.3.3).

(iii) Conversely, suppose $\mathbf{v}^\star$ solves the original problem (3.3.3). Then, $\mathbf{v}^\star$ solves the subproblem (3.4.5) for $\mathbf{v}_k := \mathbf{v}^\star$ and for any $\sigma_k \geq 0$ and $\mu_k > 0$.

Proof. The proof is immediate by direct comparison of (3.3.3) and (3.4.5). □

3.4.2 Search direction

A semismooth Newton's direction $\delta\mathbf{v} = (\delta\mathbf{x}, \delta\mathbf{y})$ at $\mathbf{v} = (\mathbf{x}, \mathbf{y})$ solves

$$\boldsymbol{V}_k(\mathbf{v})\delta\mathbf{v} = -\mathbf{r}_k(\mathbf{v}). \tag{3.4.11}$$

Here, the matrix $\boldsymbol{V}_k(\mathbf{v})$ is an element of the generalized Jacobian [70, §23] of $\mathbf{r}_k$ at $\mathbf{v}$, which has the form

$$\boldsymbol{V}_k(\mathbf{v}) = \begin{bmatrix} \boldsymbol{Q} + \sigma_k \boldsymbol{I} & \boldsymbol{A}^\top \\ (\boldsymbol{I} - \boldsymbol{P}_k(\mathbf{v}))\boldsymbol{A} & -\mu_k(\boldsymbol{I} - \boldsymbol{P}_k(\mathbf{v})/2) \end{bmatrix}. \tag{3.4.12}$$

In turn, the diagonal matrix $\boldsymbol{P}_k(\mathbf{v})$ with entries

$$\boldsymbol{P}_k^{ii}(\mathbf{v}) := \begin{cases} 1 & \text{if } \mathbf{l}^i < \mathbf{w}_k^i < \mathbf{u}^i \\ 0 & \text{otherwise} \end{cases}, \quad i = 1, \dots, m, \tag{3.4.13}$$

is an element of the generalized Jacobian of Π_C at $\mathbf{w}_k$, namely $\boldsymbol{P}_k(\mathbf{v}) \in \partial\,\Pi_C(\mathbf{w}_k)$. By selecting $\boldsymbol{P}_k^{ii}(\mathbf{v}) = 0$ for any $\mathbf{v} \in \mathbb{R}^\ell$ when $\mathbf{l}^i = \mathbf{u}^i$, definition (3.4.13) is consistent with equality constraints.

Direct calculation shows that, for any $\mathbf{v} \in \mathbb{R}^\ell$, the matrix

$$\begin{aligned} \boldsymbol{M}_k(\mathbf{v}) &:= \boldsymbol{T}_k \boldsymbol{V}_k(\mathbf{v}) \\ &= \begin{bmatrix} \boldsymbol{Q} + \sigma_k \boldsymbol{I} + \frac{2}{\mu_k}\boldsymbol{A}^\top(\boldsymbol{I} - \boldsymbol{P}_k(\mathbf{v}))\boldsymbol{A} & \boldsymbol{A}^\top(\boldsymbol{P}_k(\mathbf{v}) - \boldsymbol{I}) \\ (\boldsymbol{P}_k(\mathbf{v}) - \boldsymbol{I})\boldsymbol{A} & \mu_k(\boldsymbol{I} - \boldsymbol{P}_k(\mathbf{v})/2) \end{bmatrix} \end{aligned} \tag{3.4.14}$$

is an element of the generalized Hessian of $\mathcal{M}_k$ at $\mathbf{v}$, namely $\boldsymbol{M}_k(\mathbf{v}) \in \partial^2 \mathcal{M}_k(\mathbf{v})$. Since $\boldsymbol{T}_k$ is independent from $\mathbf{v}$, this directly follows from the identity (3.4.8). We highlight that, by considering linear system $\boldsymbol{V}_k(\mathbf{v})\delta\mathbf{v} = -\mathbf{r}_k(\mathbf{v})$ instead of the equivalent $\boldsymbol{M}_k(\mathbf{v})\delta\mathbf{v} = -\nabla\mathcal{M}_k(\mathbf{v})$, we can avoid without further transformations the possibly dense term $\boldsymbol{A}^\top(\boldsymbol{I} - \boldsymbol{P}_k(\mathbf{v}))\boldsymbol{A}$, which may destroy the problem's sparsity.

Owing to the selection of $\boldsymbol{P}_k(\mathbf{v})$ with binary entries, the linear system (3.4.11) can be rewritten in symmetric form, similar to those arising in active-set methods [92]. To this end, we notice that, if $\boldsymbol{P}_k^{ii}(\mathbf{v}) = 1$, the corresponding inner residual in (3.4.5) simplifies into $\mathbf{r}_k^{n+i}(\mathbf{v}) = -\mu_k \mathbf{y}^i/2$, and the linear equation in (3.4.11) gives $\delta\mathbf{y}^i = -\mathbf{y}^i$. This yields the crucial observation that, by (3.4.13), it holds $\boldsymbol{P}_k(\mathbf{v})\delta\mathbf{y} = -\boldsymbol{P}_k(\mathbf{v})\mathbf{y}$ for all $\mathbf{v} \in \mathbb{R}^\ell$. Then, an equivalent yet symmetric linear system is obtained, whose solution is the search direction $\delta\mathbf{v}$ at $\mathbf{v}$:

$$\begin{bmatrix} \boldsymbol{Q} + \sigma_k \boldsymbol{I} & \boldsymbol{A}^\top(\boldsymbol{I} - \boldsymbol{P}_k(\mathbf{v})) \\ (\boldsymbol{I} - \boldsymbol{P}_k(\mathbf{v}))\boldsymbol{A} & -\mu_k(\boldsymbol{I} - \boldsymbol{P}_k(\mathbf{v})/2) \end{bmatrix} \begin{pmatrix} \delta\mathbf{x} \\ \delta\mathbf{y} \end{pmatrix} = \begin{pmatrix} \boldsymbol{A}^\top \boldsymbol{P}_k(\mathbf{v})\mathbf{y} \\ \mathbf{0} \end{pmatrix} - \mathbf{r}_k(\mathbf{v}). \tag{3.4.15}$$

The active-set structure introduced by $\boldsymbol{P}_k$ allows us to obtain a symmetric linear system and adopt multi-rank factorization updates [26, 84] while maintaining structure and sparsity of the coefficient matrix [253, 233]. Factorizing the coefficient matrix can take significant effort, often the vast part for solving a linear system. Thus, when solving a sequence of related linear systems, it is advisable to employ factorization updates, whenever possible [26, 117, 119], avoiding a full re-factorization at each and every iteration. Similarly, problem-specific structures should be exploited, such as blocks and sparsity pattern; see *e.g.* [147, 231]. The coefficient matrix in (3.4.15) is symmetric quasi-definite [61], since it has the form

$$\begin{bmatrix} \boldsymbol{R} & \boldsymbol{S}^\top \\ \boldsymbol{S} & -\boldsymbol{P} \end{bmatrix}, \qquad \text{with } \boldsymbol{R},\ \boldsymbol{P} \succ 0.$$

Thus, it always admits an $LDL^\top$ factorization, with a diagonal intermediate matrix $\boldsymbol{D}$ and no need for pivoting, and the linear system (3.4.15) always has a unique solution [61]. Despite these useful properties and many relevant applications, techniques for updating sparse $LDL^\top$ factorizations are currently lacking [26, 101, 117] or limited to rank-one updates [239, 233]; these would greatly improve the performance of our method. In a different spirit, one could opt for iterative methods tailored to symmetric quasi-definite linear systems [176, 198, 245].

Before proceeding, we show that, for every $\mathbf{v}_j$ and $\boldsymbol{V}_j \in \partial\mathbf{r}_k(\mathbf{v}_j)$, a direction $\delta\mathbf{v}_j$ exists, is unique, and indeed a good search direction, namely a direction of descent for $\mathcal{M}_k$ at $\mathbf{v}_j$.

Lemma 3.4.3. *Let any $\mathbf{v}_k, \mathbf{v} \in \mathbb{R}^\ell$ be given. Denote $\delta\mathbf{v} \in \mathbb{R}^\ell$ a solution to the linear system $V\delta\mathbf{v} = -\mathbf{r}_k(\mathbf{v})$ for some $V \in \partial\mathbf{r}_k(\mathbf{v})$. Then, for all $V \in \partial\mathbf{r}_k(\mathbf{v})$,*

(i) $\delta\mathbf{v}$ exists and is unique,

(ii) $\delta\mathbf{v} = \mathbf{0}$ if and only if $\mathbf{r}_k(\mathbf{v}) = \mathbf{0}$,

(iii) if $\mathbf{r}_k(\mathbf{v}) \neq \mathbf{0}$, $\delta\mathbf{v}$ is a descent direction for $\mathcal{M}_k$ at $\mathbf{v}$, namely

$$\delta\mathbf{v}^\top \nabla\mathcal{M}_k(\mathbf{v}) < 0.$$

Proof. Let $V \in \partial\mathbf{r}_k(\mathbf{v})$ be arbitrary. As matrix T_k defined in (3.4.7) is nonsingular, a vector $\delta\mathbf{v}$ solves $V\delta\mathbf{v} = -\mathbf{r}_k(\mathbf{v})$ if and only if, by (3.4.8), it satisfies the linear system $M\delta\mathbf{v} = -\nabla\mathcal{M}_k(\mathbf{v})$ with $M := T_k V$. The latter admits a unique solution, since $\partial^2\mathcal{M}_k(\mathbf{v}) \ni M > 0$ by Lemma 3.4.1. This proves (i), and the particular case (ii) easily follows. Then, by (ii), for any $\mathbf{r}_k(\mathbf{v}) \neq \mathbf{0}$ it is $\delta\mathbf{v} \neq \mathbf{0}$. Hence, it is

$$\delta\mathbf{v}^\top \nabla\mathcal{M}_k(\mathbf{v}) = -\delta\mathbf{v}^\top M \delta\mathbf{v} < 0$$

for any $\delta\mathbf{v} \neq \mathbf{0}$, concluding the proof. □

3.4.3 Exact linesearch

Given a primal-dual pair $\mathbf{v}$ and a search direction $\delta\mathbf{v}$, we seek a stepsize $\tau > 0$ to effectively update $\mathbf{v} \leftarrow \mathbf{v} + \tau\,\delta\mathbf{v}$ in Algorithm 5. Similarly to $\mathcal{M}_k$, the function $\psi_k : \tau \mapsto \mathcal{M}_k(\mathbf{v} + \tau\delta\mathbf{v})$ is continuously differentiable, piecewise quadratic, and strictly convex. Thus, the optimal stepsize $\tau := \arg\min_{t\in\mathbb{R}} \psi_k(t)$ is found as the unique zero of ψ_k', *i.e.*, $\psi_k'(\tau) = 0$. By direct calculation from (3.4.6), for all $\tau \in \mathbb{R}$ we have

$$\begin{aligned}
\psi'(\tau) &= \delta\mathbf{v}^\top \nabla\mathcal{M}(\mathbf{v} + \tau\delta\mathbf{v}) \\
&= \begin{pmatrix}\delta\mathbf{x}\\ \delta\mathbf{y}\end{pmatrix}^\top \begin{pmatrix} Q(\mathbf{x}+\tau\delta\mathbf{x}) + \mathbf{q} + \frac{2}{\mu_k}A^\top\left[\mathbf{w}_k + \tau\delta\mathbf{w}_k - \Pi_C\left(\mathbf{w}_k + \tau\delta\mathbf{w}_k\right)\right] + \sigma_k(\mathbf{x}+\tau\delta\mathbf{x} - \mathbf{x}_k) \\ -\left[A(\mathbf{x}+\tau\delta\mathbf{x}) + \mu_k(\mathbf{y}_k - \mathbf{y} - \tau\delta\mathbf{y}) - \Pi_C\left(\mathbf{w}_k + \tau\delta\mathbf{w}_k\right)\right] \end{pmatrix} \\
&= \delta\mathbf{x}^\top\left[Q\mathbf{x} + \mathbf{q} + \sigma_k(\mathbf{x} - \mathbf{x}_k)\right] + \frac{\mu_k}{2}\delta\mathbf{y}^\top\mathbf{y} + \tau\delta\mathbf{x}^\top\left(Q + \sigma_k I\right)\delta\mathbf{x} + \tau\frac{\mu_k}{2}\delta\mathbf{y}^\top\delta\mathbf{y} \\
&\quad + \left[\frac{2}{\mu_k}A\delta\mathbf{x} - \delta\mathbf{y}\right]^\top\left[\mathbf{w}_k + \tau\delta\mathbf{w}_k - \Pi_C\left(\mathbf{w}_k + \tau\delta\mathbf{w}_k\right)\right] \\
&= \alpha_k\tau + \beta_k + \frac{2}{\mu_k}\delta\mathbf{w}_k^\top\left[\mathbf{w}_k + \tau\delta\mathbf{w}_k - \Pi_C\left(\mathbf{w}_k + \tau\delta\mathbf{w}_k\right)\right],
\end{aligned}$$

whose coefficients are given by

$$\alpha_k := \delta\mathbf{x}^\top(Q + \sigma_k I)\delta\mathbf{x} + \mu_k\delta\mathbf{y}^\top\delta\mathbf{y}/2 \tag{3.4.16a}$$

$$\beta_k := \delta\mathbf{x}^\top[Q\mathbf{x} + \mathbf{q} + \sigma_k(\mathbf{x} - \mathbf{x}_k)] + \mu_k\delta\mathbf{y}^\top\mathbf{y}/2 \tag{3.4.16b}$$

$$\mathbf{w}_k := A\mathbf{x} + \mu_k\left(\mathbf{y}_k - \mathbf{y}/2\right) \tag{3.4.16c}$$

$$\delta\mathbf{w}_k := A\delta\mathbf{x} - \mu_k\delta\mathbf{y}/2. \tag{3.4.16d}$$

Thus, the exact linesearch procedure amounts to solving a piecewise linear equation of the form

$$0 = \alpha_k\tau + \beta_k + \frac{2}{\mu_k}\delta\mathbf{w}^\top\left[\mathbf{w}_k + \tau\delta\mathbf{w}_k - \Pi_C\left(\mathbf{w}_k + \tau\delta\mathbf{w}_k\right)\right] \tag{3.4.17}$$

with respect to $\tau \in \mathbb{R}$. Thanks to its peculiar structure, (3.4.17) can be solved efficiently and exactly (up to numerical precision), *e.g.*, by sorting and linear interpolation, cf. [224, Alg. 2]. We underline that the stepsize τ is unique and strictly positive. In fact, by the strict convexity of $\mathcal{M}_k$ and by Lemma 3.4.3, we have that ψ_k' is strictly increasing and $\psi_k'(0) = \delta\mathbf{v}^\top\nabla\mathcal{M}_k(\mathbf{v}) < 0$, respectively. Thus, the optimal stepsize τ must be unique and positive.

Armijo's linesearch A viable alternative to the exact linesearch is a linesearch procedure with Armijo's sufficient decrease condition [19]. For a stepsize $\tau > 0$ to be accepted, given some $\eta \in (0, 1/2)$, this requires

$$\mathcal{M}_k(\mathbf{v} + \tau\delta\mathbf{v}) \leq \mathcal{M}_k(\mathbf{v}) + \tau\eta\delta\mathbf{v}^\top \nabla\mathcal{M}_k(\mathbf{v}). \tag{3.4.18}$$

Notice that this is well-defined, since $\delta\mathbf{v}$ is a direction of descent for $\mathcal{M}_k$ at $\mathbf{v}$, by Lemma 3.4.3. Some comments are in order. Although performing an exact linesearch procedure is advisable, it may be not necessarily the most convenient option. For example, when sufficiently close to a subproblem solution, it could be preferable to check whether the unit stepsize $\tau = 1$ could be taken or not, according to Armijo's rule, instead of sorting an array, possibly long. For QPs, the linesearch procedure may change between iterations, dynamically selected based, *e.g.*, on the problem size, regularization parameters, and current iterate. In the broader context of convex programming, although most of these ideas can be readily adapted, an exact linesearch procedure may be an inappropriate choice. For this reason, in §3.5 we establish convergence results for both, the exact and Armijo's linesearch procedures.

3.5 Convergence Analysis

This section discusses the convergence of `QPDO` as outlined in Algorithms 4 and 5, under Assumption 3.1.1. Our analysis relies on well-established results for Newton's and proximal point methods; in particular, we refer to [30, 43, 71]. Recall that indices k and j denote outer and inner iterations, respectively. We write index j alone when referring the k-th outer iteration, with k fixed and clear from the context; nonetheless, quantities indexed by j depend on k too.

