Springer Optimization and Its Applications 76

Altannar Chinchuluun
Panos M. Pardalos
Rentsen Enkhbat
Efstratios N. Pistikopoulos *Editors*

Optimization, Simulation, and Control

Springer

Springer Optimization and Its Applications

VOLUME 76

Aims and Scope

Optimization has been expanding in all directions at an astonishing rate during the last few decades. New algorithmic and theoretical techniques have been developed, the diffusion into other disciplines has proceeded at a rapid pace, and our knowledge of all aspects of the field has grown even more profound. At the same time, one of the most striking trends in optimization is the constantly increasing emphasis on the interdisciplinary nature of the field. Optimization has been a basic tool in all areas of applied mathematics, engineering, medicine, economics, and other sciences.

The series *Springer Optimization and Its Applications* publishes undergraduate and graduate textbooks, monographs and state-of-the-art expository work that focus on algorithms for solving optimization problems and also study applications involving such problems. Some of the topics covered include nonlinear optimization (convex and nonconvex), network flow problems, stochastic optimization, optimal control, discrete optimization, multi-objective programming, description of software packages, approximation techniques and heuristic approaches.

For further volumes:
http://www.springer.com/series/7393

Altannar Chinchuluun • Panos M. Pardalos
Rentsen Enkhbat • Efstratios N. Pistikopoulos
Editors

Optimization, Simulation, and Control

Springer

Editors
Altannar Chinchuluun
Institute of Mathematics
National University of Mongolia
Ulaanbaatar, Mongolia

Rentsen Enkhbat
School of Economic Studies
National University of Mongolia
Ulaanbaatar, Mongolia

Panos M. Pardalos
Department of Industrial and Systems Engineering
University of Florida
Gainesville, FL, USA

Efstratios N. Pistikopoulos
Centre for Process Systems Engineering
Department of Chemical Engineering
Imperial College London
London, UK

ISSN 1931-6828
ISBN 978-1-4614-5130-3 ISBN 978-1-4614-5131-0 (eBook)
DOI 10.1007/978-1-4614-5131-0
Springer New York Heidelberg Dordrecht London

Library of Congress Control Number: 2012951286

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Mathematics is the language with which God wrote the universe

—Galileo Galilei

Preface

Optimization and optimal control are very powerful tools in engineering and applied mathematics. Problems in the fields are derived from real-world applications in finance, economics, telecommunications, and many other fields. There have been major algorithmic and theoretical developments in the fields of optimization and optimal control during the last decade. Lately simulation-based optimization methods are becoming a very popular approach for solving optimization problems due to developments of computer hardware. This book brings together recent developments in these areas as well as recent applications of these results to real-world problems. This book is aimed at both practitioners and academics and assumes that the reader has appropriate background in the above fields. The book consists of 21 chapters contributed by experts around the world who work with optimization, control, simulation, and numerical analysis.

The first eight chapters of the book are concerned with optimization theory and algorithms.

The spatial branch-and-bound algorithm for solving mixed-integer nonlinear programming problems uses convex relaxations for multilinear terms by applying associativity. The chapter by Belotti et al. gives two different convex relaxations using associativity in different ways and proves that having fewer groupings of longer terms yields tighter convex relaxations. Numerical examples show the efficiency of the algorithm.

The gap functions are generally used to investigate variational inequality problems through optimization problems. The chapter by Altangerel and Wanka investigates properties of gap functions for vector variational inequalities using the oriented distance functions. Enkhbat and Barsbold study the problem of finding two largest inscribed balls in a polyhedron so that sum of their radiuses is maximized. The problem is formulated as a bilevel programming problem and a gradient-based method is proposed to solve the program. The chapter by Tseveendorj gives a short survey on the theoretical and algorithmic results for mathematical programs with equilibrium constraints which have many applications in telecommunication and transportation networks, economical modeling, and computational mechanics.

Many real-world optimization design problems contain uncertainties which are characterized as parameters. The chapter by Kao and Liu studies linear programming problems with interval-valued parameters. For linear programs, the objective value is also interval-valued. They formulate two bilevel programs to calculate the lower and upper bounds of the objective values of the interval linear programs. These bilevel programs are then reduced to a single-level nonlinear program which can be tackled by standard nonlinear algorithms.

In order to solve optimization or control problems, existing simulation model or tools can be used. However, the transition is not simple. Here simulation means solving a system of state equations by a fixed-point iteration. The chapter by Griewank et al. quantifies and estimates the complexity of an optimization run compared to that of a single simulation, measured in terms of contraction rates. The chapter by Majig et al. considers the generalized Nash equilibrium problem. The problem can be formulated as a quasi-variational inequality problem. Using this reformulation, they propose a method for finding multiple, hopefully all, solutions to the generalized Nash equilibrium problem. Numerical experiments are provided to show the efficiency of the proposed approach. The chapter by Lorenz and Wanka studies scalar and vector optimization problems with objective functions, which consist of a convex function and a linear mapping and cone and geometric constraints. They formulate dual problems and establish weak, strong, and converse duality results between the dual and original programs.

Network optimization is one of the main fields of optimization and has many real-world applications. The next two chapters are concerned with network optimization problems and their applications in telecommunication. The minimum connected dominating set problem has a wide range of applications in wireless sensor networks and it gives an efficient virtual backbone for routing protocols. However, in some real-world problems, routing paths between pairs of vertices might be greater than the shortest path between them. In that case, minimum routing cost connected dominating set (MOC-CDS) is applied. The chapter by Liu et al. considers a variation of the MOC-CDS in the graph so-called g-MOC-CDS. They also propose a polynomial-time approximation scheme for the problem. The chapter by Charalambous studies some distributed power control algorithms for wireless ad hoc networks and discusses their convergence under uncertainties. The chapter also suggests directions for future research in the field.

The next four chapters are concerned with direct and indirect applications of optimization.

Nowadays, urban planning has been very critical for the development of many world cities. In their chapter, Keirstead and Shah model urban planning using optimization framework. The chapter by Enkhbat and Bayanjargal studies an extension of the classical Solow growth theory where the production function is an arbitrary continuous differentiable function and the saving and depreciation rates depend on time. The per capita consumption problem is reduced to a parametric maximization problem and a finite method for the problem is proposed. The chapter by Asada studies the existence of cyclical fluctuations in continuous time dynamic optimization models with two state variables. The results are applied to a continuous

time dynamic optimization economic model. The chapter by Lippe focuses on modeling and optimizing fuzzy-rule-based expert systems. It gives an overview of existing methods that combine fuzzy-rule-based systems with neural networks and presents a new tool for modeling an existing fuzzy-rule-based system using an artificial neural network.

The next four chapters are concerned with optimal control and its applications. The chapter by Gao and Baoyin introduces a smoothing technique for solving bang-bang optimal control problems. In order to speed up the convergence of this algorithm, an integration switching method based on a termed homotopy method is applied. They also provide some numerical examples illustrating the effectiveness of their method.

There are many methods for solving optimization and optimal control problems. However, it is hard to select the best approach for specific problems. The paper by Gornov et al. discusses and provides a set of optimal control problems that can be used to test the efficiency of different algorithms.

Lately parallel computing has been widely used to tackle real-world large-sized problems. The paper by Tyatushkin gives an algorithm for solving optimal control problems in the form of parallel computing. The algorithm uses a sequence of different methods in order to obtain fast convergence to an optimal solution. The chapter by Gornov and Zarodnyuk proposes an algorithm for finding global extremum of nonlinear and nonconvex optimal control problems. The method uses a curvilinear search technique to implement the tunneling phase of the algorithm. Numerical examples are presented to describe the efficiency of the proposed approach.

It is important to note that many optimization and optimal control algorithms require using methods in numerical analysis. The remaining three chapters are concerned with methods for solving system of linear equations, nonlinear equations, and differential equations. The chapter by Garloff et al. gives a survey on methods for finding the enclosure of the solution set of a system of linear equations, where the coefficients of the matrix and the right-hand side depend on parameters. Based on the methods, the chapter presents a hybrid method for the problem when the dependency is polynomial. The chapter by Bouhamidi and Jbilou proposes a new method for solving stiff ordinary differential equations using block Krylov iterative method. Some numerical examples are given to illustrate the efficiency of the proposed method. The chapter by Tugal and Dashdondog considers modifications of the Chebyshev method for solving nonlinear equations that are free from second derivative and prove semilocal convergence theorems for the methods.

We would like to take this opportunity to thank the authors of the chapters, the anonymous referees, and Springer for making the publication of this book possible.

Ulaanbaatar, Mongolia — Altannar Chinchuluun
Gainesville, FL, USA — Panos M. Pardalos
Ulaanbaatar, Mongolia — Rentsen Enkhbat
London, UK — Efstratios N. Pistikopoulos

Contents

On the Composition of Convex Envelopes for Quadrilinear Terms

Pietro Belotti, Sonia Cafieri, Jon Lee, Leo Liberti, and Andrew J. Miller

Abstract Within the framework of the spatial Branch-and-Bound algorithm for solving mixed-integer nonlinear programs, different convex relaxations can be obtained for multilinear terms by applying associativity in different ways. The two groupings $((x_1x_2)x_3)x_4$ and $(x_1x_2x_3)x_4$ of a quadrilinear term, for example, give rise to two different convex relaxations. In Cafieri et al. (J Global Optim 47:661–685, 2010) we prove that having fewer groupings of longer terms yields tighter convex relaxations. In this chapter we give an alternative proof of the same fact and perform a computational study to assess the impact of the tightened convex relaxation in a spatial Branch-and-Bound setting.