3.5.1 Inner loop

First, we focus on the inner loop, described in Algorithm 5 and detailed in §3.4. Since linear system (3.4.15) is always solvable, the search direction $\delta\mathbf{v}$ exists and is unique. Similarly, there exists a unique optimal stepsize $\tau \in \mathbb{R}_{++}$ which solves (3.4.17). Thus, all steps are well-defined. It remains to show that the condition $\|\mathbf{r}_k(\mathbf{v})\|_\infty \leq \epsilon_k$ is eventually satisfied. Since $\mathcal{M}_k$ is continuously differentiable, strictly convex, and piecewise quadratic, the semismooth Newton's method with exact linesearch exhibits finite convergence [71, Thm 3]. Thus, $\nabla\mathcal{M}_k(\mathbf{v}) = \mathbf{0}$ after finitely many iterations. Then, by the identity in (3.4.8) with $\boldsymbol{T}_k$ nonsingular, $\mathbf{r}_k(\mathbf{v}) = \mathbf{0}$. Hence, for any $\epsilon_k > 0$, the inner stopping criterion is eventually satisfied, and the inner loop terminates.

We now discuss the convergence of the inner loop with backtracking linesearch procedure and Armijo's sufficient decrease condition. These results are included as they could be readily extended to convex programming, where performing exact linesearch is usually avoided.

Lemma 3.5.1. *Let any $\mathbf{v}_k, \mathbf{v} \in \mathbb{R}^\ell$ and $\eta \in (0, 1/2)$ be given. Denote $\delta\mathbf{v} \in \mathbb{R}^\ell$ the search direction at $\mathbf{v}$. Then, the Armijo's sufficient decrease condition (3.4.18) is satisfied by any stepsize $\tau \in [0, \overline{\tau}_k]$ with*

$$\overline{\tau}_k := 2(1 - \eta)\omega_k / L_k, \tag{3.5.1}$$

where $L_k, \omega_k \in \mathbb{R}_{++}$ are defined in Lemma 3.4.1. In particular, it is $\overline{\tau}_k > 0$. Furthermore, within Algorithm 4, $\{\overline{\tau}_k\} \subseteq \mathbb{R}_{++}$ is bounded away from zero.

Proof. By Lipschitz continuity and strict convexity of $\mathcal{M}_k$ (cf. Lemma 3.4.1), we have

$$\mathcal{M}_k(\mathbf{v} + \tau\delta\mathbf{v}) \leq \mathcal{M}_k(\mathbf{v}) + \nabla\mathcal{M}_k^\top(\mathbf{v})\tau\delta\mathbf{v} + \frac{L_k}{2}\|\tau\delta\mathbf{v}\|^2$$
$$\delta\mathbf{v}^\top \boldsymbol{M}_k(\mathbf{v})\delta\mathbf{v} \geq \omega_k\|\delta\mathbf{v}\|^2$$

for all $\tau \in \mathbb{R}$. Together with $\boldsymbol{M}_k(\mathbf{v})\delta\mathbf{v} = -\nabla\mathcal{M}_k(\mathbf{v})$, these imply

$$\begin{aligned}\eta\tau\delta\mathbf{v}^\top\nabla\mathcal{M}_k(\mathbf{v}) &= \tau\delta\mathbf{v}^\top\nabla\mathcal{M}_k(\mathbf{v}) - \tau(1-\eta)\delta\mathbf{v}^\top\nabla\mathcal{M}_k(\mathbf{v}) \\ &= \tau\delta\mathbf{v}^\top\nabla\mathcal{M}_k(\mathbf{v}) + \tau(1-\eta)\delta\mathbf{v}^\top\boldsymbol{M}_k(\mathbf{v})\delta\mathbf{v} \\ &\geq \tau\delta\mathbf{v}^\top\nabla\mathcal{M}_k(\mathbf{v}) + \tau(1-\eta)\omega_k\|\delta\mathbf{v}\|^2 \\ &\geq \mathcal{M}_k(\mathbf{v}+\tau\delta\mathbf{v}) - \mathcal{M}_k(\mathbf{v}) - \frac{L_k}{2}\|\tau\delta\mathbf{v}\|^2 + \tau(1-\eta)\omega_k\|\delta\mathbf{v}\|^2.\end{aligned}$$

Rearranging and considering $\tau \in [0, \overline{\tau}_k]$ yield

$$\begin{aligned}\mathcal{M}_k(\mathbf{v}+\tau\delta\mathbf{v}) &\leq \mathcal{M}_k(\mathbf{v}) + \eta\tau\delta\mathbf{v}^\top\nabla\mathcal{M}_k(\mathbf{v}) + \tau\left[\tau\frac{L_k}{2} - (1-\eta)\omega_k\right]\|\delta\mathbf{v}\|^2 \\ &\leq \mathcal{M}_k(\mathbf{v}) + \eta\tau\delta\mathbf{v}^\top\nabla\mathcal{M}_k(\mathbf{v}),\end{aligned}$$

since the last term is nonpositive, thus showing that (3.4.18) is satisfied.

In Algorithm 4, it is $\sigma_k \geq \sigma_{\min} > 0$ and $\mu_k \geq \mu_{\min} > 0$ for all $k \in \mathbb{N}$. Then, by Lemma 3.4.1, $\{\omega_k\}$ is bounded away from zero and $\{L_k\}$ is bounded from above. Since $\eta < 1$, it follows that $\{\overline{\tau}_k\}$ is also bounded away from zero. □

Theorem 3.5.2. *Let any $\mathbf{v}_k \in \mathbb{R}^\ell$ be given. Let $\{\mathbf{v}_j\}$ be the sequence generated by Algorithm 5 with Armijo's linesearch. Then, the sequence $\{\mathbf{v}_j\}$ is well-defined and converges to $\mathbf{v}_k^\star$, the unique minimizer of $\mathcal{M}_k$.*

Proof. Let us suppose Algorithm 5 generates an infinite sequence $\{\mathbf{v}_j\}$ such that $\mathbf{r}_k(\mathbf{v}_j) \neq \mathbf{0}$; otherwise, if $\mathbf{r}_k(\mathbf{v}_j) = \mathbf{0}$, it is $\mathbf{v}_j = \mathbf{v}_k^\star$, due to uniqueness; cf. §3.4.1. Then, the sequence $\{\mathbf{v}_j\}$ is uniquely defined because linear system (3.4.15) is always solvable (cf. §3.4.2) and the linesearch procedure always terminates with a positive stepsize (cf. §3.4.3). Recall that the condition $\mathbf{r}_k(\mathbf{v}) = \mathbf{0}$ corresponds to the unconstrained minimization of $\mathcal{M}_k$, which we use as a merit function (cf. §3.4.1). The sequence $\{\mathcal{M}_k(\mathbf{v}_j)\}$ is decreasing because search directions $\delta\mathbf{v}_j$ are descent directions, by Lemma 3.4.3, and hence there exist positive stepsizes τ_j yielding sufficient decrease. Then, for some $\eta \in (0, 1/2)$, it is

$$\mathcal{M}_k(\mathbf{v}_{j+1}) - \mathcal{M}_k(\mathbf{v}_j) \leq \eta\tau_j\delta\mathbf{v}_j^\top\nabla\mathcal{M}_k(\mathbf{v}_j) < 0.$$

As $\mathcal{M}_k$ is continuous and the optimal value is attained, the sequence $\{\mathcal{M}_k(\mathbf{v}_j)\}$ must converge to some limit and $\lim_{j\to\infty}\mathcal{M}_k(\mathbf{v}_{j+1}) - \mathcal{M}_k(\mathbf{v}_j) = 0$. Thus, $\{\tau_j\delta\mathbf{v}_j^\top\nabla\mathcal{M}_k(\mathbf{v}_j)\}$ decays to zero. However, the stepsizes τ_j are bounded away from zero because, since $\tau_j \geq \overline{\tau}_k > 0$ always yields Armijo's sufficient decrease (cf. Lemma 3.5.1). Thus, $\{\delta\mathbf{v}_j^\top\nabla\mathcal{M}_k(\mathbf{v}_j)\}$ must decay to zero. This, together with Lemma 3.4.3, implies that $\lim_{j\to\infty}\nabla\mathcal{M}_k(\mathbf{v}_j) = \mathbf{0}$. This gives also $\lim_{j\to\infty}\mathbf{r}_k(\mathbf{v}_j) = \mathbf{0}$, hence proving $\lim_{j\to\infty}\mathbf{v}_j = \mathbf{v}_k^\star$, since $\mathbf{r}_k(\mathbf{v}) = \mathbf{0}$ admits a unique solution (cf. §3.4.1). □

Lemma 3.5.3. *Let any $\mathbf{v}_k \in \mathbb{R}^\ell$ be given and $\{\mathbf{v}_j\}$ be the sequence generated by Algorithm 5. Then, the sequence $\{\delta\mathbf{v}_j\}$ converges to zero.*

Proof. Vector $\delta\mathbf{v}_j$ is the unique solution to the linear system $\boldsymbol{M}_k(\mathbf{v}_j)\delta\mathbf{v} = -\nabla\mathcal{M}_k(\mathbf{v}_j)$; cf. Lemma 3.4.3. By Theorem 3.5.2, it is $\lim_{j\to\infty}\nabla\mathcal{M}_k(\mathbf{v}_j) = \mathbf{0}$. Then, it follows that also $\lim_{j\to\infty}\delta\mathbf{v}_j = \mathbf{0}$. □

We can prove that the unit stepsize is eventually taken, namely it satisfies the Armijo's sufficient decrease condition.

Lemma 3.5.4. *Let any $\mathbf{v}_k \in \mathbb{R}^\ell$ and $\eta \in (0, 1/2)$ be given, and the sequence $\{\mathbf{v}_j\}$ be generated by Algorithm 5. Then, for j sufficiently large, the unit step size $\tau_j = 1$ satisfies the sufficient decrease condition (3.4.18).*

Proof. Since $\mathcal{M}_k$ is piecewise quadratic and strictly convex, we have that

$$\begin{aligned}\mathcal{M}_k(\mathbf{v}_j+\delta\mathbf{v}_j)-\mathcal{M}_k(\mathbf{v}_j) &\le \delta\mathbf{v}_j^\top\nabla\mathcal{M}_k(\mathbf{v}_j)+\frac{1}{2}\delta\mathbf{v}_j^\top\boldsymbol{M}_k(\mathbf{v}_j)\delta\mathbf{v}_j+\frac{\Omega_k}{6}\|\delta\mathbf{v}_j\|^3\\ &= -\frac{1}{2}\delta\mathbf{v}_j^\top\boldsymbol{M}_k(\mathbf{v}_j)\delta\mathbf{v}_j+\frac{\Omega_k}{6}\|\delta\mathbf{v}_j\|^3\\ &\le -\frac{1}{2}\delta\mathbf{v}_j^\top\boldsymbol{M}_k(\mathbf{v}_j)\delta\mathbf{v}_j+\frac{\Omega_k}{6}\|\delta\mathbf{v}_j\|\frac{\delta\mathbf{v}_j^\top\boldsymbol{M}_k(\mathbf{v}_j)\delta\mathbf{v}_j}{\omega_k}\\ &= \left(\frac{1}{2}-\frac{\Omega_k}{6\omega_k}\|\delta\mathbf{v}_j\|\right)\delta\mathbf{v}_j^\top\nabla\mathcal{M}_k(\mathbf{v}_j)\end{aligned}$$

for some finite $\Omega_k \ge 0$ and with $\omega_k > 0$ defined in Lemma 3.4.1. The second and fourth lines use that $\boldsymbol{M}_k(\mathbf{v}_j)\delta\mathbf{v}_j = -\nabla\mathcal{M}_k(\mathbf{v}_j)$, by (3.4.8) and (3.4.11). Since $\lim_{j\to\infty}\delta\mathbf{v}_j = \mathbf{0}$ by Lemma 3.5.3, for j sufficiently large it holds

$$\mathcal{M}_k(\mathbf{v}_j+\delta\mathbf{v}_j)-\mathcal{M}_k(\mathbf{v}_j) \le \eta\delta\mathbf{v}_j^\top\nabla\mathcal{M}_k(\mathbf{v}_j)$$

for any given $\eta \in (0,1/2)$. Hence, the sufficient decrease condition (3.4.18) is eventually satisfied by the unit stepsize $\tau_j = 1$. □

The following local quadratic convergence result follows a general theorem in [56]; see also [57, 69, 76, 65].

Lemma 3.5.5. *Let any $\mathbf{v}_k \in \mathbb{R}^\ell$ be given and the sequence $\{\mathbf{v}_j\}$ be generated by Algorithm 5. If $\mathbf{v}_j$ is sufficiently close to $\mathbf{v}_k^\star$ and a full step is taken, namely $\tau_j = 1$, then it holds*

$$\left\|\mathbf{v}_{j+1}-\mathbf{v}_k^\star\right\| = O\left(\left\|\mathbf{v}_j-\mathbf{v}_k^\star\right\|^2\right). \tag{3.5.2}$$

Proof. If a full step is accepted, it is $\mathbf{v}_{j+1} = \mathbf{v}_j + \delta\mathbf{v}_j$ and then

$$\begin{aligned}\|\mathbf{v}_{j+1}-\mathbf{v}_k^\star\| &= \|\mathbf{v}_j+\delta\mathbf{v}_j-\mathbf{v}_k^\star\|\\ &= \|\mathbf{v}_j-\boldsymbol{M}_k^{-1}(\mathbf{v}_j)\nabla\mathcal{M}_k(\mathbf{v}_j)-\mathbf{v}_k^\star\|\\ &\le \|\boldsymbol{M}_k^{-1}(\mathbf{v}_j)\|\|\boldsymbol{M}_k(\mathbf{v}_j)\left(\mathbf{v}_j-\mathbf{v}_k^\star\right)-\nabla\mathcal{M}_k(\mathbf{v}_j)\|\\ &= \|\boldsymbol{M}_k^{-1}(\mathbf{v}_j)\|\|\boldsymbol{M}_k(\mathbf{v}_j)\left(\mathbf{v}_j-\mathbf{v}_k^\star\right)-\nabla\mathcal{M}_k(\mathbf{v}_j)+\nabla\mathcal{M}_k(\mathbf{v}_k^\star)\|.\end{aligned}$$

Here, the second line follows from $\boldsymbol{M}_k(\mathbf{v}_j)\delta\mathbf{v}_j = -\nabla\mathcal{M}_k(\mathbf{v}_j)$; the third relies on the existence of a uniform upper bound to $\|\boldsymbol{M}_k^{-1}(\cdot)\|$ by Lemma 3.4.1; the fourth line uses the fact that $\nabla\mathcal{M}_k(\mathbf{v}_k^\star) = \mathbf{0}$. For $\mathbf{v}_j$ sufficiently close to $\mathbf{v}_k^\star$, we have that

$$\left\|\boldsymbol{M}_k(\mathbf{v}_j)\left(\mathbf{v}_j-\mathbf{v}_k^\star\right)+\nabla\mathcal{M}_k(\mathbf{v}_k^\star)-\nabla\mathcal{M}_k(\mathbf{v}_j)\right\| = O\left(\left\|\mathbf{v}_j-\mathbf{v}_k^\star\right\|^2\right),$$

by the strong semismoothness of $\nabla\mathcal{M}_k$ [57, 71, 167]. Combining with the previous expression, this establishes the result. □

Theorem 3.5.6. *Let any $\mathbf{v}_k \in \mathbb{R}^\ell$ be given and the sequence $\{\mathbf{v}_j\}$ be generated by Algorithm 5. Then, the asymptotic rate of convergence is quadratic,* i.e., *for j sufficiently large condition (3.5.2) holds.*

Proof. Theorem 3.5.2 guarantees that $\mathbf{v}_j \to \mathbf{v}_k^\star$ as $j \to \infty$, and then Lemma 3.5.4 shows that full steps are eventually accepted. Finally, Lemma 3.5.5 establishes the local quadratic convergence rate. □

3.5.2 Outer loop

Let us consider now the outer loop, sketched in Algorithm 4. This consists of inexact proximal point iterations [30], hence global and local convergence properties of the outer loop can be derived based on [43, Thm 2.1]. Recall that, by construction, the regularization parameters are non-increasing, positive, and bounded away from zero. Also, by $\epsilon_0 \in \mathbb{R}_+$ and $\kappa_\epsilon \in [0, 1)$, the sequence $\{\epsilon_k\} \subseteq \mathbb{R}_+$ is summable, since

$$\sum_{k \in \mathbb{N}} \epsilon_k = \sum_{k \in \mathbb{N}} \kappa_\epsilon^k \epsilon_0 = \frac{\epsilon_0}{1 - \kappa_\epsilon} < +\infty.$$

The following result shows that criterion (A_r) from [43] is satisfied.

Lemma 3.5.7. *Suppose $\mathcal{T}^{-1}(\mathbf{0})$ is nonempty. Let any $\mathbf{v}_0 \in \mathbb{R}^\ell$ be given, and the sequence $\{\mathbf{v}_k\}$ be generated by Algorithm 4. Then, there exists a summable sequence $\{e_k\} \subseteq \mathbb{R}_+$ such that*

$$\|\mathbf{v}_{k+1} - \mathbf{v}_k^\star\| \leq e_k \quad \forall k.$$

Proof. By the inner stopping condition, for all $k \in \mathbb{N}$ it holds $\|\mathbf{r}_k(\mathbf{v}_{k+1})\| \leq \epsilon_k$, with summable $\{\epsilon_k\} \subseteq \mathbb{R}_+$. Morever, for any given $k \in \mathbb{N}$, we have that, for some $\tilde{\eta}_k > 0$, it is

$$\tilde{\eta}_k \|\mathbf{v} - \mathbf{v}_k^\star\| \leq \|\nabla \mathcal{M}_k(\mathbf{v}) - \nabla \mathcal{M}_k(\mathbf{v}_k^\star)\| = \|\nabla \mathcal{M}_k(\mathbf{v})\| = \|\boldsymbol{T}_k \mathbf{r}_k(\mathbf{v})\|$$

for all $\mathbf{v} \in \mathbb{R}^\ell$, since $\mathcal{M}_k$ is Σ_k-strongly convex. By the boundedness of $\boldsymbol{T}_k$, there exists a constant $\eta > 0$ such that the bound $\|\mathbf{v} - \mathbf{v}_k^\star\| \leq \eta \|\mathbf{r}_k(\mathbf{v})\|$ holds for all $k \in \mathbb{N}$ and $\mathbf{v} \in \mathbb{R}^\ell$. Thus, in particular, for all $k \in \mathbb{N}$ it is

$$\|\mathbf{v}_{k+1} - \mathbf{v}_k^\star\| \leq \eta \|\mathbf{r}_k(\mathbf{v}_{k+1})\| \leq \eta \epsilon_k.$$

Let $e_k := \eta \epsilon_k$, and the proof is complete. □

Notice that we choose $r = 0$ in (A_r) for the sake of simplicity, although this may prevent faster convergence [43]. Then, since problem (3.3.5) is a polyhedral variational inequality [123, §3D], we can invoke [43, Prop. 1.2].

Theorem 3.5.8. *Suppose $\mathcal{T}^{-1}(\mathbf{0})$ is nonempty. Let any $\mathbf{v}_0 \in \mathbb{R}^\ell$ be given, and the sequence $\{\mathbf{v}_k\}$ be generated by Algorithm 4. Then, the sequence $\{\mathbf{v}_k\}$ is well-defined and converges to a solution $\mathbf{v}^\star \in \mathcal{T}^{-1}(\mathbf{0})$.*

3.6 Relationship with Similar Methods

Our approach is inspired by and shares many features with other recently developed methods. This section elaborates upon their relationship with `QPDO`.

`FBstab`, or proximally stabilized Fischer-Burmeister method, *synergistically combines the proximal point algorithm with a primal-dual semismooth Newton-type method* to solve convex QPs [243]. It takes itself apart from the other methods considered here, because it adopts the (penalized) Fischer-Burmeister (FB) function [53, 82]. Conversely, it is perhaps the most similar in spirit to `QPDO`. Considering both the FB function and the minimum function solely as NCP functions [57, 56], the two methods essentially match with each other. Nevertheless, they differ on some, possibly significant, details. While `QPDO` relies upon the primal-dual PAL function $\mathcal{M}$ as a merit function, `FBstab` adopts the squared norm of the inner residual to get a descent direction, on the vein of [53, 82, 121]. Adopting the FB function, `FBstab` may enjoy its nice regularity properties and ease of globalization, at the cost of introducing some nonlinearity; cf. [96]. On the other hand, `QPDO` builds upon the piecewise affine nature of the minimum function, which provides no additional nonlinearity besides its nondifferentiability [223]. Thus, it can exploit factorization updates, perform exact line search by solving a piecewise linear equation, and handle simultaneously bilateral constraints in a natural manner.

QPALM is a *proximal augmented Lagrangian based solver for convex quadratic programs* [224]. Recent advancements [239] allow to handle nonconvex QPs by adjusting the primal regularization term. Given a primal-dual estimate $\overline{\mathbf{v}}$, the exact, unique resolvent update $\mathbf{v}^\triangle$ of QPALM [224, Eq. 6], with $\Sigma = \text{blkdiag}(\sigma^{-1}\boldsymbol{I}, \mu^{-1}\boldsymbol{I})$, is given by

$$\mathbf{x}^\triangle = \arg\min_{\mathbf{x}\in\mathbb{R}^n} \ \varphi(\mathbf{x}), \tag{3.6.1a}$$
$$\mathbf{y}^\triangle = \overline{\mathbf{y}} + \mu^{-1}\left[\boldsymbol{A}\mathbf{x}^\triangle - \Pi_C\left(\boldsymbol{A}\mathbf{x}^\triangle + \mu\overline{\mathbf{y}}\right)\right]. \tag{3.6.1b}$$

Herein, function φ is given by [224, Eq. 8]

$$\varphi(\mathbf{x}) := f(\mathbf{x}) + \frac{1}{2\mu}\operatorname{dist}_C^2\left(\boldsymbol{A}\mathbf{x} + \mu\overline{\mathbf{y}}\right) + \frac{\sigma}{2}\|\mathbf{x} - \overline{\mathbf{x}}\|^2$$

and closely resembles $\mathcal{M}_k$ in (3.4.6); it is continuously differentiable and its gradient reads

$$\nabla\varphi(\mathbf{x}) = \nabla f(\mathbf{x}) + \boldsymbol{A}^\top\overline{\mathbf{y}} + \sigma(\mathbf{x} - \overline{\mathbf{x}}) + \frac{1}{\mu}\boldsymbol{A}^\top\left[\boldsymbol{A}\mathbf{x} - \Pi_C(\boldsymbol{A}\mathbf{x} + \mu\overline{\mathbf{y}})\right].$$

Since (3.6.1a) yields $\nabla\varphi(\mathbf{x}^\triangle) = \mathbf{0}$, combining with (3.6.1b) and rearranging yield necessary and sufficient conditions for the unique update $(\mathbf{x}^\triangle, \mathbf{y}^\triangle)$:

$$\mathbf{0} = \boldsymbol{Q}\mathbf{x} + \mathbf{q} + \boldsymbol{A}^\top\mathbf{y} + \sigma\left(\mathbf{x} - \overline{\mathbf{x}}\right), \tag{3.6.2a}$$
$$\mathbf{0} = \boldsymbol{A}\mathbf{x} + \mu\left(\overline{\mathbf{y}} - \mathbf{y}\right) - \Pi_C\left(\boldsymbol{A}\mathbf{x} + \mu\overline{\mathbf{y}}\right). \tag{3.6.2b}$$

Conditions (3.6.2) and (3.4.5) are remarkably similar and differ only in the argument of the projection Π_C: the term $-\mu\mathbf{y}/2$ is missing in (3.6.2b), since φ derives from the (primal) proximal augmented Lagrangian function; see [224, Remark 2]. This underlines the primal-dual nature of QPDO, that may better cope with changes in the active set [92] and control the quality of both primal and dual variables during iterations [158, 190], without any additional computational effort.

OSQP is a *solver for convex quadratic programs based on the alternating direction method of multipliers* [253]. Rearranging from [253, Alg. 1], with parameters $\alpha := 1$, $\rho := \mu^{-1}$, primal-dual estimate $(\overline{\mathbf{x}}, \overline{\mathbf{y}})$, and constraint estimate $\overline{\mathbf{z}} \approx \boldsymbol{A}\overline{\mathbf{x}}$, the primal-auxiliary update $(\mathbf{x}^\diamond, \mathbf{s}^\diamond)$ is the unique solution to the linear system

$$\mathbf{0} = \boldsymbol{Q}\mathbf{x} + \mathbf{q} + \boldsymbol{A}^\top\mathbf{s} + \sigma(\mathbf{x} - \overline{\mathbf{x}}), \tag{3.6.3a}$$
$$\mathbf{0} = \boldsymbol{A}\mathbf{x} + \mu(\overline{\mathbf{y}} - \mathbf{s}) - \overline{\mathbf{z}}. \tag{3.6.3b}$$

Then, the constraint update and the dual update are given by $\mathbf{z}^\diamond = \Pi_S\left(\overline{\mathbf{z}} + \mu\mathbf{s}^\diamond\right)$ and $\mathbf{y}^\diamond = \mathbf{s}^\diamond + \mu^{-1}\left(\overline{\mathbf{z}} - \mathbf{z}^\diamond\right)$, respectively. Conditions (3.6.3) closely resemble (3.4.5). However, an auxiliary variable $\mathbf{s}$ replaces the dual variable $\mathbf{y}$, and the estimate $\overline{\mathbf{z}}$ substitutes the projection in (3.4.5). This makes subproblem (3.6.3) a linear system, but leads to a first-order method, which often requires many iterations. In [253], the Authors propose an update rule to adapt the values of μ and heuristically enhance the convergence speed.

2ndMM is a *second order primal-dual algorithm for nonsmooth convex composite* problems [194], namely for the minimization of $f(\mathbf{x}) + g(\boldsymbol{A}\mathbf{x})$ with respect to $\mathbf{x}$, with function f smooth and strictly convex, function g proper, lower semicontinuous, and convex, and matrix $\boldsymbol{A}$ with full row rank. Since 2ndMM considers a broader class of problems, it does not exploit the peculiar structure of QPs. Nevertheless, it closely relates to QPDO in that it adopts both the proximal augmented Lagrangian approach and the primal-dual augmented Lagrangian function, to compute a search direction and as a merit function, respectively. However, these are not combined nor tightly intertwined as in QPDO.

QPNNLS-PROX is an algorithm for solving convex QPs using nonnegative least squares within an outer proximal-point iteration scheme [208]. Although similar, this method differs from QPDO in various aspects. It applies only a primal proximal regularization, and the resulting strictly convex

subproblem is reformulated as a partially nonnegative least squares problem. This can be efficiently solved using a tailored, numerically robust, active-set method.

IP-PMM is a primal-dual regularized interior-point method for convex quadratic programming [250]. It inexactly solves the subproblems of the proximal method of multipliers (PMM) via an infeasible interior-point (IP) method. Under standard assumptions, the algorithm exhibits polynomial complexity [250], and numerical results have demonstrated that the regularization improves the reliability of the underlying IP method.

We close this section with some general comments on modern QP solvers, briefly discussed here. Remarkably, these share many aspects and, we believe, this is not surprising. In fact, solutions to convex QPs are fully characterized by their KKT conditions. Hence, numerical methods for QP essentially solve them, since they are both necessary and sufficient. In particular, we argue, QP solvers differ in the manner they deal with the piecewise affine structure of these conditions; cf. (3.3.3). Indeed, the logarithmic barrier in IP methods and the NCP functions in Newton-type methods treat the inequalities from different perspectives. Furthermore, due to its regularization effect, many methods adopt an outer proximal layer as a mechanism to gain robustness, handle non-strictly convex problems, and manage degeneracy.

3.7 Implementation Details

We implemented QPDO in open-source C code with an interface to MATLAB™ [254]; see §0.1. Our implementation can handle any QP formulated as (3.1.1), without any requirement about the problem data other than Assumption 3.1.1. This section discusses some relevant aspects of the program, such as the linear solver, parameters update rules, infeasibility detection, and problem scaling.

3.7.1 Linear solver

The linear system (3.4.15) is solved with CHOLMOD [117], a direct sparse solver based on a supernodal Cholesky factorization. This linear solver is analogous to the one adopted in QPALM [224], for the sake of comparison. This software package can apply multi-rank factorization updates, but only for linear systems with symmetric positive definite coefficient matrix. This can be obtained from (3.4.15) via condensing, or reduction procedure. Let $\left(\mathbf{r}_k^{\text{dual}}, \mathbf{r}_k^{\text{prim}}\right)$ partition the inner residual $\mathbf{r}_k$ in (3.4.5). Then, formally solving for $\delta\mathbf{y}$ in (3.4.15), we obtain the expression (omitting subscripts and arguments)

$$\begin{aligned}\delta\mathbf{y} &= \mu^{-1}(\boldsymbol{I} - \boldsymbol{P}/2)^{-1}\left[(\boldsymbol{I} - \boldsymbol{P})\boldsymbol{A}\delta\mathbf{x} + \mathbf{r}^{\text{prim}}\right] \\ &= \mu^{-1}(\boldsymbol{I} + \boldsymbol{P})\left[(\boldsymbol{I} - \boldsymbol{P})\boldsymbol{A}\delta\mathbf{x} + \mathbf{r}^{\text{prim}}\right] \\ &= \mu^{-1}(\boldsymbol{I} - \boldsymbol{P})\boldsymbol{A}\delta\mathbf{x} + \mu^{-1}(\boldsymbol{I} + \boldsymbol{P})\mathbf{r}^{\text{prim}},\end{aligned}$$

where the second and third lines are due to the binary structure of $\boldsymbol{P}$ (3.4.13). Substituting $\delta\mathbf{y}$ and rearranging, we obtain a linear system for $\delta\mathbf{x}$:

$$\left[\boldsymbol{Q} + \sigma\boldsymbol{I} + \mu^{-1}\boldsymbol{A}^\top(\boldsymbol{I} - \boldsymbol{P})\boldsymbol{A}\right]\delta\mathbf{x} = \boldsymbol{A}^\top\boldsymbol{P}\mathbf{y} - \mu^{-1}\boldsymbol{A}^\top(\boldsymbol{I} - \boldsymbol{P})\mathbf{r}^{\text{prim}} - \mathbf{r}^{\text{dual}}.$$

This has a symmetric, positive definite coefficient matrix and can be solved by CHOLMOD. On the one hand, this approach allows multi-rank factorization updates [84], thus avoiding the need for a full re-factorization at every inner iteration. On the other hand, sparsity pattern may be lost and significant fill-in may arise due to the matrix-matrix product $\boldsymbol{A}^\top\boldsymbol{A}$. For this reason and to fully exploit the problem sparsity, the current implementation may benefit from solving (3.4.15) via sparse symmetric linear solvers, possibly based on the $LDL^\top$ factorization [61, 101], with (multi-rank) factorization updates; cf. §3.4.2.

3.7.2 Parameters selection

Solving convex QPs via the proximal point algorithm imposes mild restrictions on the sequence of primal-dual regularization parameters $\{\Sigma_k\}$. As mentioned in §3.3.2, there are no additional requirements other than being non-increasing and positive definite. However, similarly to forcing sequences in augmented Lagrangian methods [49], the sequence of regularization parameters greatly affects the behaviour of QPDO, and a careful tuning can positively impact the performance. For instance, although faster convergence rates can be expected if $\Sigma_k \to \mathbf{0}$ [43], numerical stability and machine precision should be taken into account. Following [239, §5.3] and [253, §5.2], our implementation considers only diagonal matrices of the form $\Sigma_k = \text{blkdiag}(\sigma_k \boldsymbol{I}, \text{diag}(\boldsymbol{\mu}_k))$. We refer to the effect of σ_k and $\boldsymbol{\mu}_k$ as primal and dual regularization, respectively.

Dual regularization The dual regularization parameter $\boldsymbol{\mu}_k$ proves critical for the practical performance of the method. We argue, it has such impact since it strikes the balance between the number of inner and outer iterations, seeking easy-to-solve subproblems, effective warm starting, or rapid constraints satisfaction. After all, suitable forcing sequences are crucial in all augmented Lagrangian methods. Therefore, we carefully initialize and update the value of $\boldsymbol{\mu}_k$, guided by the interpretation as a constraint penalization offered by the augmented Lagrangian framework; cf. §3.4.1. In our implementation, we consider a vector $\boldsymbol{\mu}_k$ to gain a finer control over the constraint penalization [49]. Given a (primal) initial guess $\mathbf{x}_0 \in \mathbb{R}^n$, we initialize as in [163, §12.4]:

$$\begin{aligned} \mathbf{d}_0 &:= \boldsymbol{A}\mathbf{x}_0 - \Pi_C(\boldsymbol{A}\mathbf{x}_0), \\ \boldsymbol{\mu}_0^i &:= \Pi_{[\mu_0^{\min}, \mu_0^{\max}]}\left(\kappa_\mu \frac{\max(1, (\mathbf{d}_0^i)^2/2)}{\max(1, |f(\mathbf{x}_0)|)}\right), \ i \in [1; m], \end{aligned}$$

where $\mu_0^{\max} \geq \mu_0^{\min} > 0$ and $\kappa_\mu \geq 0$. Then, following [239, §5.3], we monitor the primal residual $\mathbf{r}_{\text{prim}}(\mathbf{v}) := \boldsymbol{A}\mathbf{x} - \Pi_C(\boldsymbol{A}\mathbf{x} + \mathbf{y})$ from (3.3.3) and update the dual regularization parameter $\boldsymbol{\mu}_k$ accordingly. If $|\mathbf{r}_{\text{prim}}^i(\mathbf{v}_{k+1})| > \max\left(\theta_\mu |\mathbf{r}_{\text{prim}}^i(\mathbf{v}_k)|, \epsilon_{\text{opt}}\right)$, we set

$$\boldsymbol{\mu}_{k+1}^i = \Pi_{[\mu_{\min}, \boldsymbol{\mu}_k^i]}\left(\delta_\mu \frac{\|\mathbf{r}_{\text{prim}}(\mathbf{v}_{k+1})\|_\infty}{|\mathbf{r}_{\text{prim}}^i(\mathbf{v}_{k+1})|} \boldsymbol{\mu}_k^i\right),$$

where $\theta_\mu \in (0, 1)$, $\mu_{\min} > 0$, and $\delta_\mu \geq 0$. Otherwise, we set $\boldsymbol{\mu}_{k+1}^i = \boldsymbol{\mu}_k^i$. These rules adapt the constraint penalization on the current residual, seeking a uniform, steady progression towards feasibility, while making sure the sequences $\{\boldsymbol{\mu}_k^i\}$, $i \in [1; m]$, are non-increasing and bounded away from zero. In our implementation, the default values are $\mu_0^{\min} = 10^{-4}$, $\mu_0^{\max} = 10^4$, $\kappa_\mu = 0.1$, $\mu_{\min} = 10^{-8}$, $\delta_\mu = 10^{-2}$ and $\theta_\mu = 0.1$.