Key words Quadrilinear • Convex relaxation • Reformulation • Global optimization • Spatial Branch-and-Bound • MINLP

P. Belotti
Department of Mathematical Sciences, Clemson University, Clemson, SC 29634, USA
e-mail: pbelott@clemson.edu

S. Cafieri (✉)
Laboratoire MAIAA, Ecole Nationale de l'Aviation Civile, 31055 Toulouse, France
e-mail: sonia.cafieri@enac.fr

J. Lee
Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109-2117 USA
e-mail: jonxlee@umich.edu

L. Liberti
LIX, École Polytechnique, 91128 Palaiseau, France
e-mail: liberti@lix.polytechnique.fr

A.J. Miller
Institut de Mathématiques de Bordeaux, Université Bordeaux 1; RealOpt, INRIA Bordeaux Sud-Ouest, France
e-mail: andrew.miller@math.u-bordeaux1.fr

A. Chinchuluun et al. (eds.), *Optimization, Simulation, and Control*, Springer Optimization and Its Applications 76, DOI 10.1007/978-1-4614-5131-0_1,

1 Introduction

One of the most crucial steps of the spatial Branch-and-Bound algorithm for solving mixed-integer nonlinear programming (MINLP) problems is the lower bound computation. When the MINLP is factorable, it is possible to construct a convex relaxation automatically by means of a particular type of lifting reformulation (called MINLP standard form [10, 27]) first proposed in [16] and then exploited in most existing sBB algorithms [1,5,9,22,27,31]. If we consider polynomial problems, higher-order monomials are recursively rewritten as products of monomials of sufficiently low order for which a tight convex relaxation (possibly the convex envelope) is known. Each lower-order monomial is replaced by an added variable, and an equality constraint defining the added variable in terms of the monomial it replaces is adjoined to the MINLP. This operation is carried out recursively until the MINLP consists of a linear objective, some linear constraints, and several *defining constraints* of the form $w_j = h_j(x,w)$ for all j in some appropriate set J, where the functions h_j represent monomials. To obtain a convex relaxation, each defining constraint is replaced by a set of constraints defining the convex relaxation of its feasible set, thus yielding a convex relaxation for the whole problem.

Let $B = [x^L, x^U]$. The quadrilinear feasible set $S^4 = \{(w_1,x_1,x_2,x_3,x_4) \mid w_1 = x_1x_2x_3x_4\} \cap B$ over a box can be lifted in many different ways according to the way associativity is applied: the grouping $((x_1x_2)x_3)x_4$, for example, yields the set $S^{2,2,2} = \{(w_1,w_2,w_3,x_1,x_2,x_3,x_4) \mid w_2 = x_1x_2 \wedge w_3 = w_2x_3 \wedge w_1 = w_3x_4\} \cap B$, whereas the grouping $(x_1,x_2,x_3)x_4$ yields $S^{3,2} = \{(w_1,w_2,x_1,x_2,x_3,x_4) \mid w_2 = x_1x_2x_3 \wedge w_1 = w_2x_4\} \cap B$. Since convex/concave envelopes exist in explicit form for both bilinear [2, 16] and trilinear terms [17, 18], we can derive two different convex relaxations of S^4. The first, $\bar{S}^{2,2,2}$, consists in replacing the bilinear constraints $w_i = x_jx_k$ appearing in $S^{2,2,2}$ by the corresponding bilinear envelopes. The second, $\bar{S}^{3,2}$, consists in replacing the trilinear terms with the trilinear envelope and the bilinear term with the bilinear envelope. A question then arises naturally: which one is tighter?

In [6] we proved that $\bar{S}^{3,2} \subseteq \bar{S}^{2,2,2}$ and performed a computational study of the containment of the convex relaxations when different parameters were varied. In this chapter we provide an alternative proof (based on formal grammars) of the same result, and then test the impact of the tightened convex relaxation $\bar{S}^{3,2}$ using sBB.

The rest of this chapter is organized as follows. In Sect. 2 we present the main motivations of this work and a literature review on convex relaxations for multilinear monomials and their impact on a sBB algorithm. In Sect. 3 we propose a theoretical framework, based on concepts from the formal languages theory, to compare convex relaxations of multilinear monomials obtained as a composition of convex envelopes of lower-degree monomials. In Sect. 4 we discuss some computational experiments aimed at comparing different convex relaxations of quadrilinear terms in a spatial Branch-and-Bound setting. Concluding remarks are given in Sect. 5.

2 Motivation and Literature

The above discussion implies that deriving convex relaxations that are as strong as possible (i.e., that approximate the convex hull as closely as possible) for multilinear monomials can be critically important for the performance of a spatial Branch-and-Bound algorithm designed to globally solve nonconvex polynomial optimization problems. Because of this, numerous efforts have studied the convex hulls of sets defined by lower-order product terms and the use of these convex hulls in recursively factorized formulations (such as the MINLP standard form defined above).

Four valid inequalities for the three-dimensional set $S^2 = \{w, x_1, x_2 : w = x_1 x_2, x \in [x^L, x^U]\}$ were proposed by McCormick [16], and later Al-Khayyal and Falk [2] showed that these four inequalities suffice to describe the convex hull. At present most global MINLP solvers that use general sBB methods (among recent examples see [5, 14, 23]) use the convex hull for recursively defined instances of S^2 to define the polyhedral relaxations that are solved at each node of the Branch-and-Bound tree.

However, it may be thought that limiting solvers to the use of envelopes defined by simple bilinear terms may result in convex approximations for the original problem that are less strong (perhaps much less so) than those that exploit envelopes for more complex expressions. For problems involving multilinear multinomials defined by products of more than two variables, this consideration has motivated research into the envelopes of *trilinear* functions [17, 18]. Comparing the use of convex envelopes for bilinear and trilinear forms in building convex approximations for MINLPs motivated the study in [6], and comparisons involving more general functional forms motivate the present article.

Bi- and trilinear functions are naturally generalized to functions with *vertex polyhedral* convex envelopes. (The convex envelope of an n-dimensional function $f(x)$ is said to be vertex polyhedral if its domain X is a polyhedron, and if every extreme point of the convex hull of $\{(x, f(x) : x \in X\}$ is defined by an extreme point of X itself). In [19] Meyer and Floudas generalized the approach developed for trilinear functions to functions with vertex polyhedral convex envelopes. Essentially, their approaches can be thought of as enumerative methods that consider all possible combinations of $n+1$ extreme points of X (equivalently, extreme points of $\text{conv}(\{(x, f(x)) : x \in X\})$) and then establish conditions under which the hyperplane defined by such a set of points defines a linear inequality satisfied by all the other extreme points of $\text{conv}(\{(x, f(x)) : x \in X\})$. Such an inequality is then valid for $\{(x, f(x)) : x \in X\}$ and facet defining for the convex hull of this set.

General multilinear functions (i.e., any function composed of a sum of products of variables, in which the degree of each variable in each product is 0 or 1) were shown to have vertex polyhedral convex envelopes by Rikun [21]. An implication of this result is that many of the concepts mentioned in the preceding paragraph can be used for general mutilinear functions; their use is not limited to monomial

products (for example). The extension of such results to define convex envelopes for multilinear functions (and generalizations of them) has been discussed in [26, 28–30], among other references.

Empirical testing of the approaches mentioned above (beyond the use of bilinear envelopes defined by McCormick [16]) has been limited, but recently authors have begun exploiting some of these concepts to solve quadratically constrained quadratic programs, in which sums of bilinear products often figure prominently. In particular, the authors of [4] discuss how to dynamically generate facets of the convex hull of the sum of bilinear products in order to define a stronger relaxation of the original MINLP, and they report that strengthening the formulation with such inequalities can significantly improve the performance of BARON [23], which by default uses only McCormick envelopes to exploit multilinear terms in defining convex relaxations. Even more recently, Luedtke et al. [15] provide rigorous bounds for how much the approach of [23] (and, implicitly, of [26]) can strengthen the relaxations defined by the use of McCormick envelopes, and also provide numerical results illustrating that these bounds are tight.

It is important to note that the bounds defined by Luedtke [15] apply only to problems that have sums of bilinear products but not quadratic terms (i.e., if the quadratic function in a given constraint is represented by $f_Q(x) = x^T Qx$, the bounds defined in [15] are valid for problems in which the diagonal elements of Q are all 0). Moreover, computational experience seems to confirm that the smaller the absolute values of elements on the diagonal of Q are in comparison to the off-diagonal elements, the more important the role played by strong convex relaxations for bilinear functions becomes in defining strong relaxations for the MINLP. (Defining effective relaxations for nonconvex quadratically constrained problems in which the diagonal elements of Q are large requires, in addition to the techniques described in this section, other methods that are fundamentally different. References that discuss solving nonconvex quadratically constrained problems with large diagonal absolute values include [3, 4, 8, 24, 25] and the references contained therein.)

An unresolved issue that is directly related to much of the research on multilinear functions described above is the question of whether or not it is possible to define a description of the convex envelope of multilinear functions that does not require the explicit a priori enumeration of all of the extreme points of the domain. More formally, given an n-dimensional function $f(x) = \prod_{i=1}^{n} x_i$ over a domain $B = [x^L, x^U]$, is it possible to define a set of criteria that (1) each facet of the convex envelope must satisfy and (2) can be checked in time polynomial in n? Most of the approaches described above, as well as the motivation of this article, are based on the implicit assumption that the answer to this question is no. However, only a comparatively small number of research efforts (e.g., [15, 26]) have addressed this question directly. Moreover, their consideration of this question has been limited to establishing criteria for x^L and x^U that are sufficient to guarantee that the answer is yes.

Computational complexity theory, and in particular results of [7] suggest that a short (i.e., polynomial in n) description of the convex envelopes of multilinear

functions can be defined if and only if the following optimization problem is polynomial solvable:

$$\min \prod_{i=1}^{n} x_i - \sum_{i=1}^{n} c_i x_i \tag{1}$$

$$\text{s.t. } x_i^L \leq x_i \leq x_i^U, i = 1, \ldots, n, \tag{2}$$

where $c \in \mathbb{R}^n$ is some rational vector. It seems that this problem is likely to be *NP*-complete unless fairly restrictive assumptions on x^L and x^U are satisfied. For example, generalizing some of the results of [15], in [20] the authors show that it is possible to solve the above optimization problem in polynomial time if there exists a constant $a < 1$ such that $ax_i^L = x_i^U$ for $i = 1, \ldots, n$. It is also clear that slightly more general conditions can be established. However, the authors of [20] conjecture that the above optimization problem is *NP*-complete in general and the complexity of this problem remains an important open question in the area of how best to approximate the convex envelopes of functions involving multilinear terms.

We will next turn to the general question of when, and how, one approach to defining convex relaxations of factorable functions can be shown to yield relaxations that are stronger than those generated by another approach. The primary contribution of this article is to establish a general result concerning this issue. We should perhaps first note, however, that this contribution does not tell us how much stronger the dominant formulation will be; this is necessarily an empirical question. Moreover, the comparative ease with which different relaxations can be solved is also a necessarily empirical criterion, and in general both of these considerations must be weighed in considering relaxation to use in a given situation.

3 The Composition of Convex Envelopes

In this section we prove that a stronger relaxation is obtained when one replaces "large terms" with tight convex relaxations instead of breaking up such terms in sums/products of smaller terms before replacing each small term with its respective convex relaxation. Although we find that this is quite an intuitive result, because of the inherently recursive nature of factorable functions and of the fact that we deal with a recursive symbolic procedure for constructing the convex relaxation, we did not find it easy to prove this result formally. For this purpose, we use theoretical tools that are well known to the formal languages community but perhaps not so commonly found in the optimization literature: this is why we detail every step and attempt to be somewhat didactical in presentation, alternating formal statements to informal explanations and examples. To the well versed in such matters, a brief glimpse to the section might suffice to understand our strategy: assign a special semantic value (the corresponding convex relaxation) to each operator node of

an expression tree, define the semantics of the composition operator, and finally compare the resulting relaxation with the tight convex relaxation given for the composite operator at "atomic" level.

3.1 Alphabets, Languages, and Grammars

An *alphabet* $\mathscr{A}$ is a set of symbols. We let $\mathscr{A}^*$ be the set of all finite sequences of elements of $\mathscr{A}$. A *formal language* $\mathscr{L}$ is a subset of $\mathscr{A}^*$. A language $\mathscr{L}$ is *decidable* if, given a string $s \in \mathscr{A}^*$, there exists a finite algorithmic procedure that decides whether $s \in \mathscr{L}$ or not.

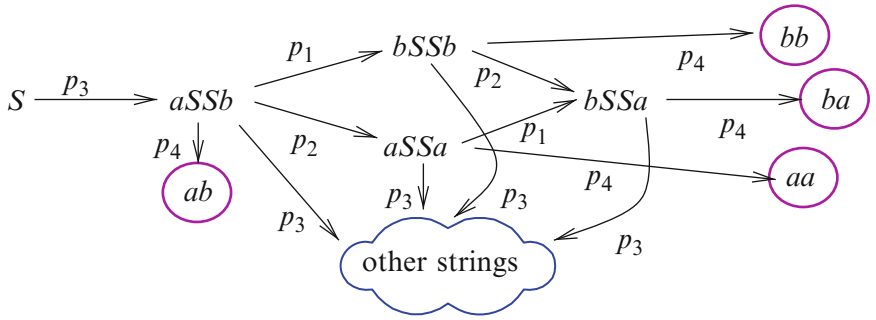
Informally, decidability of a language is concerned with its syntax: is a string a valid element of the language or not? Having decided what a language is, we have to decide what it says: to every string there corresponds a semantic value, which, in the theory and language of Zermelo–Fraenkel, is usually a set. In this setting, our formal language is the set of all valid functions $f(x)$ that can be written as finite strings of symbols in infix notation. The semantic values assigned to $f(x)$ are sets such as $\{(w,x) \in \mathbb{R}^{n+1} \mid w = f(x) \wedge x^L \leq x \leq x^U\}$ (exact semantics) and $\{(w,x) \in \mathbb{R}^{n+1} \mid w \in R(f,x^L,x^U) \wedge x^L \leq x \leq x^U\}$ (relaxed semantics) where $R(f,x^L,x^U)$ is a convex relaxation of the exact semantics. Since the cardinality of our language is countably infinite, we cannot explicitly assign exact/relaxed semantics to each function in the language. Instead, we recall that a decidable language has finite procedure for recognizing strings in the language: for each of the (finitely many) operations specified by this procedure we define a corresponding operation on the semantic values involved, thus obtaining a semantic definition for the whole language.

To this effect, we make use of possibly the best known device for specifying the syntax of a formal language $\mathscr{L}$, i.e., a *formal grammar*. This is a quadruplet $\Gamma = (\Sigma, N, P, S)$ such that:

- $\Sigma \subseteq \mathscr{A}$ is the set of *terminal symbols*.
- N is a set of *nonterminal symbols* ($N \cap \Sigma = \emptyset$).
- P is a set of *rewriting* or *production rules* ($P \subseteq (\Sigma \cup N)^* N (\Sigma \cup N)^* \to (\Sigma \cup N)^*$).
- $S \in N$ is the *start symbol*.

In practice, one recursively applies the production rules to the start symbol as many times as possible, generating strings in $(\Sigma \cup N)^*$. Those generated strings that are in Σ^* are strings of the language $\mathscr{L}$. If a string in $\mathscr{A}^*$ is not in the set of all strings in Σ^* that the grammar generates, then it is not in $\mathscr{L}$.

Example 1. Consider the alphabet $\{a,b\}$ and the grammar given by $N = \{S\}$ where S is the start symbol, $\Sigma = \{a,b\}$ and the production rules $\langle p_1 = aS \to bS, p_2 = Sb \to Sa, p_3 = S \to aSSb, p_4 = SS \to \emptyset \rangle$. We repeatedly apply $p_1, \ldots, p_4$ to the start symbol, obtaining the situation below:



From this, we conclude that aa, ab, ba, bb are in the language specified by the grammar. It must be remarked that formal grammars can also be given for languages which are not decidable (e.g., if the recursion does not terminate); this is one such grammar: the repeated application of p_3 yields longer and longer strings all involving the nonterminal symbol S.

3.2 Mathematical Expression Language: Syntax

We now formally define our function language through the use of a formal grammar. We use an alphabet $\mathscr{A} = \mathbb{X} \cup \mathbb{K} \cup \mathbb{B} \cup \mathbb{O}$ where $\mathbb{X} = \{x_1, \ldots, x_n\}$ is the set of symbols denoting original variables, $\mathbb{K}$ is the set of all computable numbers, $\mathbb{B} = \{\text{`(', `)'}\}$, and $\mathbb{O}$ is a finite set of operators $\{+, -, \times, \div, \uparrow, \sqrt{}, \log, \exp, \sin, \cos, \tan\}$, where $+, \times$ are binary operators, $-$ can be unary or binary, and $\uparrow$ is the (binary) power operator. The grammar Γ is defined as follows. The start symbol is $\mathscr{F}$, $N = \{\mathscr{F}\}$, $\Sigma = \mathscr{A}$, and P is

$$\mathscr{F} \longrightarrow x_i \in \mathbb{X} \quad (3) \qquad \mathscr{F} \longrightarrow \cos(\mathscr{F}) \quad (10)$$

$$\mathscr{F} \longrightarrow k \in \mathbb{K} \quad (4) \qquad \mathscr{F} \longrightarrow \tan(\mathscr{F}) \quad (11)$$

$$\mathscr{F} \longrightarrow (\mathscr{F}) \quad (5) \qquad \mathscr{F} \longrightarrow (\mathscr{F} - \mathscr{F}) \quad (12)$$

$$\mathscr{F} \longrightarrow (-\mathscr{F}) \quad (6) \qquad \mathscr{F} \longrightarrow (\mathscr{F} \div \mathscr{F}) \quad (13)$$

$$\mathscr{F} \longrightarrow \log(\mathscr{F}) \quad (7) \qquad \mathscr{F} \longrightarrow (\mathscr{F} \uparrow \mathscr{F}) \quad (14)$$

$$\mathscr{F} \longrightarrow \exp(\mathscr{F}) \quad (8) \qquad \mathscr{F} \longrightarrow (\mathscr{F} + \mathscr{F}) \quad (15)$$

$$\mathscr{F} \longrightarrow \sin(\mathscr{F}) \quad (9) \qquad \mathscr{F} \longrightarrow (\mathscr{F} \times \mathscr{F}) \quad (16)$$

Notice that rules (3)–(4) are given in schematic form: i.e., the string on the left of the arrow is not in $(\Sigma \cup N)^*$, but it is possible to define "sublanguages" that decide whether a string is in $\mathbb{X}$ or in $\mathbb{K}$.

Example 2. In order to recognize that the string $F \equiv x_1 + ((x_2 \uparrow 2) + (x_3 \times (x_4 \times \log(x_1))))$ is in $\mathscr{L}$ we can apply the production rules as follows (there are other possible orders in which the rules can be applied yielding the same result):

$$
\begin{aligned}
\mathscr{F} &\to (\mathscr{F}+\mathscr{F}) && \text{by (15)}\\
&\to (\mathscr{F}+(\mathscr{F}+\mathscr{F})) && \text{by (15)}\\
&\to (\mathscr{F}+((\mathscr{F}\uparrow\mathscr{F})+\mathscr{F})) && \text{by (14)}\\
&\to (\mathscr{F}+((\mathscr{F}\uparrow\mathscr{F})+(\mathscr{F}\times\mathscr{F}))) && \text{by (16)}\\
&\to (\mathscr{F}+((\mathscr{F}\uparrow\mathscr{F})+(\mathscr{F}\times(\mathscr{F}\times\mathscr{F})))) && \text{by (16)}\\
&\to (\mathscr{F}+((\mathscr{F}\uparrow\mathscr{F})+(\mathscr{F}\times(\mathscr{F}\times\log(\mathscr{F}))))) && \text{by (7)}\\
&\to (x_1+((x_2\uparrow\mathscr{F})+(\mathscr{F}\times(\mathscr{F}\times\log(\mathscr{F}))))) && \text{by (3)}\\
&\to (x_1+((x_2\uparrow\mathscr{F})+(\mathscr{F}\times(\mathscr{F}\times\log(\mathscr{F}))))) && \text{by (3)}\\
&\to (x_1+((x_2\uparrow\mathscr{F})+(x_3\times(\mathscr{F}\times\log(\mathscr{F}))))) && \text{by (3)}\\
&\to (x_1+((x_2\uparrow\mathscr{F})+(x_3\times(x_4\times\log(\mathscr{F}))))) && \text{by (3)}\\
&\to (x_1+((x_2\uparrow\mathscr{F})+(x_3\times(x_4\times\log(x_1))))) && \text{by (3)}\\
&\to (x_1+((x_2\uparrow 2)+(x_3\times(x_4\times\log(x_1))))) && \text{by (4)}\\
&\to x_1+((x_2\uparrow 2)+(x_3\times(x_4\times\log(x_1)))) && \text{by (5).}
\end{aligned}
$$

We need to apply 13 rewriting rules in order to recognize that $F \in \mathscr{L}$.