Remark 3.7.1. Owing to (3.4.13), the dual regularization parameter μ affects the identification of the active set, which is far from being a trivial or negligible task; see [74]. It is interesting to notice the interpretation given in [92, Rem. 3.4] of the primal-dual active-set strategy or, equivalently, semismooth Newton's method, as a prediction strategy for the true active and inactive sets. On the one hand, small values for μ lead to strong constraint penalization and the active set of the problem solution could be readily identified. On the other hand, however, the value of μ should be sufficiently large to act as a regularization term and to avoid numerical difficulties in (3.4.15). Considering a vector $\boldsymbol{\mu}$ in place of a scalar μ might mitigate this issue, allowing a finer tuning of these trade-offs for each constraint.

Primal regularization The primal regularization term turns out to be less crucial with respect to the dual counterpart. For this reason, it is associated to a scalar value and tuned independently from the residual. Starting from $\sigma_0 > 0$, we apply

$$\sigma_{k+1} = \max(\sigma_{\min}, \kappa_\sigma \sigma_k),$$

where $\sigma_{\min} > 0$ and $\kappa_\sigma \in [0, 1]$. In our implementation the default values are $\sigma_0 = 0.1$, $\sigma_{\min} = 10^{-7}$, and $\kappa_\sigma = 0.1$.

Early termination The inner tolerance ϵ_k also affects the performance of QPDO, since it balances subproblem accuracy and early termination. In Algorithm 4, these aspects relate to the parameters ϵ_0 and κ_ϵ, which drive $\{\epsilon_k\}$ to zero. However, finite precision should also be taken into account. In fact, although the semismooth Newton's method converges in finitely many iterations, the solution provided is exact up to round-off errors and numerical precision. Therefore, we deviate from Algorithm 4 in this respect and employ the update rule

$$\epsilon_{k+1} = \max(\epsilon_{\min}, \kappa_\epsilon \epsilon_k),$$

where $0 \leq \epsilon_{\min} \leq \epsilon_{\text{opt}}$. In our implementation, the default values are $\epsilon_0 = 1$, $\kappa_\epsilon = 0.1$, $\epsilon_{\min} = 10^{-14}$, and $\epsilon_{\text{opt}} = 10^{-6}$.

3.7.3 Infeasibility detection

A routine for detecting primal and dual infeasibility of problem (3.1.1) is included in Algorithm 4. This allows the algorithm to terminate with either a primal-dual solution or a certificate of primal or dual infeasibility, for some given tolerances. We adopt the mechanism developed in [218, §5.2], which holds whenever the proximal point algorithm is employed to solve the KKT conditions (3.3.2). Problem (3.1.1) is declared primal or dual infeasible based on some conditions on $\Delta \mathbf{x}_k := \mathbf{x}_{k+1} - \mathbf{x}_k$ and $\Delta \mathbf{y}_k := \mathbf{y}_{k+1} - \mathbf{y}_k$, $k \geq 0$. The reader may refer to [253, §3.4], [224, §V.C], and [243, §4.1], and [250, §4] for analogous applications.

3.7.4 Preconditioning

Although the Newton's direction and the exact linesearch stepsize are invariant to scaling of the variables, this may affect the behaviour of the overall optimization algorithm; see [98]. Preconditioning, or scaling, the problem may alleviate ill-conditioning and mitigate numerical issues, especially when the problem data span across many orders of magnitude. Automatic scaling, or equilibration, of optimization problems is an active field of research, spanning from linear systems to nonlinear programming [44, 60, 66, 130]. In our implementation, we closely follow [239, §5.2] and scale the problem data by performing the Ruiz's equilibration procedure [86] on the constraint matrix $\boldsymbol{A}$. This procedure iteratively scales the rows and columns of a matrix in order to make their infinite norms approach one. By default, QPDO performs 10 scaling iterations. Slightly different routines are adopted, *e.g.*, in [253, §5.1] and [250, §5.1.2]. Note that, by default, if the problem is initially scaled, the termination conditions for both, optimality and infeasibility, refer to the original, unscaled problem.

3.8 Numerical Results

We discuss details of our open-source implementation of QPDO and present computational results on random problems and the Maros–Mészáros set [79]. We test and compare QPDO against the open-source, full-fledged solvers OSQP [253] and QPALM [224, 239]. Although our current implementation proves competitive with more mature solvers, we plan to improve and further extend it. The interested reader may refer to [208, 243, 253, 239] for more extensive numerical evaluations.

Setup We consider the tolerance $\epsilon_{\text{opt}} = 10^{-5}$, and set the tolerances in OSQP and QPALM to $\epsilon_{\text{abs}} = \epsilon_{\text{opt}}$ and $\epsilon_{\text{rel}} = 0$. In addition, we set the maximum number of iterations and the time limit to 10^{12} and 100 s, respectively, for every solver, and we leave all the other settings to the internal defaults. It is worth mentioning that, since no initial guess is provided, all the solvers start with $\mathbf{v}_0 = \mathbf{0}$. We deem optimal a primal-dual pair $\mathbf{v}^\star = (\mathbf{x}^\star, \mathbf{y}^\star)$ returned by a solver if it satisfies the conditions

$$\|\boldsymbol{Q}\mathbf{x}^\star + \mathbf{q} + \boldsymbol{A}^\top \mathbf{y}^\star\|_\infty \leq \epsilon_{\text{opt}}, \qquad \text{and} \qquad \|\boldsymbol{A}\mathbf{x}^\star - \Pi_C(\boldsymbol{A}\mathbf{x}^\star + \mathbf{y}^\star)\|_\infty \leq \epsilon_{\text{opt}},$$

otherwise we consider it a failure. All the experiments were carried out on a desktop running Ubuntu 16.04 with Intel Core i7-8700 and 16 GB RAM. The shifted geometric mean (sgm) of the run times, the performance profiles, and the time profiles are used to evaluate and compare the solvers on a test set, as detailed in §1.6.2.

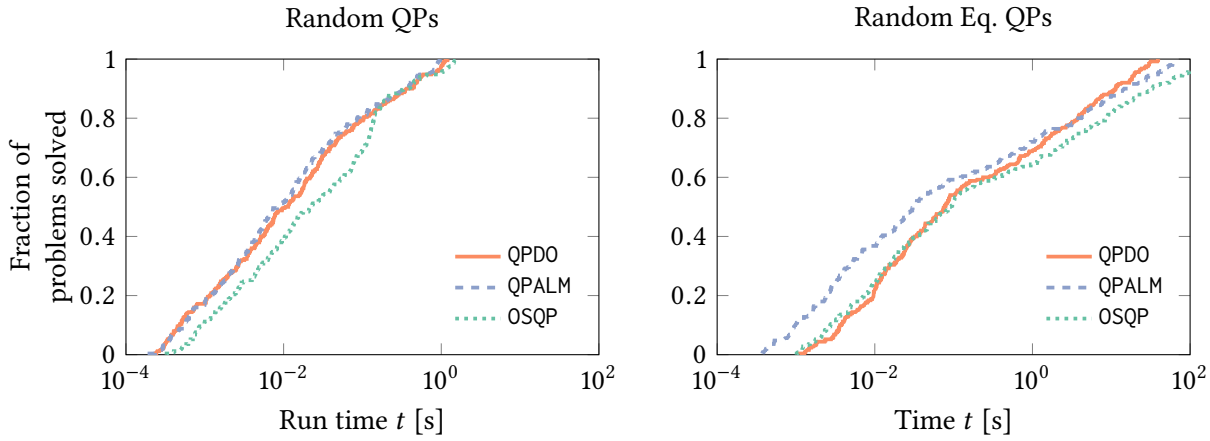


Figure 3.1: Comparison on random problems with data profiles: fraction of problems solved by each solver as a function of run time.

Table 3.1: Comparison on different problem classes with run time and failure rate.

			QPDO	QPALM	OSQP
Random QPs	Timing (sgm)	[s]	0.090	0.079	0.112
	Failure rate	[%]	0.00	0.00	0.00
Random Eq. QPs	Timing (sgm)	[s]	0.988	1.045	1.648
	Failure rate	[%]	0.00	0.00	4.00
Maros–Mészáros	Timing (sgm)	[s]	0.061	0.435	3.141
	Failure rate	[%]	0.00	6.85	24.66

3.8.1 Random problems

We considered QPs in the form of (3.1.1) with randomly generated problem data. In each problem instance, the number of variables is $n = \lceil 10^a \rceil$ and ranges between 10^1 and 10^3, with a uniformly distributed, *i.e.*, $a \sim \mathcal{U}(1, 3)$. The number of constraints is $m = 10\,n$. The linear cost is normally distributed, *i.e.*, $\mathbf{q}_i \sim \mathcal{N}(0, 1)$. The cost matrix is $\boldsymbol{Q} = \boldsymbol{P}\boldsymbol{P}^\top$, where $\boldsymbol{P} \in \mathbb{R}^{n \times n}$ has 10% nonzero entries $\boldsymbol{P}_{ij} \sim \mathcal{N}(0, 1)$. The constraint matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ contains 10% nonzero entries $\boldsymbol{A}_{ij} \sim \mathcal{N}(0, 1)$. The bounds are uniformly distributed, *i.e.*, $\mathbf{l}_i \sim \mathcal{U}(-1, 0)$ and $\mathbf{u}_i \sim \mathcal{U}(0, 1)$. We also investigated equality-constrained QPs. For these problems, n ranges between 10^2 and 10^4, $m = \lceil n/10 \rceil$, and $\mathbf{l}_i = \mathbf{u}_i \sim \mathcal{N}(0, 1)$. We generated 250 instances from each problem class.

Results Computational results are summarized in Table 3.1 and Figures 3.1 and 3.2. QPDO and QPALM succeeded in all the problem instances, whereas OSQP reached the time limit 10 times. Performance profiles suggest that, for both problem classes, QPALM exhibits the best performance, with QPDO slightly behind and OSQP third. However, the time profiles in Figure 3.1 show that, on equality-constrained QPs, QPDO scales better than the other solvers. Indeed, QPDO is the first to complete the test set of random problems. OSQP reaches the time limit on few problems, due to the relative high accuracy requirement. Overall, all solvers prove competitive.

3.8.2 Maros–Mészáros problems

We considered the Maros–Mészáros test set [79] of hard QPs. This test set is often used to benchmark convex QP solvers, as it includes many large-scale and ill-conditioned problems. Selecting those with $n \leq 10^3$ yields 73 problems, with $2 \leq n \leq 1000$, $3 \leq m \leq 1750$, and $6 \leq \text{nnz}(\boldsymbol{Q}) + \text{nnz}(\boldsymbol{A}) \leq 22292$.

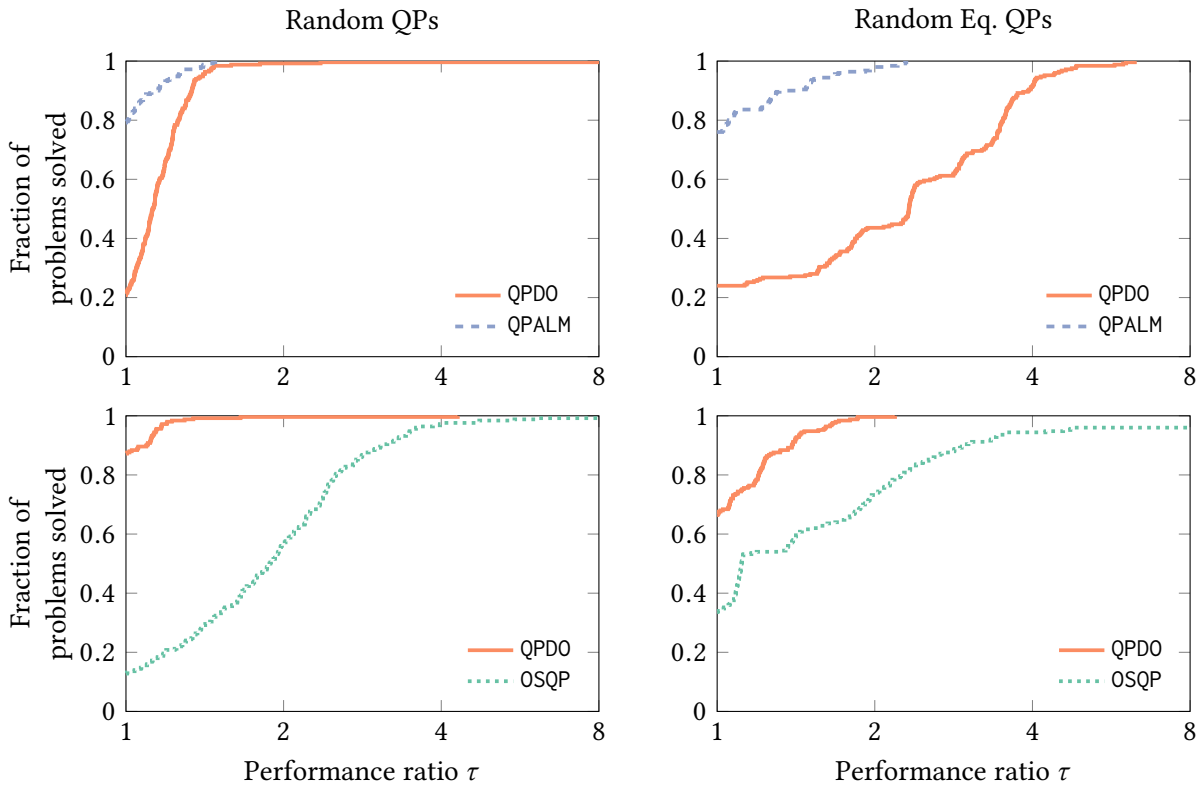


Figure 3.2: Comparison on random problems with performance profiles: fraction of problems solved by each solver as a function of performance ratio.

Results Computational results are summarized in Table 3.1 and Figures 3.3 and 3.4. On this test set, QPDO demonstrates its robustness, succeeding with all 73 problems. OSQP reaches the maximum number of iterations in 18 cases, while QPALM fails 5 times. OSQP is very fast for some problems but has a comparatively high failure rate. As a first-order method, OSQP may take many, yet computationally cheap, iterations to cope with ill-conditioning and the relatively high accuracy requirements. QPALM fails on some problems, presumably due to linear algebra issues; its relatively high timing in Table 3.1 is associated to the failure rate. Considering only the problems it solved, QPALM's timing (sgm) is 0.050 s, whereas QPDO takes 0.054 s. Overall, the reliable performance of QPDO appears effective and, indeed, promising.

3.9 Summary

We presented a primal-dual Newton-type proximal method for convex quadratic programs. This builds upon a simple yet crucial result: a suitable merit function for the proximal subproblem is found in the proximal primal-dual augmented Lagrangian function. This allows us to effectively weave the proximal point method together with semismooth Newton's, yielding structured symmetric linear systems, exact linesearch, and the possibility to apply sparse multi-rank factorization updates. Our method requires the solution of symmetric quasi-definite linear systems, that are always solvable, imposing no assumptions on the problem data other than convexity. The method is simple and easily warm started, can exploit sparsity pattern, handle degeneracy, and detect infeasibility. We have implemented our method QPDO in a general-purpose solver, written in open-source C code. Our solver can take advantage of arbitrary initial guesses and can provide accurate solutions. We benchmarked it against state-of-the-art QP solvers, comparing run times and failure rates. Numerical tests on randomly generated problems and the Maros-Mészáros test set demonstrated promising results. QPDO proved reliable, effective, and competitive.