3.3 Mathematical Expression Language: Semantics

We are now going to use the formal grammar Γ to assign semantic values to strings. Informally, we assign different sets to the different occurrences of the symbol $\mathscr{F}$ in each production rule, in such a way that the set assigned to $\mathscr{F}$ appearing in the left-hand side of each rule is defined in terms of the sets assigned to the symbols $\mathscr{F}$ appearing in the right-hand side. More precisely, for a production rule ρ in (3)–(16) of the form $\mathscr{F} \to T$, where $T \in (\Sigma \cup N)^*$, let $\nu(\rho)$ be the number of occurrences of the symbol $\mathscr{F}$ in the string T. Let $X_0(\rho)$ be the set assigned to the symbol $\mathscr{F}$ appearing on the left-hand side of ρ, and for all $i \in \{1,\ldots,\nu(\rho)\}$ let $X_i(\rho)$ be the set assigned to the ith occurrence of the symbol $\mathscr{F}$ in T.

3.3.1 Exact Semantics

The *exact semantics* of $\mathscr{L}$ is defined according to the following rules.

$$
\begin{aligned}
\mathscr{F} &\longrightarrow x_i \in \mathbb{X} && : X_0 = [x_i^L, x_i^U]\\
\mathscr{F} &\longrightarrow k \in \mathbb{K} && : X_0 = \{k\}\\
\mathscr{F} &\longrightarrow (\mathscr{F}) && : X_0 = X_1\\
\mathscr{F} &\longrightarrow (-\mathscr{F}) && : X_0 = \{(w,x) \mid w = -x \wedge x \in X_1\}\\
\mathscr{F} &\longrightarrow \log(\mathscr{F}) && : X_0 = \{(w,x) \mid w = \log(x) \wedge x \in X_1\}\\
\mathscr{F} &\longrightarrow \exp(\mathscr{F}) && : X_0 = \{(w,x) \mid w = \exp(x) \wedge x \in X_1\}\\
\mathscr{F} &\longrightarrow \sin(\mathscr{F}) && : X_0 = \{(w,x) \mid w = \sin(x) \wedge x \in X_1\}\\
\mathscr{F} &\longrightarrow \cos(\mathscr{F}) && : X_0 = \{(w,x) \mid w = \cos(x) \wedge x \in X_1\}\\
\mathscr{F} &\longrightarrow \tan(\mathscr{F}) && : X_0 = \{(w,x) \mid w = \tan(x) \wedge x \in X_1\}
\end{aligned}
$$

$$\begin{aligned}
\mathscr{F} &\longrightarrow (\mathscr{F}-\mathscr{F}) : X_0 = \{(w,x_1,x_2) \mid w = x_1 - x_2 \wedge \forall i \in \{1,2\}\, x_i \in X_i\}\\
\mathscr{F} &\longrightarrow (\mathscr{F}\div\mathscr{F}) : X_0 = \{(w,x_1,x_2) \mid w = x_1/x_2 \wedge \forall i \in \{1,2\}\, x_i \in X_i\}\\
\mathscr{F} &\longrightarrow (\mathscr{F}\uparrow\mathscr{F}) : X_0 = \{(w,x_1,x_2) \mid w = x_1^{x_2} \wedge \forall i \in \{1,2\}\, x_i \in X_i\}\\
\mathscr{F} &\longrightarrow (\mathscr{F}+\mathscr{F}) : X_0 = \{(w,x_1,x_2) \mid w = x_1 + x_2 \wedge \forall i \in \{1,2\}\, x_i \in X_i\}\\
\mathscr{F} &\longrightarrow (\mathscr{F}\times\mathscr{F}) : X_0 = \{(w,x_1,x_2) \mid w = x_1 x_2 \wedge \forall i \in \{1,2\}\, x_i \in X_i\}
\end{aligned}$$

A meta-linguistic note: the naming of the semantic values X_0, X_1, X_2 must be local to each rule. Otherwise, if the same rule ρ is applied twice, we might get two different definitions assigned to the same name $X_0(\rho)$. In order to obtain a consistent naming, we observe that the recursive nature of string recognition in $\mathscr{L}$ is finite, so the different strings of $(\Sigma \cup N)^*$ generated during the recognition procedure can be listed in the order of rewriting, as in Example 2. For a string $f \in \mathscr{L}$ let $r(f)$ be the length of this list. For all $k \leq r(f)$, we can now let X_0^k be the semantic value assigned to $\mathscr{F}$ appearing in the left-hand side of the production rule ρ being applied at the kth rewriting step, and let $X_1^k, \ldots, X_{\nu(\rho)}^k$ be the sets assigned to the various occurrences of $\mathscr{F}$ in the right-hand side of ρ.

As will appear clear in Example 3, some of the semantic sets will be projections of other semantic sets on some of their coordinates. For every semantic set X we shall therefore let $\mathscr{V}(X)$ be the sequence of variable symbols in terms of which X is defined (so that $X \subseteq \mathbb{R}^{|\mathscr{V}(X)|}$), and for all $W \subseteq \mathscr{V}(X)$ let $\pi(X,W)$ be the projection of X on the w coordinate (if $W = \{w\}$, we write $\pi(X,w)$).

Example 3. The exact semantics of F, as defined in Example 2, is derived as follows.

$$\begin{aligned}
X_0^1 &= \{(w_1,w_2,w_3) \mid w_1 = w_2 + w_3 \wedge w_2 \in X_1^1 \wedge w_3 \in X_2^1\}\\
X_0^2 &= \{(w_3,w_4,w_5) \mid w_3 = w_4 + w_5 \wedge w_4 \in X_1^2 \wedge w_5 \in X_2^2\} \text{ and } X_2^1 = \pi(X_0^2, w_3)\\
X_0^3 &= \{(w_4,w_6,w_7) \mid w_4 = w_6^{w_7} \wedge w_6 \in X_1^3 \wedge w_7 \in X_2^3\} \text{ and } X_1^2 = \pi(X_0^3, w_4)\\
X_0^4 &= \{(w_5,w_8,w_9) \mid w_5 = w_8 w_9 \wedge w_8 \in X_1^4 \wedge w_9 \in X_2^4\} \text{ and } X_2^2 = \pi(X_0^4, w_5)\\
X_0^5 &= \{(w_9,w_{10},w_{11}) \mid w_9 = w_{10} w_{11} \wedge w_{10} \in X_1^5 \wedge w_{11} \in X_2^5\} \text{ and } X_2^4 = \pi(X_0^5, w_9)\\
X_0^6 &= \{(w_{11},w_{12}) \mid w_{11} = \log(w_{12}) \wedge w_{12} \in X_1^6\} \text{ and } X_2^5 = \pi(X_0^6, w_{11})\\
X_0^7 &= [x_1^L, x_1^U] \text{ and } X_1^1 = X_0^7\\
X_0^8 &= [x_2^L, x_2^U] \text{ and } X_1^3 = X_0^8\\
X_0^9 &= [x_3^L, x_3^U] \text{ and } X_1^4 = X_0^9\\
X_0^{10} &= [x_4^L, x_4^U] \text{ and } X_1^5 = X_0^{10}\\
X_0^{11} &= [x_1^L, x_1^U] \text{ and } X_1^6 = X_0^{11}\\
X_0^{12} &= \{2\} \text{ and } X_2^3 = X_0^{12}\\
X_0^{13} &= X_0^1.
\end{aligned}$$

Replacing symbols where possible, we obtain a definition of the exact semantics of our string in function of only six sets and ten variables (four original variables and six added variables):

$$
\begin{aligned}
X_0^1 &= \{(w_1,x_1,w_3) \mid w_1 = x_1 + w_3 \wedge x_1 \in [x_1^L, x_1^U] \wedge w_3 \in \pi(X_0^2, w_3)\} \\
X_0^2 &= \{(w_3,w_4,w_5) \mid w_3 = w_4 + w_5 \wedge w_4 \in \pi(X_0^3, w_4) \wedge w_5 \in \pi(X_0^4, w_5)\} \\
X_0^3 &= \{(w_4,x_2) \mid w_4 = x_2^2 \wedge x_2 \in [x_2^L, x_2^U]\} \\
X_0^4 &= \{(w_5,x_3,w_9) \mid w_5 = w_8 w_9 \wedge x_3 \in [x_3^L, x_3^U] \wedge w_9 \in \pi(X_0^5, w_9)\} \\
X_0^5 &= \{(w_9,x_4,w_{11}) \mid w_9 = w_{10} w_{11} \wedge x_4 \in [x_4^L, x_4^U] \wedge w_{11} \in \pi(X_0^6, w_{11})\} \\
X_0^6 &= \{(w_{11},x_1) \mid w_{11} = \log(x_1) \wedge x_1 \in [x_1^L, x_1^U]\}.
\end{aligned}
$$

Suppose now we consider an enriched alphabet $\mathscr{A}'$ with one more 4-ary operator $\otimes$ such that $\otimes(x_1,\ldots,x_4) = x_1 + x_2^2 + x_3 x_4 \log(x_1)$ and an extended grammar with one more production rule $\rho' \equiv \mathscr{F} \longrightarrow \mathscr{F} + \mathscr{F} \uparrow 2 + \mathscr{F} \times \mathscr{F} \log(\mathscr{F})$. The generated language $\mathscr{L}$' is identical to $\mathscr{L}$ because we showed previously that $\mathscr{L}$ contains strings as that appearing in the right-hand side of ρ' even without the production rule ρ'. However, using the extended grammar, the string F can be recognized in only one step. By replacement of the appropriate variable symbols w_ℓ, the exact semantics $\{(w,x) \mid w = \otimes(x_1,\ldots,x_4) \wedge \forall i \leq 4 x_i \in [x_i^L, x_i^U]\}$ of F computed with the extended grammar is precisely the projection of X_0^1 on the subspace of $\mathbb{R}^{10}$ spanned by $(w_1,x_1,\ldots,x_4)$.

3.3.2 Relaxed Semantics

We now define the relaxed semantics of $\mathscr{L}$. Whereas in the exact semantics we assigned to each string the set of values taken by the corresponding function as its arguments range in the appropriate (recursively defined) sets, the relaxed semantics assigns to strings convex relaxations of such sets. To this end, we shall describe an operator $\mathscr{R}_\Gamma$ that computes the convex relaxation of a set using the composition of production rules in Γ. For each operator $\oplus \in \mathbb{O}$, let $\alpha(\oplus)$ be its arity (the number of its arguments). Denote $\alpha(\oplus)$ by ℓ, $\mathbb{I}$ the class of all closed and bounded intervals in $\mathbb{R}$, and let $I_1,\ldots,I_\ell \in \mathbb{I}$; then we use the notation $\mathscr{R}_\Gamma(\oplus, I_1,\ldots,I_\ell)$ to indicate a convex relaxation in $\mathbb{R}^{n+1}$ of the exact semantic value of $\oplus$, i.e., the set $\{(w_0,w_1,\ldots,w_\ell) \mid w_0 = \oplus(w_1,\ldots,w_\ell) \wedge \forall i \leq n\ (w_i \in I_i)\}$. We impose a consistency (monotonicity) requirement:

$$
\begin{aligned}
\forall \oplus \in \mathbb{O}, I_1,\ldots,I_\ell, J \in \mathbb{I} \ \text{ s.t. } \ & \exists i \leq \ell\ J \subseteq I_i \\
& \mathscr{R}_\Gamma(\oplus, I_1,\ldots,I_\ell) \ \supseteq \ \mathscr{R}_\Gamma(\oplus, I_1,\ldots,I_{i-1}, J, I_{i+1},\ldots,I_\ell)), \qquad (17)
\end{aligned}
$$

which means that convex relaxations should get tighter when the definition intervals get smaller.

We remark that $\mathscr{R}$ is a symbol in the metalanguage, in the sense that it should be replaced by an actual description of the convex sets assigned to each operator (in other words, it stands for the sentence "for all possible ways of defining convex relaxations of operators…"). A typical definition of $\mathscr{R}_\Gamma$ used by most sBB solver codes (e.g., ooOPS [13] and COUENNE [5], both based on a grammar very similar to Γ) is as follows: for all linear operators, $\mathscr{R}_\Gamma$ applied to that operator is the same as the exact semantics (because, as an affine space defined over a cartesian product of intervals, it is convex). The $\log, \exp$ operators are concave/convex univariate, and hence $\mathscr{R}_\Gamma$ is defined as a convex subset of $\mathbb{R}^2$ delimited by the function itself and the secant at the interval endpoints [9]; for piecewise convex/concave functions we employ the convex envelope defined in [12]; for trigonometric functions it is easy to work out convex relaxations/envelopes using secants and convex/concave portions of the functions themselves. We remark that providing convex/concave relaxations/envelopes of convex/concave functions and piecewise convex/concave functions suffices to define $\mathscr{R}_\Gamma$ over all univariate monomials of the form x^k where $x \in I \in \mathbb{I}$. For bilinear products, we employ the well-known McCormick envelopes:

$$\begin{aligned}\mathscr{R}(\times, [w_1^L, w_1^U], [w_2^L, w_2^U]) = \{(w_0, w_1, w_2) \mid \\ w_0 \geq w_1^L w_2 + w_2^L w_1 - w_1^L w_2^L \wedge \\ w_0 \geq w_1^U w_2 + w_2^U w_1 - w_1^U w_2^U \wedge \\ w_0 \leq w_1^L w_2 + w_2^U w_1 - w_1^L w_2^U \wedge \\ w_0 \leq w_1^U w_2 + w_2^L w_1 - w_1^U w_2^L) \wedge \\ w_1 \in [w_1^L, w_1^U] \wedge w_2 \in [w_2^L, w_2^U]\}.\end{aligned}$$

It is easy to check that the above definition of $\mathscr{R}$ satisfies (17).

The *relaxed semantics* of $\mathscr{L}$ is defined according to the rules:

$$\mathscr{F} \longrightarrow \oplus(\mathscr{F}, \ldots, \mathscr{F}) : X_0 = \mathscr{R}_\Gamma(\oplus, I_1, \ldots, I_{\alpha(\oplus)}).$$

Relaxed semantics can be combined following grammatic production rule composition in much the same way as exact semantics can, by noticing that when X is a convex subset of $\mathbb{R}^n$, the projection of X on one coordinate axis is always an interval (because projection preserves convexity).

Now let F be a valid string of $\mathscr{L}$: then F is a mathematical expression with, say, $x = (x_1, \ldots, x_n)$ as variable symbol arguments corresponding to a certain mathematical function $f : \mathbb{R}^n \to \mathbb{R}$. Then we can certainly add the following rule to Γ:

$$\rho' \equiv \mathscr{F} \longrightarrow F(\underbrace{\mathscr{F}, \ldots, \mathscr{F}}_{n}), \tag{18}$$

yielding an extended grammar Γ', and still obtain $\mathscr{L}$ as generated language. The advantage is that Γ' allows recognition of the string F in one step and assignment

of a special relaxed semantics to F (instead of relying on the composition of relaxed semantics of substrings of F through the production rules). This is useful for those operators which do not appear in the list of production rules but for which we have a tight convex relaxation (or a convex envelope).

3.4 Comparison of Relaxed Semantics

Let $F \in \mathscr{L}$ represent an n-ary function such that ρ', defined as in (18), is not a production rule of Γ. Define $\mathscr{A}'$ as $\mathscr{A} \cup \{F\}$ and Γ' as Γ with ρ' as an added production rule. Assume that the given relaxed semantics for F in Γ' is included in the computed relaxed semantics for F in Γ (which is usually the case in practice, for otherwise we would not add the "useless" rule ρ' to Γ'), i.e., that, for all $I_1, \ldots, I_\ell \in \mathbb{I}$,

$$\mathscr{R}_{\Gamma'}(F, I_1, \ldots, I_\ell) \subseteq \mathscr{R}_{\Gamma}(F, I_1, \ldots, I_\ell). \tag{19}$$

Theorem 1. *For all strings $T \in \mathscr{L}$ that are functions of p variable symbol arguments and for all $I_1, \ldots, I_p \in \mathbb{I}$, we have $\mathscr{R}_{\Gamma'}(T, I_1, \ldots, I_p) \subseteq \mathscr{R}_{\Gamma}(T, I_1, \ldots, I_p)$.*

Proof. If recognition of T through Γ' never involves rule ρ', both grammars yield the same relaxed semantics. Otherwise, consider the *last* time that ρ' is used on T: then Γ' matches a string $\mathscr{F}$ which is an operator F of n arguments. Let $J_1, \ldots, J_\ell$ be the relaxed semantics assigned to each of the n arguments. Since this is the last time ρ' is used, each of the J_i ($i \leq n$) is the same whether we use Γ or Γ', which means that, by (19), $J_{\Gamma'} = \mathscr{R}_{\Gamma'}(F, J_1, \ldots, J_\ell) \subseteq \mathscr{R}_{\Gamma}(F, J_1, \ldots, J_\ell) = J_\Gamma$. By (17), any relaxed semantics involving $J_{\Gamma'}$ will be contained in the same relaxed semantics with $J_{\Gamma'}$ replaced by J_Γ. Thus, if the statement holds from the $(k+1)$-st to the last time rule ρ' is used, the kth time ρ' is used the argument intervals of the relaxed semantics in Γ' must be contained in the argument intervals of the corresponding relaxed semantics in Γ. □

In particular, we have the following.

Corollary 1. *If $F(x_1, x_2, x_3) = x_1 x_2 x_3$ and we assign to F the relaxed semantics given by the trilinear envelopes given in [17, 18], the convex relaxation obtained through Γ' is at least as tight as that obtained through Γ for any mathematical function in $\mathscr{L}$.*

Proof. Assumption (19) holds by definition of convex envelope. □

4 Computational Results

In this section, we computationally evaluate the tightness of convex relaxations for quadrilinear monomials obtained combining bilinear and trilinear convex envelopes in different ways. Specifically, we consider relaxations of the following four sets:

$$S^{222} = \{(x,w) \in \mathbb{R}^4 \times \mathbb{R}^3 \,|\, x_i \in [x_i^L, x_i^U] \wedge w_1 = x_1x_2, w_2 = w_1x_3, w_3 = w_2x_4\},$$
$$\tilde{S}^{222} = \{(x,w) \in \mathbb{R}^4 \times \mathbb{R}^3 \,|\, x_i \in [x_i^L, x_i^U] \wedge w_1 = x_1x_2, w_2 = x_3x_4, w_3 = w_1w_2\},$$
$$S^{32} = \{(x,w) \in \mathbb{R}^4 \times \mathbb{R}^2 \,|\, x_i \in [x_i^L, x_i^U] \wedge w_1 = x_1x_2x_3, w_2 = w_1x_4\},$$
$$S^{23} = \{(x,w) \in \mathbb{R}^4 \times \mathbb{R}^2 \,|\, x_i \in [x_i^L, x_i^U] \wedge w_1 = x_1x_2, w_2 = w_1x_3x_4\}.$$

In [6] numerical experiments were carried out in order to evaluate the relative tightness of the four considered relaxations. The comparison was mainly made in terms of volume of the corresponding enveloping polytopes (projected onto $\mathbb{R}^5$ to have comparable results) on a set of randomly generated instances. It showed that the smallest values of volumes correspond to relaxations involving the composition of trilinear and bilinear envelopes, and in particular the best results for more than 80 % of the considered instances were obtained using relaxation S^{23}. Numerical experiments on some real-life problems were carried out using a bound evaluation algorithm, whose purpose is to assess the quality of the proposed alternative bounds for quadrilinear terms. This "partial sBB" algorithm at each branching step only records the most promising node and discards the other, thus exploring a single branch up to a leaf. The best bounds were obtained using a relaxation involving a trilinear envelope.