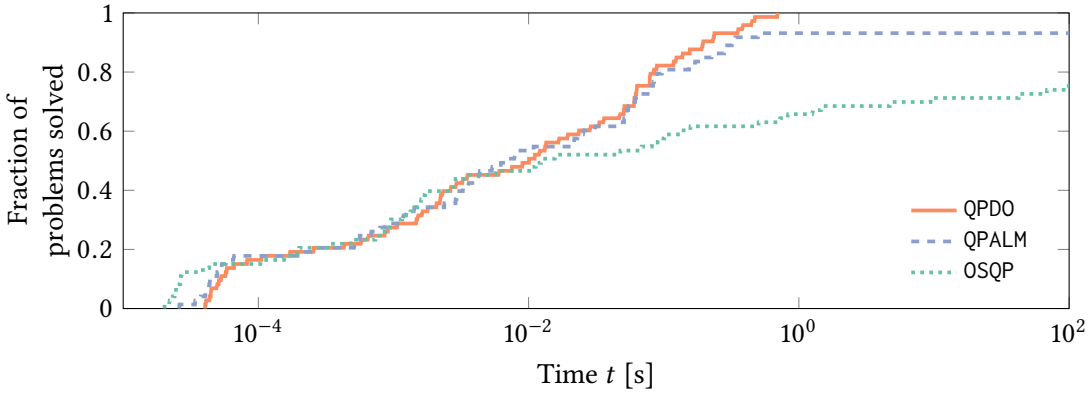


Figure 3.3: Comparison on Maros–Mészáros problems with data profiles: fraction of problems solved by each solver as a function of run time.

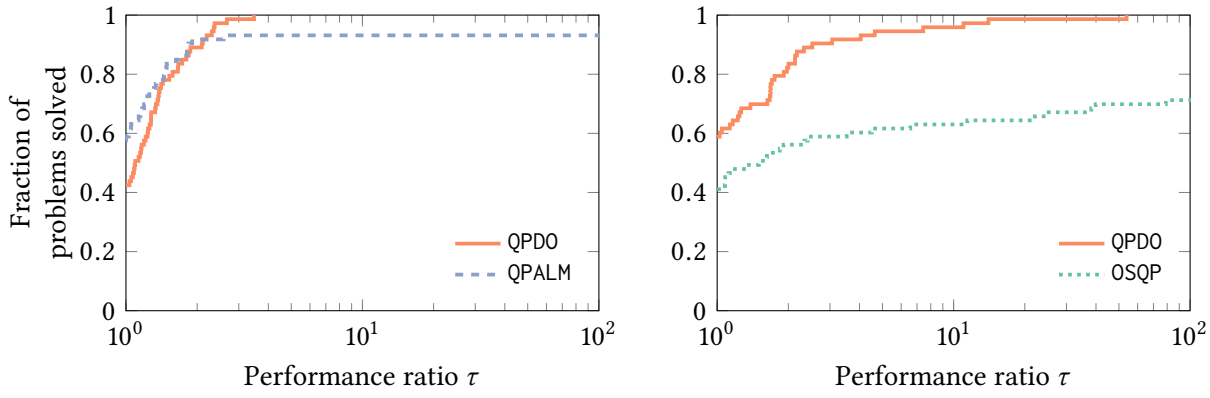


Figure 3.4: Comparison on Maros–Mészáros problems with performance profiles: fraction of problems solved by each solver as a function of performance ratio.

Chapter 4

Conclusions

This is not the end.
It is not even the beginning of the end.
But it is, perhaps, the end of the beginning.
—W. Churchill, 1942

In this thesis we have explored the class of constrained structured optimization problems and pioneered general methods for their numerical solution, based on the proximal augmented Lagrangian framework.

In Chapter 1 we have introduced constrained structured programs (NCSPs) and analyzed their necessary optimality conditions. Leveraging the augmented Lagrangian framework, we devised a nested numerical method whose subproblems are structured optimization problems. Then we proposed an algorithmic scheme able to handle the broad class of NCSPs and we were able to show its global convergence under standard assumptions. We presented our implementation `ALPX` with some illustrative examples involving vanishing and disjunctive constraints. Focusing on nonlinear programs, we benchmarked `ALPX` against a state-of-the-art NLP solver and showed its robustness and efficacy.

In Chapter 2 we considered switching time optimization (STO) problems with switching costs and constraints. Based on the sparsity-inducing cardinality function, these problems are reformulated as constrained structured problems. Then we derived routines to evaluate the simplex-constrained proximal mapping arising in fixed final time problems. Finally, numerical examples evidenced the potential of switching costs as a modelling tool.

In Chapter 3 we examined convex quadratic programming from the viewpoint of the proximal augmented Lagrangian framework. We found the method to be equivalent to a proximal point iteration, and designed the numerical scheme accordingly. Owing to the problem structure, the inner minimization procedure uses semismooth Newton's directions and exact linesearch, leading to large updates in the active set and fast linear system solves. Our solver `QPDO` demonstrated reliable and efficient performance compared to state-of-the-art solvers.

Outlook

This thesis leaves us with more questions than it answered.

Chapter 1 brings together proximal and augmented Lagrangian methods to deal with constrained structured optimization problems, following a research direction suggested in [200]. Nevertheless, we believe this could be better integrated with the continuous-Lyapunov descent (CLyD) framework developed in [214], which constitutes a significant tool for the analysis of proximal methods. Another still unanswered issue relates to the assumptions needed for ensuring convergence of our augmented Lagrangian proximal method. In particular, we believe the requirement of continuity needed for g on its domain could be relaxed, as shown in §2.3.1.

In Chapter 2 we considered STO problems with switching costs. It would be desirable to consider instead general MIOCPs, involving continuous controls, state and control constraints, boundary and switching conditions, as well as switching costs. These could be reformulated as constrained structured problems via, *e.g.*, direct multiple shooting or indirect methods [141]. Moreover, following the suggestion in [135], the STO approach should be coupled with the combinatorial integral approximation (CIA) [144] and shortest path approaches [230] to obtain tighter formulations, gain robustness, and eventually deliver better solutions.

We developed a QP solver in Chapter 3. The development of linear algebra routines for the efficient update of sparse factorizations would be greatly beneficial, as in [84, 233]. Also, one could investigate in which cases the exact linesearch procedure may yield unstable behaviour, *e.g.*, due to finite precision, and when Armijo's sufficient decrease rule may be an appropriate substitute. The forcing sequences of regularization parameters and inner tolerance also play a role; devising rules for their adaptive tuning is a topic for future research. Also warm starting could have a major impact of the performance. Running some iterations of a first-order method, *e.g.*, `OSQP` [253], could quickly improve the active set identification, hence providing `QPDO` with a good initial point. Finally, as we did for QPs, the augmented Lagrangian proximal framework could be tailored to convex programming and beyond to nonlinear programming, *mutatis mutandis*. This yields primal-dual regularized subproblems, on the vein of [213, 205, 217], that could be efficiently solved via semismooth Newton's method.

Bibliography

[1] J. Kepler. *Nova stereometria doliorum vinariorum*. 1615.

[2] G. Galilei. *Il Saggiatore*. Rome, 1623.

[3] G. Galilei. *Discorsi e dimostrazioni matematiche intorno a due nuove scienze*. Leiden, 1638.

[4] G. W. Leibniz. "Nova Methodus pro Maximis et Minimis, ..." In: *Acta Eruditorum* 10 (Oct. 1684), pp. 467–473.

[5] I. Newton. *Philosophiæ Naturalis Principia Mathematica*. 1687.

[6] J. Bernoulli. "Problema novum ad cujus solutionem Mathematici invitantur". In: *Acta Eruditorum* 18 (June 1696), p. 269.

[7] W. Karush. "Minima of functions of several variables with inequalities as side conditions". MA thesis. Chicago, Illinois: University of Chicago, Department of Mathematics, Dec. 1939.

[8] K. Levenberg. "A Method for the Solution of Certain Problems in Least Squares". In: *Quart. Appl. Math.* 2 (1944), pp. 164–168.

[9] W. Fenchel. *Convex Cones, Sets, and Functions*. Lecture Notes. Princeton, New Jersey, 1951.

[10] H. W. Kuhn and A. W. Tucker. "Nonlinear Programming". In: *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley, California: University of California Press, 1951, pp. 481–492.

[11] J. Douglas and H. H. Rachford. "On the numerical solution of heat conduction problems in two and three space variables". In: *Transactions of the American Mathematical Society* 82.2 (1956), pp. 421–439.

[12] M. Frank and P. Wolfe. "An algorithm for quadratic programming". In: *Naval Research Logistics Quarterly* 3.1–2 (1956), pp. 95–110. DOI: 10.1002/nav.3800030109.

[13] P. Wolfe. "The Simplex Method for Quadratic Programming". In: *Econometrica* 27.3 (1959), pp. 382–398. URL: http://www.jstor.org/stable/1909468.

[14] G. J. Minty. "Monotone (nonlinear) operators in Hilbert space". In: *Duke Mathematical Journal* 29.3 (Sept. 1962), pp. 341–346. DOI: 10.1215/S0012-7094-62-02933-2.

[15] D. Morrison, J. D. Riley, and J. F. Zancanaro. "Multiple Shooting Method for Two-Point Boundary Value Problems". In: 5.12 (1962), pp. 613–614. DOI: 10.1145/355580.369128.

[16] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. Mishchenko. *The mathematical theory of optimal processes*. Interscience Publishers, 1962.

[17] D. Marquardt. "An Algorithm for Least Squares Estimation on Nonlinear Parameters". In: *SIAM Journal on Applied Mathematics* 11 (1963), pp. 431–441.

[18] J. J. Moreau. "Proximité et dualité dans un espace hilbertien". fr. In: *Bulletin de la Société Mathématique de France* 93 (1965), pp. 273–299. DOI: 10.24033/bsmf.1625.

[19] L. Armijo. "Minimization of functions having Lipschitz continuous first partial derivatives". In: *Pacific Journal of Mathematics* 16.1–3 (1966).

[20] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. New York: Wiley, 1968.

[21] M. R. Hestenes. "Multiplier and gradient methods". In: *Journal of Optimization Theory and Applications* 4.5 (Nov. 1969), pp. 303–320. DOI: 10.1007/BF00927673.

[22] M. J. D. Powell. "A method for nonlinear constraints in minimization problems". In: *Optimization*. Ed. by R. Fletcher. Academic Press, 1969, pp. 283–298.

[23] M. J. D. Powell. "A Hybrid Method for Non-Linear Equations". In: *Numerical Methods for Non-Linear Algebraic Equations*. Ed. by P. Rabinowitz. London: Gordon and Breach Science, 1970, pp. 87–144.

[24] R. T. Rockafellar. "A dual approach to solving nonlinear programming problems by unconstrained optimization". In: *Mathematical Programming* 5.1 (Dec. 1973), pp. 354–373. DOI: 10.1007/BF01580138.

[25] R. T. Rockafellar. "The multiplier method of Hestenes and Powell applied to convex programming". In: *Journal of Optimization Theory and Applications* 12.6 (Dec. 1973), pp. 555–562. DOI: 10.1007/BF00934777.

[26] P. E. Gill, G. H. Golub, W. Murray, and M. A. Saunders. "Methods for modifying matrix factorizations". In: *Mathematics of Computation* 28.126 (Apr. 1974), pp. 505–535.

[27] R. T. Rockafellar. "Augmented Lagrange Multiplier Functions and Duality in Nonconvex Programming". In: *SIAM Journal on Control* 12.2 (1974), pp. 268–285. DOI: 10.1137/0312021.

[28] D. Gabay and B. Mercier. "A dual algorithm for the solution of nonlinear variational problems via finite element approximation". In: *Computers & Mathematics with Applications* 2.1 (1976), pp. 17–40. DOI: 10.1016/0898-1221(76)90003-1.

[29] R. T. Rockafellar. "Augmented Lagrangians and applications of the proximal point algorithm in convex programming". In: *Mathematics of operations research* 1.2 (May 1976), pp. 97–116. DOI: 10.1287/moor.1.2.97.

[30] R. T. Rockafellar. "Monotone Operators and the Proximal Point Algorithm". In: *SIAM Journal on Control and Optimization* 14.5 (1976), pp. 877–898. DOI: 10.1137/0314056.

[31] S. P. Han. "A globally convergent method for nonlinear programming". In: *Journal of Optimization Theory and Applications* 22.3 (July 1977), pp. 297–309. DOI: 10.1007/BF00932858.

[32] R. Mifflin. "Semismooth and Semiconvex Functions in Constrained Optimization". In: *SIAM Journal on Control and Optimization* 15.6 (1977), pp. 959–972. DOI: 10.1137/0315061.

[33] B. A. Murtagh and M. A. Saunders. "Large-scale linearly constrained optimization". In: *Mathematical Programming* 14 (1978), pp. 41–72.

[34] M. J. D. Powell. "A fast algorithm for nonlinearly constrained optimization calculations". In: *Numerical Analysis*. Ed. by G. A. (Watson. Lecture Notes in Mathematics. Berlin, Heidelberg: Springer Berlin Heidelberg, 1978, pp. 144–157. DOI: 10.1007/BFb0067703.

[35] M. J. D. Powell. "Algorithms for nonlinear constraints that use Lagrangian functions". In: *Mathematical Programming* 14 (1 Dec. 1978), pp. 224–248. DOI: 10.1007/BF01588967.

[36] D. R. Hofstadter. *Gödel, Escher, Bach: an Eternal Golden Braid*. Basic Books, 1979.

[37] P. L. Lions and B. Mercier. "Splitting Algorithms for the Sum of Two Nonlinear Operators". In: *SIAM Journal on Numerical Analysis* 16.6 (1979), pp. 964–979. DOI: 10.1137/0716071.

[38] J. Nocedal. "Updating Quasi-Newton Matrices with Limited Storage". In: *Mathematics of Computation* 35.151 (1980), pp. 773–782.

[39] B. A. Murtagh and M. A. Saunders. "A projected Lagrangian algorithm and its implementation for sparse nonlinear constraints". In: *Mathematical Programming Study* 16 (1982), pp. 84–117.

[40] F. H. Clarke. *Optimization and Nonsmooth Analysis*. New York: John Wiley & Sons, 1983.

[41] H. Bock and K. Plitt. "A Multiple Shooting Algorithm for Direct Solution of Optimal Control Problems*". In: *IFAC Proceedings Volumes* 17.2 (1984). 9th IFAC World Congress: A Bridge Between Control Science and Technology, Budapest, Hungary, 2–6 July 1984, pp. 1603–1608. DOI: 10.1016/S1474-6670(17)61205-9.

[42] N. Karmarkar. "A new polynomial-time algorithm for linear programming". In: *Combinatorica* 4.4 (Dec. 1984), pp. 373–395. DOI: 10.1007/BF02579150.

[43] F. J. Luque. "Asymptotic Convergence Analysis of the Proximal Point Algorithm". In: *SIAM Journal on Control and Optimization* 22.2 (1984), pp. 277–293. DOI: 10.1137/0322019.

[44] I. S. Duff, A. M. Erisman, and J. K. Reid. *Direct Methods for Sparse Matrices*. London: Oxford University Press, 1986.

[45] R. Fletcher. *Practical Methods of Optimization*. New York, NY, USA: Wiley-Interscience, 1987.

[46] A. R. Conn, N. I. M. Gould, and P. L. Toint. "Global Convergence of a Class of Trust Region Algorithms for Optimization with Simple Bounds". In: *SIAM Journal on Numerical Analysis* 25.2 (1988), pp. 433–460. DOI: 10.1137/0725029.

[47] B. Kummer. "Newton's method for nondifferentiable functions". In: *Advances in Mathematical Optimization*. Ed. by J. Guddat, B. Bank, H. Hollatz, P. Kall, D. Klatte, B. Kummer, K. Lommatzsch, L. Tammer, M. Vlach, and K. Zimmermann. Berlin: Akademi-Verlag, 1988, pp. 114–125.

[48] E. O. Omojokun. "Trust region algorithms for optimization with nonlinear equality and inequality constraints". PhD thesis. University of Colorado, Boulder, 1989.

[49] A. R. Conn, N. I. M. Gould, and P. L. Toint. "A Globally Convergent Augmented Lagrangian Algorithm for Optimization with General Constraints and Simple Bounds". In: *SIAM Journal on Numerical Analysis* 28.2 (Apr. 1991), pp. 545–572. DOI: 10.1137/0728030.

[50] K. L. Teo, C. J. Goh, and K. H. Wong. *A Unified computational approach for optimal control problems*. New York: Longman Scientific and Technical, 1991.

[51] A. R. Conn, N. I. M. Gould, and P. L. Toint. *LANCELOT: a Fortran package for large-scale nonlinear optimization (Release A)*. Vol. 17. Springer Series in Computational Mathematics. Heidelberg, New York: Springer Verlag, 1992.

[52] J. Eckstein and D. P. Bertsekas. "On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators". In: *Mathematical Programming* 55.1 (Apr. 1992), pp. 293–318. DOI: 10.1007/BF01581204.