In the present chapter, we further investigate the strength of the proposed relaxations in a sBB algorithm. To that effect, we implemented the computation of the four relaxations for quadrilinear monomials in COUENNE [5]. Computational experiments were carried out running COUENNE on seven instances of the molecular distance geometry problem (MDGP) [11], the problem of finding an embedding $x : V \to \mathbb{R}^3$ of the vertices V of a weighted graph $G = (V,E)$ such that all the edge weights d_{uv} (for $\{u,v\} \in E$) are equal to the Euclidean distances $\|x_u - x_v\|$. The MDGP mathematical programming formulation is:

$$\min_x \sum_{\{u,v\} \in E} \left(\|x_u - x_v\|^2 - d_{uv}^2\right)^2, \tag{20}$$

a nonconvex NLP involving polynomials of fourth degree. In our experiments we impose a time limit equal to 4 h. Results were obtained on a 2.4 GHz Intel Xeon CPU of a computer with 8 GB RAM shared by three other similar CPU running Linux. For the smallest MDGP instance, the optimal solution is computed within the time limit using all the considered relaxations. A comparison of CPU time is reported in Table 1 and shows that the time needed to solve the problem when relaxation S^{23} is used is 81 % smaller than the time needed using S^{222}, which is the second best time to solve the problem. For the other instances, for which the optimal solution is not reached within the time limit, we compare the (lower) bounds obtained with the four relaxations. Results are shown in Table 2. These results confirm the results obtained in [6]. It appears that the best bounds are always obtained using a relaxation involving a trilinear envelope and, in five cases out of six, correspond to relaxation S^{23}. The sBB based on this relaxation gives bounds which are significantly better than the ones obtained using a relaxation based on the composition of bilinear

Table 1 Comparison of CPU time (seconds) obtained by running COUENNE with relaxations S^{222}, $\tilde{S}^{222}$, S^{32}, S^{23} on the smallest MDGP instance

Instance	S^{222}	$\tilde{S}^{222}$	S^{32}	S^{23}
lavor3	311.934	372.306	475.835	**58.0872**

The best value is reported in bold face
Solutions were obtained on a 2.4 GHz Intel Xeon CPU of a computer with 8 GB RAM shared by three other similar CPU running Linux

Table 2 Comparison of lower bounds obtained by running COUENNE with relaxations S^{222}, $\tilde{S}^{222}$, S^{32}, S^{23} on MDGP instances

Instance	S^{222}	$\tilde{S}^{222}$	S^{32}	S^{23}
lavor5	**228.574**$^{(*)}$	199.864	200.45	**228.574**$^{(*)}$
lavor6	93.4905	135.899	84.9467	**144.399**
lavor7	2.75184	90.3962	70.9786	**207.255**
lavor8	24.5401	95.0223	36.421	**334.968**
lavor10	−266.843	−105.584	−91.4539	**93.6579**
lavor20	−1571.58	−1215.7	−**589.636**	−1146.5

Bounds were obtained within a 4 h time limit
The best values are reported in bold face
The symbol (*) denotes optimal solutions found
Solutions were obtained on a 2.4 GHz Intel Xeon CPU of a computer with 8 GB RAM shared by three other similar CPU running Linux

envelopes, in particular on the largest instances. For the first instance in Table 2 the optimal solution is found with relaxations S^{222} and S^{23} within the time limit. It took 8,311.97 s in the first case and 7,063.73 s in the second one.

5 Conclusion

We analyzed four different convex relaxations for quadrilinear monomials, obtained by the composition of the known convex envelopes for bilinear and trilinear monomials. Starting from theoretical as well as computational results given in [6], we further investigated these relaxations. We provided an alternative proof of the fact that a relaxation of k-linear terms that employs a successive use of relaxing bilinear terms (via the bilinear convex envelope) can be improved by employing instead a relaxation of a trilinear term (via the trilinear convex envelope). We computationally evaluated the impact of the tightened convex relaxations in a spatial Branch-and-Bound algorithm on a set of instances of a real-life problem.

Acknowledgements The second and the fourth authors gratefully acknowledge financial support under ANR grant 07-JCJC-0151. The work of the third author was partially supported by NSF Grant CMMI–1160915.

References

1. C.S. Adjiman, S. Dallwig, C.A. Floudas, and A. Neumaier. A global optimization method, αBB, for general twice-differentiable constrained NLPs: I. Theoretical advances. *Computers & Chemical Engineering*, 22(9):1137–1158, 1998.
2. F.A. Al-Khayyal and J.E. Falk. Jointly constrained biconvex programming. *Mathematics of Operations Research*, 8(2):273–286, 1983.
3. K.M. Anstreicher. Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming. *Journal of Global Optimization*, 43(2-3):471–484, 2009.
4. X. Bao, N.V. Sahinidis, and M. Tawarmalani. Multiterm polyhedral relaxations for nonconvex, quadratically constrained quadratic programs. *Optimization Methods and Software*, 24:485–504, 2009.
5. P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, 24(4):597–634, 2009.
6. S. Cafieri, J. Lee, and L. Liberti. On convex relaxations of quadrilinear terms. *Journal of Global Optimization*, 47:661–685, 2010.
7. R.M. Karp and C.H. Papadimitrou. On linear characterizations of combinatorial optimization problems. *SIAM Journal on Computing*, 11:620–632, 1982.
8. S. Kim and M. Kojima. Second order cone programming relaxation of nonconvex quadratic optimization problems. *Optimization Methods and Software*, 15:201–204, 2001.
9. L. Liberti. Writing global optimization software. In L. Liberti and N. Maculan, editors, *Global Optimization: from Theory to Implementation*, pages 211–262. Springer, Berlin, 2006.
10. L. Liberti, S. Cafieri, and F. Tarissan. Reformulations in mathematical programming: a computational approach. In A. Abraham, A.-E. Hassanien, P. Siarry, and A. Engelbrecht, editors, *Foundations on Computational Intelligence vol.3*, volume 203 of *Studies in Computational Intelligence*, pages 153–234. Springer, Berlin, 2009.
11. L. Liberti, C. Lavor, A. Mucherino, and N. Maculan. Molecular distance geometry methods: from continuous to discrete. *International Transactions in Operational Research*, 18:33–51, 2010.
12. L. Liberti and C.C. Pantelides. Convex envelopes of monomials of odd degree. *Journal of Global Optimization*, 25:157–168, 2003.
13. L. Liberti, P. Tsiakis, B. Keeping, and C.C. Pantelides. *ooOPS*. Centre for Process Systems Engineering, Chemical Engineering Department, Imperial College, London, UK, 2001.
14. Y. Lin and L. Schrage. The global solver in the LINDO API. *Optimization Methods and Software*, 24:657–668, 2009.
15. J. Luedtke, M. Namazifar, and J. Linderoth. Some results on the strength of relaxations of multilinear functions. Technical Report #1678, University of Wisconsin-Madison. Submitted, 2010.
16. G.P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I — Convex underestimating problems. *Mathematical Programming*, 10:146–175, 1976.
17. C.A. Meyer and C.A. Floudas. Trilinear monomials with positive or negative domains: Facets of the convex and concave envelopes. In C.A. Floudas and P.M. Pardalos, editors, *Frontiers in Global Optimization*, pages 327–352. Kluwer Academic Publishers, Amsterdam, 2003.
18. C.A. Meyer and C.A. Floudas. Trilinear monomials with mixed sign domains: Facets of the convex and concave envelopes. *Journal of Global Optimization*, 29(2):125–155, 2004.
19. C.A. Meyer and C.A. Floudas. Convex envelopes for edge-concave functions. *Mathematical Programming*, 103:207–224, 2005.
20. M. Namazifar, P. Belotti, and A.J. Miller. Valid inequalities, separation, and convex hulls for bounded multilinear functions. In preparation. Presented at MIP 2010, Atlanta, USA, 2010.
21. A. Rikun. A convex envelope formula for multilinear functions. *Journal of Global Optimization*, 10(4):425–437, 1997.
22. H.S. Ryoo and N.V. Sahinidis. A branch-and-reduce approach to global optimization. *Journal of Global Optimization*, 8(2):107–138, March 1996.

23. N.V. Sahinidis and M. Tawarmalani. Baron 8.1.1: Global optimization of mixed-integer nonlinear programs. Users Manual. Available at http://www.gams.com/dd/docs/solvers/baron.pdf, 2008.
24. A. Saxena, P. Bonami, and J. Lee. Convex relaxations of non-convex mixed integer quadratically constrained programs: Extended formulations. *Mathematical Programming B*, 124:383–411, 2010.
25. A. Saxena, P. Bonami, and J. Lee. Convex relaxations of non-convex mixed integer quadratically constrained programs: Projected formulations. *Mathematical Programming*, 130(2): 359–413, 2011.
26. H.D. Sherali. Convex envelopes of multilinear functions over a unit hypercube and over special discrete sets. *Acta Mathematica Vietnamica*, 22:245–270, 1997.
27. E.M.B. Smith and C.C. Pantelides. A symbolic reformulation/spatial Branch-and-Bound algorithm for the global optimisation of nonconvex MINLPs. *Computers & Chemical Engineering*, 23:457–478, 1999.
28. F. Tardella. Existence and sum decomposition of vertex polyhedral convex envelopes. *Optimization Letters*, 2:363–375, 2008.
29. F. Tardella. On the existence of polyhedral convex envelopes. In C.A. Floudas and P.M. Pardalos, editors, *Frontiers in Global Optimization*, pages 149–188. Kluwer Academic Amsterdam Publishers, Amsterdam, 2008.
30. M. Tawarmalani and N.V. Sahinidis. Convex extensions and convex envelopes of l.s.c. functions. *Mathematical Programming*, 93:247–263, 2002.
31. M. Tawarmalani and N.V. Sahinidis. Global optimization of mixed integer nonlinear programs: A theoretical and computational study. *Mathematical Programming*, 99:563–591, 2004.

An Oriented Distance Function Application to Gap Functions for Vector Variational Inequalities

Lkhamsuren Altangerel, Gert Wanka, and Oleg Wilfer

Abstract This paper aims to extend some results dealing with gap functions for vector variational inequalities from the literature by using the so-called oriented distance function.

Key words Vector variational inequalities • Gap function • Oriented distance function.

1 Introduction

The so-called gap function approach allows to reduce the investigation of variational inequalities into the study of optimization problems. Let us mention several papers which are devoted to the study of set-valued gap functions for vector variational inequalities. Specially, the generalizations of Auslender's and Giannessi's gap functions for vector variational inequalities have been introduced in [5]. More recently, a conjugate duality approach to the construction of a gap function has been applied to vector variational inequalities (see [2]).

On the other hand, scalarization techniques in vector optimization have been applied to the construction of a gap function for vector variational inequalities.

L. Altangerel (✉)
School of Mathematics and Computer Science, National University of Mongolia
e-mail: lkal@num.edu.mn

G. Wanka • O. Wilfer
Faculty of Mathematics, Chemnitz University of Technology, Germany
e-mail: gert.wanka@mathematik.tu-chemnitz.de; oleg.wilfer@mathematik.tu-chemnitz.de

A. Chinchuluun et al. (eds.), *Optimization, Simulation, and Control*, Springer Optimization and Its Applications 76, DOI 10.1007/978-1-4614-5131-0_2,

For instance, we refer to [3, 11, 13, 17] for vector variational inequalities, to [12] for generalized vector variational inequalities and to [15] for set-valued vector variational-like inequalities.

This paper concentrates on scalar-valued gap functions for vector variational inequalities on the basis of the oriented distance function and the approach presented in [14]. For some investigations dealing with the oriented distance function we refer to [6–10, 16] and [18]. The oriented distance function allows us to extend some results dealing with gap functions for vector variational inequalities from the literature (cf. [11–13, 15] and [17]).

The paper is organized as follows. In section 2 we recall some preliminary results dealing with the oriented distance function. The section 3 is devoted to introduce gap functions for vector variational inequalities. Moreover, we suggest another type of gap functions, which are based on dual problems. For this purpose, we use the powerful approach of the perturbation theory of the conjugate duality. We conclude our paper with the extension to some set-valued problems in section 4.

2 Mathematical preliminaries

Let X be a Hausdorff locally convex space. The dual space of X is denoted by X^*. For $x \in X$ and $x^* \in X^*$, let $\langle x^*, x\rangle := x^*(x)$ be the value of the linear continuous functional x^* at x. For a subset $A \subseteq X$ we define the indicator function $\delta_A : X \to \overline{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$ by

$$\delta_A(x) := \begin{cases} 0, & \text{if } x \in A, \\ +\infty, & \text{otherwise.} \end{cases}$$

For a given function $h : X \to \overline{\mathbb{R}}$ the effective domain is

$$\operatorname{dom} h := \{x \in X : \ h(x) < +\infty\}.$$

The function h is called proper if $\operatorname{dom} h \neq \emptyset$ and $h(x) > -\infty$ for all $x \in X$. For a nonempty subset $E \subseteq X$ we define the conjugate function of h by

$$h_E^* : X^* \to \overline{\mathbb{R}}, \ h_E^*(p^*) = (h + \delta_E)^*(p^*) = \sup_{x \in E}\{\langle x^*, x\rangle - h(x)\}.$$

When $E = X$, one can see that h_E^* turns into the classical Fenchel-Moreau conjugate function of h denoted by h^*.

Let Y be a real Banach space partially ordered by a closed convex pointed cone C with nonempty interior, i.e. $\operatorname{int} C \neq \emptyset$. A weak ordering in Y is defined by

$$y \prec x \Leftrightarrow x - y \in \operatorname{int} C \ \text{ and } \ y \not\prec x \Leftrightarrow x - y \notin \operatorname{int} C, \ x, y \in Y.$$

Let $C^* = \{y^* \in Y^* : \ \langle y^*, y\rangle \geq 0, \ \forall y \in C\}$ be the dual cone of C. The space of the linear continuous mappings from X to Y is denoted by $\mathcal{L}(X,Y)$. Moreover, let $S_Y := \{y \in Y : \ \|y\| = 1\}$ and $S(M) := \{y \in M : \ \|y\| = 1\}$, $M \subseteq Y$.
Further, we consider a Hausdorff locally convex vector space Z and a nonempty convex cone $D \subseteq Z$, which induces on Z a partial ordering $\leqq_D$, i.e. for $x, y \in Z$ it holds $x \leqq_D y \Leftrightarrow y - x \in D$. We attach to Z a greatest and a smallest element with respect to "$\leqq_D$", denoted by $+\infty_D$ and $-\infty_D$, respectively, which do not belong to Z and denote $\overline{Z} = Z \cup \{\pm\infty_D\}$. Besides, we define $x \leq_D y$ if and only if $x \leqq_D y$ and $x \neq y$. For all $x \in \overline{Z}$ it holds $-\infty_D \leqq_D x \leqq_D +\infty_D$ and for all $x \in Z$ it holds $-\infty_D \leq_D x \leq_D +\infty_D$.

In this paper, we consider on $\overline{Z}$ the following operations and conventions (cf. [4]): $x + (+\infty_D) = (+\infty_D) + x := +\infty_D \ \forall x \in Z \cup \{+\infty_D\}$, $x + (-\infty_D) = (-\infty_D) + x := -\infty_D \ \forall x \in Z \cup \{-\infty_D\}$, $\lambda \cdot (+\infty_D) := +\infty_D \ \forall \lambda \in (0, +\infty]$, $\lambda \cdot (+\infty_D) := -\infty_D \ \forall \lambda \in [-\infty, 0)$, $\lambda \cdot (-\infty_D) := -\infty_D \ \forall \lambda \in (0, +\infty]$, $\lambda \cdot (-\infty_D) := +\infty_D \ \forall \lambda \in [-\infty, 0)$, $(+\infty_D) + (-\infty_D) = (-\infty_D) + (+\infty_D) := +\infty_D$, $0(+\infty_D) := +\infty_D$ and $0(-\infty_D) := 0$. Further, define $\langle z^*, +\infty_D\rangle := +\infty_D$ for $z^* \in D^*$.
For a vector function $g : X \to \overline{Z}$ the domain is the set $\operatorname{dom} g := \{x \in X : g(x) \neq +\infty_D\}$. If $g(x) \neq -\infty_D$ for all $x \in X$ and $\operatorname{dom} g \neq \emptyset$, then the vector function g is called proper.

When $g(\lambda x + (1-\lambda)y) \leqq_D \lambda g(x) + (1-\lambda) g(y)$ holds for all $x, y \in X$ and all $\lambda \in [0,1]$ the vector function g is said to be D-convex.

Definition 2.1. Let $M \subseteq Y$. Then the function $\Delta_M : Y \to \overline{\mathbb{R}}$ defined by

$$\Delta_M(y) := d_M(y) - d_{M^c}(y), \ y \in Y,$$

is called the oriented distance function, where $d_M(y) = \inf_{z \in M} \|y - z\|$ is the distance function from the point $y \in Y$ to the set M and $M^c := Y \backslash M$.

The oriented distance function was introduced by Hiriart-Urruty ([9], [10]) in order to investigate optimality conditions in nonsmooth optimization. The main properties of Δ_M can be summarized as follows.

Proposition 2.1. *([18]) Let $M \subseteq Y$ be an nontrivial subset of Y, i.e., $M \neq \emptyset$ and $M \neq Y$. Then*

(i) *Δ_M is real-valued.*
(ii) *Δ_M is Lipschitz function with constant 1.*
(iii) $\Delta_M(y) = \begin{cases} < 0, \ \forall y \in \operatorname{int} M, \\ = 0, \ \forall y \in \partial M, \\ > 0, \ \forall y \in \operatorname{int} M^c. \end{cases}$
(iv) *if M is closed, then it holds that $M := \{y : \Delta_M(y) \leq 0\}$.*
(v) *if M is convex, then Δ_M is convex.*
(vi) *if M is a cone, then Δ_M is positively homogeneous.*

(vii) if M is a closed convex cone, then Δ_M is nonincreasing with respect to the ordering relation induced on Y, i.e., if $y_1, y_2 \in Y$, then

$$y_1 - y_2 \in M \;\Rightarrow\; \Delta_M(y_1) \leq \Delta_M(y_2);$$

if $\operatorname{int} M \neq \emptyset$, then

$$y_1 - y_2 \in \operatorname{int} M \;\Rightarrow\; \Delta_M(y_1) < \Delta_M(y_2).$$

Proposition 2.2. *([14]) Let $M \subseteq Y$ be convex and $ri(M) \neq \emptyset$. Then Δ_M can be represented as*

$$\Delta_M(y) = \sup_{x^* \in S_{Y^*}} \inf_{x \in M} \langle x^*, y - x \rangle, \; \forall y \in Y,$$

where $ri\,(M) := \begin{cases} rint\, M, & \text{if } aff(M) \text{ is closed}, \\ \emptyset, & \text{otherwise} \end{cases}$ and by $rint\, M$ we denote the interior of M with respect to affine hull $aff(M)$.

Remark. The above-mentioned canonical representation of a convex set has been investigated also in [6], [7] and [8].

Corollary 2.1. *([14]) For a convex cone C with $\operatorname{int} C \neq \emptyset$, we have that*

$$\Delta_C(y) = \sup_{x^* \in S(C^*)} \langle -x^*, y \rangle, \; \forall y \in Y.$$

Let $M \subseteq Y$ be a given set. Then one can introduce the function ξ defined by

$$\xi_{M,C}(y) := -\inf_{z \in M} \Delta_C(y - z), \; \forall y \in Y.$$

Proposition 2.3. *(cf. [14]) The following assertions are true.*

(i) $\xi_{M,C}(y) = \sup_{x \in M} \inf_{x^* \in S(C^*)} \langle x^*, y - x \rangle, \; \forall y \in Y.$

(ii) $\xi_{M,C}(y) \geq 0, \; \forall y \in M.$

Proof:

(i) This formula follows directly from Corollary 2.1 and the Definition of $\xi_{M,C}$.