[53] A. Fischer. "A special Newton-type optimization method". In: *Optimization* 24 (1992), pp. 269–284. DOI: 10.1080/02331939208843795.

[54] B. Kummer. "Newton's method based on generalized derivatives for nonsmooth functions: convergence analysis". In: *Advances in Optimization*. Ed. by W. Oettli and D. Pallaschke. Berlin, Heidelberg: Springer, 1992, pp. 171–194. DOI: 10.1007/978-3-642-51682-5_12.

[55] S. Mehrotra. "On the Implementation of a Primal-Dual Interior Point Method". In: *SIAM Journal on Optimization* 2.4 (1992), pp. 575–601. DOI: 10.1137/0802028.

[56] L. Qi. "Convergence analysis of some algorithms for solving nonsmooth equations". In: *Mathematics of Operations Research* 18.1 (Feb. 1993), pp. 227–244. DOI: 10.1287/moor.18.1.227.

[57] L. Qi and J. Sun. "A nonsmooth version of Newton's method". In: *Mathematical Programming* 58.1 (Jan. 1993), pp. 353–367. DOI: 10.1007/BF01581275.

[58] M. I. Zelikin and V. F. Borisov. *Theory of Chattering Control with applications to Astronautics, Robotics, Economics, and Engineering*. Boston: Birkhäuser, 1994. DOI: 10.1007/978-1-4612-2702-1.

[59] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. "A Limited Memory Algorithm for Bound Constrained Optimization". In: *SIAM Journal on Scientific Computing* 16.5 (1995), pp. 1190–1208. DOI: 10.1137/0916069.

[60] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. Frontiers in Applied Mathematics. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1995.

[61] R. J. Vanderbei. "Symmetric Quasidefinite Matrices". In: *SIAM Journal on Optimization* 5.1 (1995), pp. 100–113. DOI: 10.1137/0805005.

[62] D. P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Athena Scientific, 1996.

[63] R. Tibshirani. "Regression Shrinkage and Selection via the Lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1 (1996), pp. 267–288. URL: http://www.jstor.org/stable/2346178.

[64] G. D. Cheng and X. Guo. "ϵ-relaxed approach in structural topology optimization". In: *Structural Optimization* 13 (1997), pp. 258–266.

[65] A. Fischer. "Solution of monotone complementarity problems with locally Lipschitzian functions". In: *Mathematical Programming* 76.3 (Mar. 1997), pp. 513–532. DOI: 10.1007/BF02614396.

[66] A. Greenbaum. *Iterative Methods for Solving Linear Systems*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1997. DOI: 10.1137/1.9781611970937.

[67] H. W. J. Lee, K. L. Teo, V. Rehbock, and L. S. Jennings. "Control parametrization enhancing technique for time optimal control problems". In: *Dynamic Systems and Applications* 6 (Jan. 1997), pp. 243–262.

[68] J.-S. Pang. "Error bounds in mathematical programming". In: *Mathematical Programming* 79.1 (Oct. 1997), pp. 299–332. DOI: 10.1007/BF02614322.

[69] L. Qi and H. Jiang. "Semismooth Karush-Kuhn-Tucker Equations and Convergence Analysis of Newton and Quasi-Newton Methods for Solving these Equations". In: *Mathematics of Operations Research* 22.2 (1997), pp. 301–325. DOI: 10.1287/moor.22.2.301.

[70] R. T. Rockafellar. *Convex Analysis*. Princeton, NJ: Princeton University Press, 1997.

[71] J. Sun. "On piecewise quadratic Newton and trust region problems". In: *Mathematical Programming* 76.3 (Mar. 1997), pp. 451–467. DOI: 10.1007/BF02614393.

[72] S. J. Wright. *Primal-Dual Interior-Point Methods*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1997.

[73] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. "Algorithm 778: L-BFGS-B: Fortran Subroutines for Large-Scale Bound-Constrained Optimization". In: *ACM Trans. Math. Softw.* 23.4 (Dec. 1997), pp. 550–560. DOI: 10.1145/279232.279236.

[74] F. Facchinei, A. Fischer, and C. Kanzow. "On the Accurate Identification of Active Constraints". In: *SIAM Journal on Optimization* 9.1 (1998), pp. 14–32. DOI: 10.1137/S1052623496305882.

[75] H. Lee, K. Teo, and X. Cai. "An optimal control approach to nonlinear mixed integer programming problems". In: *Computers & Mathematics with Applications* 36.3 (1998), pp. 87–105.

[76] F. Facchinei, A. Fischer, C. Kanzow, and J.-M. Peng. "A Simply Constrained Optimization Reformulation of KKT Systems Arising from Variational Inequalities". In: *Applied Mathematics and Optimization* 40.1 (Feb. 1999), pp. 19–37. DOI: 10.1007/s002459900114.

[77] H. W. J. Lee, K. L. Teo, V. Rehbock, and L. S. Jennings. "Control parametrization enhancing technique for optimal discrete-valued control problems". In: *Automatica* 35.8 (1999), pp. 1401–1407.

[78] C.-J. Lin and J. J. Moré. "Newton's Method for Large Bound-Constrained Optimization Problems". In: *SIAM Journal on Optimization* 9.4 (1999), pp. 1100–1127. DOI: 10.1137/S1052623498345075.

[79] I. Maros and C. Mészáros. "A repository of convex quadratic programming problems". In: *Optimization Methods and Software* 11.1–4 (1999), pp. 671–681. DOI: 10.1080/10556789908805768.

[80] D. Sun and L. Qi. "On NCP-Functions". In: *Computational Optimization and Applications* 13.1 (Apr. 1999), pp. 201–220. DOI: 10.1023/A:1008669226453.

[81] K. L. Teo, L. S. Jennings, H. W. J. Lee, and V. Rehbock. "The control parameterization enhancing transform for constrained optimal control problems". In: *The Journal of the Australian Mathematical Society. Series B. Applied Mathematics* 40.3 (1999), pp. 314–335. DOI: 10.1017/S0334270000010936.

[82] X. Chen Bintong and Chen and C. Kanzow. "A penalized Fischer-Burmeister NCP-function". In: *Mathematical Programming* 88.1 (June 2000), pp. 211–216. DOI: 10.1007/PL00011375.

[83] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust Region Methods*. Society for Industrial and Applied Mathematics, 2000. DOI: 10.1137/1.9780898719857.

[84] T. A. Davis and W. W. Hager. "Multiple-Rank Modifications of a Sparse Cholesky Factorization". In: *SIAM Journal on Matrix Analysis and Applications* 22.4 (2001), pp. 997–1013. DOI: 10.1137/S0895479899357346.

[85] A. Locatelli. *Optimal control: An introduction*. Basel: Birkhäuser, 2001. URL: https://www.springer.com/de/book/9783764364083.

[86] D. Ruiz. *A scaling algorithm to equilibrate both rows and columns norms in matrices*. Tech. rep. RAL-TR-2001-034. Oxon, UK: Rutherford Appleton Laboratory, Sept. 2001. URL: ftp://ftp.numerical.rl.ac.uk/pub/reports/drRAL2001034.ps.gz.

[87] E. G. Birgin and J. M. Martínez. "Large-scale active-set box-constrained optimization method with spectral projected gradients". In: *Computational Optimization and Applications* 23.1 (2002), pp. 101–125.

[88] E. D. Dolan and J. J. Moré. "Benchmarking optimization software with performance profiles". In: *Mathematical Programming* 91.2 (2002), pp. 201–213. DOI: 10.1007/s101070100263.

[89] R. Fletcher and S. Leyffer. "Nonlinear programming without a penalty function". In: *Mathematical Programming* 91.2 (Jan. 2002), pp. 239–269. DOI: 10.1007/s101070100244.

[90] A. Forsgren, P. E. Gill, and M. H. Wright. "Interior Methods for Nonlinear Optimization". In: *SIAM Review* 44.4 (2002), pp. 525–597. DOI: 10.1137/S0036144502414942.

[91] P. Gill, W. Murray, and M. Saunders. "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization". In: *SIAM Journal on Optimization* 12.4 (2002), pp. 979–1006. DOI: 10.1137/S1052623499350013.

[92] M. Hintermüller, K. Ito, and K. Kunisch. "The Primal-Dual Active Set Strategy as a Semismooth Newton Method". In: *SIAM Journal on Optimization* 13.3 (2002), pp. 865–888. DOI: 10.1137/S1052623401383558.

[93] H. Y. Benson and R. J. Vanderbei. "Solving Problems with Semidefinite and Related Constraints Using Interior-Point Methods for Nonlinear Programming". In: *Mathematical Programming* 95.2 (Feb. 2003), pp. 279–302. DOI: 10.1007/s10107-002-0350-x.

[94] N. I. M. Gould and S. Leyffer. "An Introduction to Algorithms for Nonlinear Optimization". In: *Frontiers in Numerical Analysis: Durham 2002*. Ed. by J. F. Blowey, A. W. Craig, and T. Shardlow. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 109–197. DOI: 10.1007/978-3-642-55692-0_4.

[95] M. Kocvara and M. Stingl. "PENNON". In: *High Performance Algorithms and Software for Nonlinear Optimization*. Ed. by G. Di Pillo and A. Murli. Boston, MA: Springer US, 2003, pp. 303–321. DOI: 10.1007/978-1-4613-0241-4_14.

[96] S. Pieraccini, M. G. Gasparo, and A. Pasquali. "Global Newton-type methods and semismooth reformulations for NCP". In: *Applied Numerical Mathematics* 44.3 (2003), pp. 367–384. DOI: 10.1016/S0168-9274(02)00169-1.

[97] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Mar. 2004.

[98] P. Deuflhard. *Newton Methods for Nonlinear Problems. Affine Invariance and Adaptive Algorithms*. Vol. 35. Series Computational Mathematics. Springer, 2004. DOI: 10.1007/978-3-642-23899-4.

[99] M. Ulbrich, S. Ulbrich, and L. N. Vicente. "A globally convergent primal-dual interior-point filter method for nonlinear programming". In: *Mathematical Programming* 100.2 (June 2004), pp. 379–410. DOI: 10.1007/s10107-003-0477-4.

[100] R. B. Wilson. "A Simplicial Method for Convex Programming". PhD thesis. Harvard University, 2004.

[101] T. A. Davis. "Algorithm 849: A Concise Sparse Cholesky Factorization Package". In: *ACM Trans. Math. Softw.* 31.4 (Dec. 2005), pp. 587–591. DOI: 10.1145/1114268.1114277.

[102] M. Gerdts. "Solving mixed-integer optimal control problems by branch & bound: a case study from automobile test-driving with gear shift". In: *Optimal Control Applications and Methods* 26.1 (2005), pp. 1–18. DOI: 10.1002/oca.751.

[103] P. E. Gill, W. Murray, and M. A. Saunders. "SNOPT: An SQP algorithm for large-scale constrained optimization". In: *SIAM Review. A Publication of the Society for Industrial and Applied Mathematics* 47 (2005), pp. 99–131.

[104] N. I. M. Gould, D. Orban, and Ph. L. Toint. "Numerical methods for large-scale nonlinear optimization". In: *Acta Numerica* 14 (May 2005), pp. 299–361. DOI: 10.1017/S0962492904000248.

[105] S. Sager. "Numerical methods for mixed-integer optimal control problems". Interdisciplinary Center for Scientific Computing. PhD thesis. University of Heidelberg, 2005.

[106] A. Wächter and L. T. Biegler. "Line Search Filter Methods for Nonlinear Programming: Local Convergence". In: *SIAM Journal on Optimization* 16.1 (2005), pp. 32–48. DOI: 10.1137/S1052623403426544.

[107] A. Wächter and L. T. Biegler. "Line Search Filter Methods for Nonlinear Programming: Motivation and Global Convergence". In: *SIAM Journal on Optimization* 16.1 (2005), pp. 1–31. DOI: 10.1137/S1052623403426556.

[108] M. H. Wright. "The interior-point revolution in optimization: history, recent developments, and lasting consequences". In: *American Mathematical Society. Bulletin. New Series* 42.1 (2005), pp. 39–56. DOI: 10.1090/S0273-0979-04-01040-7.

[109] R. H. Byrd, J. Nocedal, and R. A. Waltz. "Knitro: An Integrated Package for Nonlinear Optimization". In: *Large-Scale Nonlinear Optimization*. Ed. by G. Di Pillo and M. Roma. Boston, MA: Springer US, 2006, pp. 35–59. DOI: 10.1007/0-387-30065-1_4.

[110] M. Egerstedt, Y. Wardi, and H. Axelsson. "Transition-time optimization for switched-mode dynamical systems". In: *IEEE Transactions on Automatic Control* 51.1 (Jan. 2006), pp. 110–115. DOI: 10.1109/TAC.2005.861711.

[111] M. Gerdts. "A variable time transformation method for mixed-integer optimal control problems". In: *Optimal Control Applications and Methods* 27.3 (2006), pp. 169–182. DOI: 10.1002/oca.778.

[112] J. Nocedal and S. J. Wright. *Numerical Optimization*. 2nd. New York, NY, USA: Springer, 2006.

[113] A. Wächter and L. T. Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". In: *Mathematical Programming* 106.1 (Mar. 2006), pp. 25–57. DOI: 10.1007/s10107-004-0559-y.

[114] D. P. Robinson. "Primal-Dual Methods for Nonlinear Optimization". PhD thesis. University of California, San Diego, Sept. 2007.

[115] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt. "On Augmented Lagrangian Methods with General Lower–Level Constraints". In: *SIAM Journal on Optimization* 18.4 (2008), pp. 1286–1309. DOI: 10.1137/060654797.

[116] E. van den Berg and M. P. Friedlander. "Probing the Pareto frontier for basis pursuit solutions". In: *SIAM Journal on Scientific Computing* 31.2 (2008), pp. 890–912. DOI: 10.1137/080714488.

[117] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam. "Algorithm 887: CHOLMOD, Supernodal Sparse Cholesky Factorization and Update/Downdate". In: *ACM Trans. Math. Softw.* 35.3 (Oct. 2008), pp. 1–14. DOI: 10.1145/1391989.1391995.

[118] F. E. Curtis and J. Nocedal. "Flexible penalty functions for nonlinear constrained optimization". In: *IMA Journal of Numerical Analysis* 28.4 (Mar. 2008), pp. 749–769. DOI: 10.1093/imanum/drn003.

[119] H.-r. Fang and D. P. OLeary. "Modified Cholesky algorithms: a catalog with new approaches". In: *Mathematical Programming* 115.2 (Oct. 2008), pp. 319–349. DOI: 10.1007/s10107-007-0177-6.

[120] M. P. Friedlander and P. Tseng. "Exact Regularization of Convex Programs". In: *SIAM Journal on Optimization* 18.4 (2008), pp. 1326–1350. DOI: 10.1137/060675320.

[121] M. Gerdts and M. Kunkel. "A nonsmooth Newton's method for discretized optimal control problems with state and control constraints". In: *Journal of Industrial & Management Optimization* 4.2 (2008), pp. 247–270. DOI: 10.3934/jimo.2008.4.247.

[122] A. Beck and M. Teboulle. "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems". In: *SIAM Journal on Imaging Sciences* 2.1 (2009), pp. 183–202. DOI: 10.1137/080716542.

[123] A. L. Dontchev and R. T. Rockafellar. *Implicit functions and solution mappings*. Monogr. Math. Springer, 2009.

[124] T. Hoheisel. "Mathematical Programs with Vanishing Constraints". PhD thesis. University of Würzburg, 2009.

[125] J. J. Moré and S. M. Wild. "Benchmarking Derivative-Free Optimization Algorithms". In: *SIAM Journal on Optimization* 20.1 (2009), pp. 172–191. DOI: 10.1137/080724083.

[126] S. Sager. "Reformulations and algorithms for the optimization of switching decisions in nonlinear optimal control". In: *Journal of Process Control* 19.8 (Sept. 2009). Special Section on Hybrid Systems: Modeling, Simulation and Optimization, pp. 1238–1247. DOI: 10.1016/j.jprocont.2009.03.008.

[127] M. H. Wright. *The Dual Flow Between Linear Algebra and Optimization*. History of Numerical Linear Algebra Minisymposium - Part II, SIAM Conference on Applied Linear Algebra (Monterey, California) (Accessed June 2021). Oct. 2009. URL: https://www.siam.org/meetings/la09/talks/wright.pdf.

[128] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo. "Sparse Reconstruction by Separable Approximation". In: *IEEE Transactions on Signal Processing* 57.7 (2009), pp. 2479–2493. DOI: 10.1109/TSP.2009.2016892.

[129] R. Andreani, J. M. Martínez, and B. F. Svaiter. "A New Sequential Optimality Condition for Constrained Optimization and Algorithmic Consequences". In: *SIAM Journal on Optimization* 20.6 (Jan. 2010), pp. 3533–3554. DOI: 10.1137/090777189.

[130] A. Bradley. "Algorithms for the equilibration of matrices and their application to limited-memory quasi-Newton methods". PhD thesis. Stanford, CA: Stanford University, May 2010.

[131] E. V. Castelani, A. L. M. Martinez, J. M. Martínez, and B. F. Svaiter. "Addressing the greediness phenomenon in Nonlinear Programming by means of Proximal Augmented Lagrangians". In: *Computational Optimization and Applications* 46.2 (June 2010), pp. 229–245. DOI: 10.1007/s10589-009-9271-4.

[132] J. Gondzio. "Matrix-free interior point method". In: *Computational Optimization and Applications* 51.2 (Oct. 2010), pp. 457–480. DOI: 10.1007/s10589-010-9361-3.

[133] N. I. M. Gould and P. L. Toint. "Nonlinear programming without a penalty function or a filter". In: *Mathematical Programming* 122.1 (Mar. 2010), pp. 155–196. DOI: 10.1007/s10107-008-0244-7.

[134] C. Kirches. "Fast numerical methods for mixed-integer nonlinear model-predictive control". PhD thesis. Heidelberg University, July 2010. DOI: 10.11588/heidok.00011636.

[135] C. Kirches, S. Sager, H. G. Bock, and J. P. Schlöder. "Time-optimal control of automobile test drives with gear shifts". In: *Optimal Control Applications and Methods* 31.2 (2010), pp. 137–153. DOI: 10.1002/oca.892.

[136] R. Andreani, G. Haeser, and J. M. Martínez. "On sequential optimality conditions for smooth constrained optimization". In: *Optimization* 60.5 (2011), pp. 627–641. DOI: 10.1080/02331930903578700.

[137] S. Becker, J. Bobin, and E. J. Candés. "NESTA: A Fast and Accurate First-Order Method for Sparse Recovery". In: *SIAM Journal on Imaging Sciences* 4.1 (2011), pp. 1–39. DOI: 10.1137/090756855.

[138] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. now, 2011. DOI: 10.1561/2200000016.

[139] A. Chambolle and T. Pock. "A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging". In: *Journal of Mathematical Imaging and Vision* 40.1 (May 2011), pp. 120–145. DOI: 10.1007/s10851-010-0251-1.

[140] P. L. Combettes and J.-C. Pesquet. "Proximal splitting methods in signal processing". In: *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. New York: Springer, 2011, pp. 185–212.

[141] M. Gerdts. *Optimal Control of ODEs and DAEs*. De Gruyter, 2011. DOI: 10.1515/9783110249996.

[142] B. Houska, H. J. Ferreau, and M. Diehl. "ACADO toolkit — An open-source framework for automatic control and dynamic optimization". In: *Optimal Control Applications and Methods* 32.3 (2011), pp. 298–312. DOI: 10.1002/oca.939.

[143] R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis*. Vol. 317. Berlin: Springer, 2011.

[144] S. Sager, M. Jung, and C. Kirches. "Combinatorial integral approximation". In: *Mathematical Methods of Operations Research* 73.3 (Apr. 2011), pp. 363–380. DOI: 10.1007/s00186-011-0355-4.

[145] E. G. Birgin and J. M. Martínez. "Augmented Lagrangian method with nonmonotone penalty parameters for constrained optimization". In: *Computational Optimization and Applications* 51.3 (Apr. 2012), pp. 941–965. DOI: 10.1007/s10589-011-9396-0.

[146] F. E. Curtis. "A penalty-interior-point algorithm for nonlinear constrained optimization". In: *Mathematical Programming Computation* 4.2 (June 2012), pp. 181–209. DOI: 10.1007/s12532-012-0041-4.

[147] A. Domahidi, A. U. Zgraggen, M. N. Zeilinger, M. Morari, and C. N. Jones. "Efficient interior point methods for multistage problems arising in receding horizon control". In: *51st IEEE Conference on Decision and Control (CDC)*. Maui, HI, 2012, pp. 668–674. DOI: 10.1109/CDC.2012.6426855.

[148] M. P. Friedlander and D. Orban. "A primal-dual regularized interior-point method for convex quadratic programs". In: *Mathematical Programming Computation* 4.1 (Mar. 2012), pp. 71–107. DOI: 10.1007/s12532-012-0035-2.

[149] P. E. Gill and D. P. Robinson. "A primal-dual augmented Lagrangian". In: *Computational Optimization and Applications* 51.1 (Jan. 2012), pp. 1–25. DOI: 10.1007/s10589-010-9339-1.

[150] P. E. Gill and E. Wong. "Sequential Quadratic Programming Methods". In: *Mixed Integer Nonlinear Programming*. Ed. by J. Lee and S. (Leyffer. Vol. 154. The IMA Volumes in Mathematics and its Applications. New York: Springer, 2012, pp. 147–224. DOI: 10.1007/978-1-4614-1927-3_6.

[151] J. Gondzio. "Interior point methods 25 years later". In: *European Journal of Operational Research* 218.3 (2012), pp. 587–601. DOI: 10.1016/j.ejor.2011.09.017.

[152] T. Hoheisel, C. Kanzow, and A. Schwartz. "Mathematical programs with vanishing constraints: a new regularization approach with strong convergence properties". In: *Optimization* 61.6 (2012), pp. 619–636. DOI: 10.1080/02331934.2011.608164.

[153] S. Sager. "A benchmark library of mixed-integer optimal control problems". In: *Mixed Integer Nonlinear Programming*. Ed. by J. Lee and S. Leyffer. New York, NY: Springer New York, 2012, pp. 631–670. URL: https://mathopt.de/PUBLICATIONS/Sager2012b.pdf.

[154] S. Sager, H. G. Bock, and M. Diehl. "The integer approximation error in mixed-integer optimal control". In: *Mathematical Programming* 133.1 (June 2012), pp. 1–23. DOI: 10.1007/s10107-010-0405-3.

[155] H. Attouch, J. Bolte, and B. F. Svaiter. "Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods". In: *Mathematical Programming* 137.1 (Feb. 2013), pp. 91–129. DOI: 10.1007/s10107-011-0484-9.

[156] A. Beck and Y. C. Eldar. "Sparsity Constrained Nonlinear Optimization: Optimality Conditions and Algorithms". In: *SIAM Journal on Optimization* 23.3 (2013), pp. 1480–1509. DOI: 10.1137/120869778.

[157] C. Büskens and D. Wassel. "The ESA NLP Solver WORHP". In: *Modeling and Optimization in Space Engineering*. Ed. by G. Fasano and J. D. Pintér. Vol. 73. Springer New York, 2013, pp. 85–110. DOI: 10.1007/978-1-4614-4469-5_4.

[158] P. E. Gill and D. P. Robinson. "A Globally Convergent Stabilized SQP Method". In: *SIAM Journal on Optimization* 23.4 (2013), pp. 1983–2010. DOI: 10.1137/120882913.

[159] C. Kirches, A. Potschka, H. G. Bock, and S. Sager. "A Parametric Active Set Method for a Subclass of Quadratic Programs with Vanishing Constraints". In: *Pacific Journal of Optimization* 9.2 (2013), pp. 275–299. URL: http://www.ybook.co.jp/online2/pjov9n2.html.

[160] R. Loxton, Q. Lin, and K. L. Teo. "Minimizing control variation in nonlinear optimal control". In: *Automatica* 49.9 (2013), pp. 2652–2664. DOI: j.automatica.2013.05.027.

[161] P. Patrinos and A. Bemporad. "Proximal Newton methods for convex composite optimization". In: *52nd IEEE Conference on Decision and Control (CDC)*. Florence, Italy, Dec. 2013, pp. 2358–2363. DOI: 10.1109/CDC.2013.6760233.

[162] U. Ali and M. Egerstedt. "Optimal control of switched dynamical systems under dwell time constraints". In: *53rd IEEE Conference on Decision and Control (CDC)*. Dec. 2014, pp. 4673–4678. DOI: 10.1109/CDC.2014.7040117.

[163] E. G. Birgin and J. M. Martínez. *Practical Augmented Lagrangian Methods for Constrained Optimization*. Philadelphia, PA: Society for Industrial and Applied Mathematics, Apr. 2014.

[164] J. Bolte, S. Sabach, and M. Teboulle. "Proximal alternating linearized minimization for nonconvex and nonsmooth problems". In: *Mathematical Programming* 146.1 (Aug. 2014), pp. 459–494. DOI: 10.1007/s10107-013-0701-9.

[165] J. V. Burke, F. E. Curtis, and H. Wang. "A Sequential Quadratic Optimization Algorithm with Rapid Infeasibility Detection". In: *SIAM Journal on Optimization* 24.2 (2014), pp. 839–872. DOI: 10.1137/120880045.

[166] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl. "qpOASES: A parametric active-set algorithm for quadratic programming". In: *Mathematical Programming Computation* 6.4 (Dec. 2014), pp. 327–363. DOI: 10.1007/s12532-014-0071-1.

[167] A. F. Izmailov and M. V. Solodov. *Newton-Type Methods for Optimization and Variational Problems*. Springer, 2014. DOI: 10.1007/978-3-319-04247-3.

[168] J. D. Lee, Y. Sun, and M. A. Saunders. "Proximal Newton-Type Methods for Minimizing Composite Functions". In: *SIAM Journal on Optimization* 24.3 (2014), pp. 1420–1443. DOI: 10.1137/130921428.

[169] D. O'Connor and L. Vandenberghe. "Primal-Dual Decomposition by Operator Splitting and Applications to Image Deblurring". In: *SIAM Journal on Imaging Sciences* 7.3 (2014), pp. 1724–1754. DOI: 10.1137/13094671X.

[170] N. Parikh and S. Boyd. "Proximal Algorithms". In: *Foundations and Trends in Optimization* 1.3 (2014), pp. 127–239. DOI: 10.1561/2400000003.

[171] P. Patrinos and A. Bemporad. "An Accelerated Dual Gradient-Projection Algorithm for Embedded Linear Model Predictive Control". In: *IEEE Transactions on Automatic Control* 59.1 (Jan. 2014), pp. 18–33. DOI: 10.1109/TAC.2013.2275667.

[172] Y. Sun, G. Aw, R. Loxton, and K. L. Teo. "An Optimal Machine Maintenance Problem with Probabilistic State Constraints". In: *Information Sciences* 281 (Oct. 2014), pp. 386–398. DOI: 10.1016/j.ins.2014.05.051.

[173] E. G. Birgin, J. M. Martínez, and L. F. Prudente. "Optimality properties of an Augmented Lagrangian method on infeasible problems". In: *Computational Optimization and Applications* 60.3 (Apr. 2015), pp. 609–631. DOI: 10.1007/s10589-014-9685-5.

[174] O. Burdakov, C. Kanzow, and A. Schwartz. "On a Reformulation of Mathematical Programs with Cardinality Constraints". In: *Advances in Global Optimization*. Ed. by D. Gao, N. Ruan, and W. Xing. Cham: Springer, 2015, pp. 3–14. DOI: 10.1007/978-3-319-08377-3_1.

[175] N. I. M. Gould, D. Orban, and P. L. Toint. "CUTEst: a Constrained and Unconstrained Testing Environment with safe threads for mathematical optimization". In: *Computational Optimization and Applications* 60.3 (Apr. 2015), pp. 545–557. DOI: 10.1007/s10589-014-9687-3.

[176] D. Orban. "Limited-memory $LDL^\top$ factorization of symmetric quasi-definite matrices with application to constrained optimization". In: *Numerical Algorithms* 70.1 (Sept. 2015), pp. 9–41. DOI: 10.1007/s11075-014-9933-x.

[177] D. P. Robinson. "Primal-Dual Active-Set Methods for Large-Scale Optimization". In: *Journal of Optimization Theory and Applications* 166.1 (July 2015), pp. 137–171. DOI: 10.1007/s10957-015-0708-x.

[178] Y. Wardi, M. Egerstedt, and M. Hale. "Switched-mode systems: gradient-descent algorithms with Armijo step sizes". In: *Discrete Event Dynamic Systems* 25.4 (Dec. 2015), pp. 571–599. DOI: 10.1007/s10626-014-0198-2.

[179] S. Arreckx, A. Lambe, J. R. Martins, and D. Orban. "A matrix-free augmented Lagrangian algorithm with application to large-scale structural design optimization". In: *Optimization and Engineering* 17.2 (June 2016), pp. 359–384. DOI: 10.1007/s11081-015-9287-9.

[180] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2016.

[181] F. Biral, E. Bertolazzi, and P. Bosetti. "Notes on Numerical Methods for Solving Optimal Control Problems". In: *IEEJ Journal of Industry Applications* 5.2 (2016), pp. 154–166. DOI: 10.1541/ieejjia.5.154.

[182] R. H. Byrd, J. Nocedal, and F. Oztoprak. "An inexact successive quadratic approximation method for L-1 regularized optimization". In: *Mathematical Programming* 157.2 (June 2016), pp. 375–396. DOI: 10.1007/s10107-015-0941-y.

[183] P. E. Gill, V. Kungurtsev, and D. P. Robinson. "A stabilized SQP method: global convergence". In: *IMA Journal of Numerical Analysis* 37.1 (May 2016), pp. 407–443. DOI: 10.1093/imanum/drw004.

[184] N. Gould and J. Scott. "A Note on Performance Profiles for Benchmarking Software". In: *ACM Trans. Math. Softw.* 43.2 (Sept. 2016). DOI: 10.1145/2950048.

[185] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd. "Conic Optimization via Operator Splitting and Homogeneous Self-Dual Embedding". In: *Journal of Optimization Theory and Applications* 169.3 (June 2016), pp. 1042–1068. DOI: 10.1007/s10957-016-0892-3.

[186] N. Perraudin, V. Kalofolias, D. Shuman, and P. Vandergheynst. *UNLocBoX: a MATLAB convex optimization toolbox using proximal-splitting methods*. 2016. arXiv: 1402.0779.

[187] G. Stathopoulos, H. Shukla, A. Szucs, Y. Pu, and C. N. Jones. *Operator Splitting methods in control*. 2016. DOI: 10.1561/2600000008.

[188] B. Stellato, S. Ober-Blöbaum, and P. J. Goulart. "Optimal Control of Switching Times in Switched Linear Systems". In: *55th IEEE Conference on Decision and Control (CDC)*. Las Vegas, NV, Dec. 2016, pp. 7228–7233. DOI: 10.1109/CDC.2016.7799384.

[189] A. Ali, E. Wong, and J. Z. Kolter. "A Semismooth Newton Method for Fast, Generic Convex Programming". In: *Proceedings of the 34th International Conference on Machine Learning (ICML)*. Sydney, Aug. 2017, pp. 70–79. arXiv: 1705.00772 [math.OC]. URL: http://proceedings.mlr.press/v70/ali17a.html.

[190] P. Armand and R. Omheni. "A globally and quadratically convergent primal–dual augmented Lagrangian algorithm for equality constrained optimization". In: *Optimization Methods and Software* 32.1 (2017), pp. 1–21. DOI: 10.1080/10556788.2015.1025401.

[191] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2017. DOI: 10.1007/978-3-319-48311-5.

[192] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. "Julia: A fresh approach to numerical computing". In: *SIAM Review* 59.1 (2017), pp. 65–98. DOI: 10.1137/141000671.

[193] D. Davis and W. Yin. "A Three-Operator Splitting Scheme and its Optimization Applications". In: *Set-Valued and Variational Analysis* 25.4 (Dec. 2017), pp. 829–858. DOI: 10.1007/s11228-017-0421-z.

[194] N. K. Dhingra, S. Z. Khong, and M. R. Jovanović. "A second order primal-dual algorithm for nonsmooth convex composite optimization". In: *56th IEEE Conference on Decision and Control (CDC)*. Dec. 2017, pp. 2868–2873. DOI: 10.1109/CDC.2017.8264075.

[195] P. E. Gill, V. Kungurtsev, and D. P. Robinson. "A stabilized SQP method: superlinear convergence". In: *Mathematical Programming* 163.1 (May 2017), pp. 369–410. DOI: 10.1007/s10107-016-1066-7.

[196] C. Kanzow and D. Steck. "An example comparing the standard and safeguarded augmented Lagrangian methods". In: *Operations Research Letters* 45.6 (2017), pp. 598–603. DOI: 10.1016/j.orl.2017.09.005.

[197] P. Latafat and P. Patrinos. "Asymmetric forward-backward-adjoint splitting for solving monotone inclusions involving three operators". In: *Computational Optimization and Applications* 68.1 (Sept. 2017), pp. 57–93. DOI: 10.1007/s10589-017-9909-6.

[198] D. Orban and M. Arioli. *Iterative Solution of Symmetric Quasi-Definite Linear Systems*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2017. DOI: 10.1137/1.9781611974737.

[199] M. Ringkamp, S. Ober-Blöbaum, and S. Leyendecker. "On the time transformation of mixed integer optimal control problems using a consistent fixed integer control function". In: *Math. Program. Ser. A* 161 (2017), pp. 551–581. DOI: 10.1007/s10107-016-1023-5.

[200] L. Stella. "Proximal envelopes: Smooth optimization algorithms for nonsmooth problems". PhD thesis. IMT School for Advanced Studies, Lucca, 2017.

[201] L. Stella, A. Themelis, and P. Patrinos. "Forward-backward quasi-Newton methods for nonsmooth optimization problems". In: *Computational Optimization and Applications* 67.3 (July 2017), pp. 443–487. DOI: 10.1007/s10589-017-9912-y.

[202] L. Stella, A. Themelis, P. Sopasakis, and P. Patrinos. "A simple and efficient algorithm for nonlinear model predictive control". In: *56th IEEE Conference on Decision and Control (CDC)*. Melbourne, VIC: IEEE, 2017, pp. 1939–1944. DOI: 10.1109/CDC.2017.8263933.

[203] B. Stellato. "Mixed-Integer Optimal Control of Fast Dynamical Systems". PhD thesis. University of Oxford, 2017. URL: https://ora.ox.ac.uk/objects/uuid:b8a7323c-e36e-45ec-ae8d-6c9eb4350629.

[204] B. Stellato, S. Ober-Blöbaum, and P. J. Goulart. "Second-Order Switching Time Optimization for Switched Dynamical Systems". In: *IEEE Transaction on Automatic Control* 62.10 (2017), pp. 5407–5414. DOI: 10.1109/TAC.2017.2697681.

[205] S. Arreckx and D. Orban. "A Regularized Factorization-Free Method for Equality-Constrained Optimization". In: *SIAM Journal on Optimization* 28.2 (2018), pp. 1613–1639. DOI: 10.1137/16M1088570.

[206] E. Balas. *Disjunctive Programming*. Cham: Springer, 2018. DOI: 10.1007/978-3-030-00148-3.

[207] A. Beck and N. Hallak. "Optimization problems involving group sparsity terms". In: *Mathematical Programming* (Apr. 2018). DOI: 10.1007/s10107-018-1277-1.

[208] A. Bemporad. "A Numerically Stable Solver for Positive Semidefinite Quadratic Programs Based on Nonnegative Least Squares". In: *IEEE Transactions on Automatic Control* 63.2 (Feb. 2018), pp. 525–531. DOI: 10.1109/TAC.2017.2735938.

[209] M.-C. Corbineau, E. Chouzenoux, and J.-C. Pesquet. "PIPA: A New Proximal Interior Point Algorithm for Large-Scale Convex Optimization". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Calgary, AB: IEEE, Apr. 2018, pp. 1343–1347. DOI: 10.1109/icassp.2018.8462173.

[210] M. Feng, J. E. Mitchell, J.-S. Pang, X. Shen, and A. Wächter. "Complementarity Formulations of ℓ_0-norm Optimization Problems". In: *Pacific Journal of Optimization* 14.2 (2018), pp. 273–305.

[211] C. Kanzow, D. Steck, and D. Wachsmuth. "An Augmented Lagrangian Method for Optimization Problems in Banach Spaces". In: *SIAM Journal on Control and Optimization* 56.1 (2018), pp. 272–291. DOI: 10.1137/16M1107103.

[212] R. Kuhlmann and C. Büskens. "A primal–dual augmented Lagrangian penalty-interior-point filter line search algorithm". In: *Mathematical Methods of Operations Research* 87.3 (June 2018), pp. 451–483. DOI: 10.1007/s00186-017-0625-x.

[213] D. Ma, K. L. Judd, D. Orban, and M. A. Saunders. "Stabilized Optimization Via an NCL Algorithm". In: *Numerical Analysis and Optimization*. Ed. by M. Al-Baali, L. Grandinetti, and A. Purnama. Cham: Springer, 2018, pp. 173–191. DOI: 10.1007/978-3-319-90026-1_8.

[214] A. Themelis. "Proximal Algorithms for Structured Nonconvex Optimization". Faculty of Engineering Science. PhD thesis. KU Leuven, Arenberg Doctoral School, Dec. 2018.

[215] A. Themelis, L. Stella, and P. Patrinos. "Forward-Backward Envelope for the Sum of Two Nonconvex Functions: Further Properties and Nonmonotone Linesearch Algorithms". In: *SIAM Journal on Optimization* 28.3 (2018), pp. 2274–2303. DOI: 10.1137/16M1080240.

[216] R. Andreani, N. S. Fazzio, M. L. Schuverdt, and L. D. Secchin. "A Sequential Optimality Condition Related to the Quasi-normality Constraint Qualification and Its Algorithmic Consequences". In: *SIAM Journal on Optimization* 29.1 (2019), pp. 743–766. DOI: 10.1137/17M1147330.

[217] P. Armand and N. N. Tran. "An Augmented Lagrangian Method for Equality Constrained Optimization with Rapid Infeasibility Detection Capabilities". In: *Journal of Optimization Theory and Applications* 181.1 (Apr. 2019), pp. 197–215. DOI: 10.1007/s10957-018-1401-7.

[218] G. Banjac, P. Goulart, B. Stellato, and S. Boyd. "Infeasibility Detection in the Alternating Direction Method of Multipliers for Convex Optimization". In: *Journal of Optimization Theory and Applications* 183.2 (2019), pp. 490–519. DOI: 10.1007/s10957-019-01575-y.

[219] F. Bestehorn, C. Hansknecht, C. Kirches, and P. Manns. "A switching cost aware rounding method for relaxations of mixed-integer optimal control problems". In: *58th IEEE Conference on Decision and Control (CDC)*. Nice, France, Dec. 2019, pp. 7134–7139. DOI: 10.1109/CDC40024.2019.9030063.

[220] C. Clason, S. Mazurenko, and T. Valkonen. "Acceleration and Global Convergence of a First-Order Primal-Dual Method for Nonconvex Problems". In: *SIAM Journal on Optimization* 29.1 (2019), pp. 933–963. DOI: 10.1137/18M1170194.

[221] A. De Marchi. "On the Mixed-Integer Linear-Quadratic Optimal Control with Switching Cost". In: *IEEE Control Systems Letters* 3.4 (Oct. 2019), pp. 990–995. DOI: 10.1109/LCSYS.2019.2920425.

[222] N. K. Dhingra, S. Z. Khong, and M. R. Jovanović. "The Proximal Augmented Lagrangian Method for Nonsmooth Composite Optimization". In: *IEEE Transactions on Automatic Control* 64.7 (July 2019), pp. 2861–2868. DOI: 10.1109/TAC.2018.2867589.

[223] J.-P. Dussault, M. Frappier, and J. C. Gilbert. *Polyhedral Newton–min algorithms for complementarity problems*. Tech. rep. INRIA Paris; Université de Sherbrooke (Québec, Canada), Oct. 2019. HAL: 02306526.

[224] B. Hermans, A. Themelis, and P. Patrinos. "QPALM: A Newton-type Proximal Augmented Lagrangian Method for Quadratic Programs". In: *IEEE 58th Conference on Decision and Control (CDC)*. Nice, France, Dec. 2019, pp. 4325–4330. DOI: 10.1109/CDC40024.2019.9030211.

[225] C. Kanzow, P. Mehlitz, and D. Steck. "Relaxation schemes for mathematical programmes with switching constraints". In: *Optimization Methods and Software* (2019), pp. 1–36. DOI: 10.1080/10556788.2019.1663425.

[226] C. Kirches, E. A. Kostina, A. Meyer, and M. Schlöder. *Numerical Solution of Optimal Control Problems with Switches, Switching Costs and Jumps*. Mar. 2019. URL: http://www.optimization-online.org/DB_FILE/2018/10/6888.pdf.

[227] A. Themelis and P. Patrinos. "SuperMann: A Superlinearly Convergent Algorithm for Finding Fixed Points of Nonexpansive Operators". In: *IEEE Transactions on Automatic Control* 64.12 (Dec. 2019), pp. 4875–4890. DOI: 10.1109/TAC.2019.2906393.

[228] R. Andreani, G. Haeser, L. M. Mito, A. Ramos, and L. D. Secchin. *On the best achievable quality of limit points of augmented Lagrangian schemes*. Submitted. July 2020. URL: http://www.optimization-online.org/DB_HTML/2020/07/7929.html.

[229] N. Antonello, L. Stella, P. Patrinos, and T. van Waterschoot. *Proximal Gradient Algorithms: Applications in Signal Processing*. Submitted. 2020. arXiv: 1803.01621 [eess.SP].

[230] F. Bestehorn, C. Hansknecht, C. Kirches, and P. Manns. "Mixed-Integer Optimal Control Problems with switching costs: A shortest path approach". In: *Mathematical Programming Series B* (Oct. 2020). DOI: 10.1007/s10107-020-01581-3.

[231] A. Britzelmeier and M. Gerdts. "A Nonsmooth Newton Method for Linear Model-Predictive Control in Tracking Tasks for a Mobile Robot With Obstacle Avoidance". In: *IEEE Control Systems Letters* 4.4 (2020), pp. 886–891. DOI: 10.1109/LCSYS.2020.2996959.

[232] J. V. Burke, F. E. Curtis, H. Wang, and J. Wang. "Inexact Sequential Quadratic Optimization with Penalty Parameter Updates within the QP Solver". In: *SIAM Journal on Optimization* 30.3 (2020), pp. 1822–1849. DOI: 10.1137/18M1176488.

[233] K. Cheshmi, D. M. Kaufman, S. Kamil, and M. M. Dehnavi. "NASOQ: Numerically Accurate Sparsity-Oriented QP Solver". In: *ACM Transactions on Graphics* 39.4 (July 2020). DOI: 10.1145/3386569.3392486.

[234] C. Clason, S. Mazurenko, and T. Valkonen. "Primal-dual proximal splitting and generalized conjugation in non-smooth non-convex optimization". In: *Applied Mathematics & Optimization* (Apr. 2020). DOI: 10.1007/s00245-020-09676-1.

[235] A. De Marchi. "Constrained and Sparse Switching Times Optimization via Augmented Lagrangian Proximal Methods". In: *2020 American Control Conference (ACC)*. Denver, CO, USA: IEEE, 2020, pp. 3633–3638. DOI: 10.23919/ACC45564.2020.9147892.

[236] A. De Marchi. *On a Primal-Dual Newton Proximal Method for Convex Quadratic Programs*. Submitted. Dec. 2020. DOI: 10.13140/RG.2.2.33215.12964.

[237] A. De Marchi and M. Gerdts. "Sparse Switching Times Optimization and a Sweeping Hessian Proximal Method". In: *Operations Research Proceedings 2019*. Ed. by J. S. Neufeld, U. Buscher, R. Lasch, D. Möst, and J. Schönberger. Cham: Springer, 2020, pp. 89–95. DOI: 10.1007/978-3-030-48439-2_11.

[238] P. E. Gill, V. Kungurtsev, and D. P. Robinson. "A Shifted Primal-Dual Penalty-Barrier Method for Nonlinear Optimization". In: *SIAM Journal on Optimization* 30.2 (2020), pp. 1067–1093. DOI: 10.1137/19M1247425.

[239] B. Hermans, A. Themelis, and P. Patrinos. *QPALM: A Proximal Augmented Lagrangian Method for Nonconvex Quadratic Programs*. Oct. 2020. arXiv: 2010.02653 [math.OC].

[240] T. Hoheisel, B. Pablos, A. Pooladian, A. Schwartz, and L. Steverango. "A Study of One-Parameter Regularization Methods for Mathematical Programs with Vanishing Constraints". In: *Optimization Methods and Software* (July 2020), pp. 1–43. DOI: 10.1080/10556788.2020.1797025.

[241] C. Kirches, F. Lenders, and P. Manns. "Approximation Properties and Tight Bounds for Constrained Mixed-Integer Optimal Control". In: *SIAM Journal on Control and Optimization* 58.3 (2020), pp. 1371–1402. DOI: 10.1137/18M1182917.

[242] S. Leyffer and C. Vanaret. "An augmented Lagrangian filter method". In: *Mathematical Methods of Operations Research* (June 2020). DOI: 10.1007/s00186-020-00713-x.

[243] D. Liao-McPherson and I. Kolmanovsky. "FBstab: A proximally stabilized semismooth algorithm for convex quadratic programming". In: *Automatica* 113 (2020), p. 108801. DOI: 10.1016/j.automatica.2019.108801.

[244] R. Loxton, Q. Lin, F. Padula, and L. Ntogramatzidis. "Minimizing control volatility for nonlinear systems with smooth piecewise-quadratic input signals". In: *Systems & Control Letters* 145 (2020), p. 104797. DOI: 10.1016/j.sysconle.2020.104797.

[245] A. Montoison and D. Orban. *TriCG and TriMR: Two Iterative Methods for Symmetric Quasi-Definite Systems*. Tech. rep. G-2020-41. Montréal, QC, Canada: GERAD, Aug. 2020. DOI: 10.13140/RG.2.2.12344.16645.

[246] D. Orban, A. S. Siqueira, and contributors. *CUTEst.jl: Julia's CUTEst Interface*. Oct. 2020. DOI: 10.5281/zenodo.3984328.

[247] D. Orban, A. S. Siqueira, and contributors. *NLPModels.jl: Data Structures for Optimization Models*. July 2020. DOI: 10.5281/zenodo.2558627.

[248] D. Orban, A. S. Siqueira, and contributors. *NLPModelsIpopt.jl: A thin IPOPT wrapper for NLPModels*. July 2020. DOI: 10.5281/zenodo.2629034.

[249] D. Orban and A. Soares Siqueira. "A regularization method for constrained nonlinear least squares". In: *Computational Optimization and Applications* 76.3 (June 2020), pp. 961–989. DOI: 10.1007/s10589-020-00201-2.

[250] S. Pougkakiotis and J. Gondzio. "An interior point-proximal method of multipliers for convex quadratic programming". In: *Computational Optimization and Applications* (Nov. 2020). DOI: 10.1007/s10589-020-00240-9.

[251] S. Sager and M. Tetschke. *Properties of time transformed mixed-integer optimal control problems*. Submitted. Mar. 2020. URL: http://www.optimization-online.org/DB_FILE/2020/03/7698.pdf.

[252] L. Stella, N. Antonello, M. Fält, D. Volodin, D. Herceg, E. Saba, F. B. Carlson, T. Kelman, P. Sopasakis, and P. Patrinos. *ProximalOperators.jl*. Version v0.12.0. Sept. 2020. DOI: 10.5281/zenodo.4020559.

[253] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. "OSQP: An Operator Splitting Solver for Quadratic Programs". In: *Mathematical Programming Computation* (Feb. 2020). DOI: 10.1007/s12532-020-00179-2.

[254] The MathWorks, Inc. *MATLAB Release 2020b*. Natick, MA, United States. 2020. URL: www.mathworks.com.

[255] A. Themelis and P. Patrinos. "Douglas–Rachford Splitting and ADMM for Nonconvex Optimization: Tight Convergence Results". In: *SIAM Journal on Optimization* 30.1 (2020), pp. 149–181. DOI: 10.1137/18M1163993.

[256] A. De Marchi. *QPDO: the Quadratic Primal-Dual Optimizer*. 2021. URL: https://github.com/aldma/qpdo.

[257] P. Sopasakis, E. Fresk, and P. Patrinos. "OpEn: Code Generation for Embedded Nonconvex Optimization". In: *IFAC-PapersOnLine* 53.2 (Apr. 2021), pp. 6548–6554. DOI: 10.1016/j.ifacol.2020.12.071.

[258] G. Chierchia, E. Chouzenoux, P. L. Combettes, and J.-C. Pesquet. *The Proximity Operator Repository*. (Accessed June 2021). URL: http://proximity-operator.net.

[259] H. D. Mittelmann. *Benchmarks for Optimization Software*. (Accessed June 2021). URL: http://plato.asu.edu/bench.html.

List of Acronyms

ADMM	Alternating Direction Method of Multipliers
AKKT	Approximate Karush-Kuhn-Tucker
AL	Augmented Lagrangian
ALP	Augmented Lagrangian Proximal
CIA	Combinatorial Integral Approximation
CPET	Control Parametrization Enhancing Technique
FBE	Forward-Backward Envelope
FBS	Forward-Backward Splitting
IP	Interior Point
KKT	Karush-Kuhn-Tucker
LICQ	Linear Independence Constraint Qualification
MIOCP	Mixed-Integer Optimal Control Problem
MPVC	Mathematical Program with Vanishing Constraints
NCP	Nonlinear Complementarity Problem
NCSP	Nonlinear, Nonconvex, Nonsmooth Constrained Structured Programming
NLP	Nonlinear Programming
OCP	Optimal Control Problem
PAL	Proximal Augmented Lagrangian
PM	Proximal Methods
PMM	Proximal Method of Multipliers
QP	Quadratic Programming
QPVC	Quadratic Program with Vanishing Constraints
SQP	Sequential Quadratic Programming
STO	Switching Time Optimization
VTT	Variable Time Transformation