(ii) For $y \in M$ there is

$$\xi_{M,C}(y) = \sup_{x \in M} \inf_{x^* \in S(C^*)} \langle x^*, y - x \rangle \geq \inf_{x^* \in S(C^*)} \langle x^*, y - y \rangle = 0.$$

□

3 Gap functions for vector variational inequalities

Let X be a Hausdorff locally convex space and Y be a real Banach space, $K \subseteq X$ be a nonempty set and $F : K \to \mathcal{L}(X,Y)$ be a given mapping. We consider the weak vector variational inequality which consists in finding $\bar{x} \in K$ such that

$$(WVVI) \qquad \langle F(\bar{x}), y - \bar{x}\rangle \not< 0, \ \forall y \in K,$$

where $\langle F(\bar{x}), y - \bar{x}\rangle \in Y$ denotes the image of $y - \bar{x} \in X$ under the linear continuous mapping $F(\bar{x}) \in \mathcal{L}(X,Y)$, we use this notation synonymously with $F(\bar{x})(y - \bar{x})$.

In this section we concentrate on the investigation of scalar-valued gap functions for the problem $(WVVI)$ on the basis of the oriented distance function. Recently a similar approach was applied to vector optimization in [14]. Let us recall the definition of a gap function for $(WVVI)$.

Definition 3.1. A function $\gamma : K \to \overline{\mathbb{R}}$ is said to be a gap function for the problem $(WVVI)$ if it satisfies the following properties:

(i) $\gamma(x) \geq 0, \ \forall x \in K$;
(ii) $\gamma(\bar{x}) = 0$ if and only if $\bar{x}$ solves the problem $(WVVI)$.

Additionally, we want to consider another type of gap functions which have weaker properties as the gap functions defined above. These functions are called weak gap functions.

Definition 3.2. A function $\gamma : K \to \overline{\mathbb{R}}$ is said to be a weak gap function for the problem $(WVVI)$ if it satisfies the following properties:

(i) $\gamma(x) \geq 0, \ \forall x \in K$;
(ii) $\gamma(\bar{x}) = 0 \Rightarrow \bar{x}$ solves the problem $(WVVI)$.

Let us introduce with

$$F(x)K := \{z \in Y : \exists w \in K \text{ such that } z = \langle F(x), w\rangle\}$$

(notice again that $\langle F(x), w\rangle$ stands for $F(x)w$ and $F(x) \in \mathcal{L}(X,Y)$) the function (setting $M = F(x)K$ for any $x \in K$)

$$\begin{aligned}\gamma^F_{\Delta,x}(y) :&= \xi_{F(x)K,C}(y) = -\inf_{z \in F(x)K} \Delta_C(y - z)\\ &= -\inf_{w \in K} \Delta_C(y - \langle F(x), w\rangle), \ \forall y \in Y.\end{aligned}$$

Then setting $y = \langle F(x), x\rangle$ in $\gamma^F_{\Delta,x}(y)$ we define for $x \in K$

$$\begin{aligned}\gamma^F_{\Delta}(x) &:= \gamma^F_{\Delta,x}(\langle F(x), x\rangle) = \xi_{F(x)K,C}(\langle F(x), x\rangle)\\ &= -\inf_{w \in K} \Delta_C(\langle F(x), x - w\rangle)\end{aligned}$$

$$= -\inf_{w\in K}\sup_{y^*\in S(C^*)}\langle -y^*,\langle F(x),x-w\rangle\rangle$$

$$= \sup_{w\in K}\inf_{y^*\in S(C^*)}\langle y^*,\langle F(x),x-w\rangle\rangle$$

because of Corollary 2.1.

Theorem 3.1. *γ_Δ^F is a gap function for $(WVVI)$.*

Proof. (i) By Proposition 2.3(ii) it holds $\gamma_\Delta^F(x) = \xi_{F(x)K,C}(\langle F(x),x\rangle) \geq 0$, since $\langle F(x),x\rangle \in F(x)K$ for any $x \in K$.

(ii) Let $x \in K$ be fixed. Then $\gamma_\Delta^F(x) > 0$ if and only if $\exists \widetilde{x} \in K$ such that

$$\Delta_C(\langle F(x),x-\widetilde{x}\rangle) < 0 \Leftrightarrow \langle F(x),x-\widetilde{x}\rangle \in \operatorname{int} C.$$

This equivalently means that $\langle F(x),\widetilde{x}-x\rangle \prec 0$, i.e., x is not a solution to $(WVVI)$. Consequently, taking (i) into account, for some $x \in K$ it holds $\gamma_\Delta^F(x) = 0$ if and only if x is a solution to $(WVVI)$. □

In order to suggest some other gap functions, let us consider optimization problems having the composition with a linear continuous mapping in the objective function and formulate some duality results. As mentioned in the introduction we use for our investigations the perturbation theory (cf. [4]), where to a general primal problem

$$(P) \qquad \inf_{x\in X}\Phi(x,0),$$

with the perturbation function $\Phi : X \times Y \to \overline{\mathbb{R}}$, the conjugate dual problem is defined by:

$$(D) \qquad \sup_{p^*\in Y^*}\{-\Phi^*(0,p^*)\}.$$

Assume that $f : Y \to \overline{\mathbb{R}}$ and $g : X \to \overline{Z}$ are proper functions, $S \subseteq X$ a nonempty set and $A \in \mathcal{L}(X,Y)$ fullfilling $(y - A^{-1}(\operatorname{dom} f)) \cap g^{-1}(-D) \cap S \neq \emptyset$. Consider the following primal optimization problem

$$(P^C) \qquad \inf_{x\in K} f(A(y-x))$$
$$K = \{x \in S : g(x) \in -D\},$$

where $y \in K$ is fixed and $S \subseteq X$.
The first dual problem of interest is the well-known Lagrange-dual problem:

$$(D^{C_L}) \qquad \sup_{z^*\in D^*}\inf_{x\in S}\{f(A(y-x)) + \langle z^*,g(x)\rangle\}.$$

To construct another dual problem we introduce the following perturbation function $\Phi^{C_F} : X \times Y \to \overline{\mathbb{R}}$,

$$\Phi^{C_F}(x,p) := \begin{cases} f(A(y-x)+p), \text{ if } x \in S, g(x) \in -D, \\ +\infty, \text{ otherwise,} \end{cases}$$

where $p \in Y$ is the perturbation variable. The perturbation function can be written as

$$\Phi^{C_F}(x,p) = f(A(y-x)+p) + \delta_K(x).$$

For the formula of the conjugate function $(\Phi^{C_F})^* : X^* \times Y^* \to \overline{\mathbb{R}}$ of Φ^{C_F} we get for all $(x^*,p^*) \in X^* \times Y^*$:

$$\begin{aligned}
(\Phi^{C_F})^*(x^*,p^*) &= \sup_{x\in X, p\in Y} \{\langle x^*,x\rangle + \langle p^*,p\rangle - \Phi^{C_F}(x,p)\} \\
&= \sup_{x\in X, p\in Y} \{\langle x^*,x\rangle + \langle p^*,p\rangle - f(A(y-x)+p) - \delta_K(x)\} \\
&= \sup_{x\in X, r\in Y} \{\langle x^*,x\rangle + \langle p^*,r-A(y-x)\rangle - f(r) - \delta_K(x)\} \\
&= \sup_{x\in X, r\in Y} \{\langle x^*,x\rangle + \langle p^*,r\rangle - \langle p^*,Ay\rangle + \langle p^*,Ax\rangle - f(r) - \\
&\quad \delta_K(x)\} \\
&= \sup_{x\in X}\{\langle x^*+A^*p^*,x\rangle - \delta_K(x)\} + \sup_{r\in Y}\{\langle p^*,r\rangle - f(r)\} - \\
&\quad \langle A^*p^*,y\rangle \\
&= \delta_K^*(x^*+A^*p^*) + f^*(p^*) - \langle A^*p^*,y\rangle.
\end{aligned}$$

This leads to the following dual problem to (P^C):

$$(D^{C_F}) \qquad \sup_{p^*\in Y^*} \{\langle A^*p^*,y\rangle - \delta_K^*(A^*p^*) - f^*(p^*)\},$$

which can be interpreted as a Fenchel dual problem. As the above construction shows it applies also if K is any nonempty set not necessarily given in the form as in (P^C).

Remark. From the calculations we made above for the Fenchel dual problem, we can easily conclude that to the primal problem

$$(\overline{P}) \qquad \inf_{x\in X}\{f(A(y-x)) + g(x)\},$$

where $A \in \mathcal{L}(X,Y)$ and $f : Y \to \overline{\mathbb{R}}$ and $g : X \to \overline{\mathbb{R}}$ are proper functions fullfilling $(y - A^{-1}(\operatorname{dom} f)) \cap \operatorname{dom} g \neq \emptyset$, the Fenchel dual problem looks like

$$(\overline{D}) \qquad \sup_{p^*\in Y^*} \{\langle A^*p^*,y\rangle - g^*(A^*p^*) - f^*(p^*)\}.$$

The last perturbation function we consider leads to the Fenchel-Lagrange dual problem and is defined by $\Phi^{C_{FL}} : X \times Y \times Z \to \overline{\mathbb{R}}$,

$$\Phi^{C_{FL}}(x,p,z) := \begin{cases} f(A(y-x)+p), \text{ if } x \in S, g(x) \in z - D, \\ +\infty, \text{ otherwise,} \end{cases}$$

where $(p,z) \in Y \times Z$ are the perturbation variables. We define $(z^* g)(x) := \langle z^*, g(x) \rangle$ and obtain for the conjugate of $\Phi^{C_{FL}}$, $(\Phi^{C_{FL}})^* : X^* \times Y^* \times Z^* \to \overline{\mathbb{R}}$, for all $(x^*, p^*, z^*) \in X^* \times Y^* \times Z^*$:

$$\begin{aligned}
(\Phi^{C_{FL}})^*(x^*,p^*,z^*) &= \sup_{\substack{x \in X, p \in Y \\ z \in Z}} \{\langle x^*, x\rangle + \langle p^*, p\rangle + \langle z^*, z\rangle - \Phi^{C_{FL}}(x,p,z)\} \\
&= \sup_{\substack{x \in S, (p,z) \in Y \times Z \\ g(x) \in z - D}} \{\langle x^*, x\rangle + \langle p^*, p\rangle + \langle z^*, z\rangle - \\
&\quad f(A(y-x)+p)\} \\
&= \sup_{\substack{x \in S, r \in Y \\ s \in -D}} \{\langle x^*, x\rangle + \langle p^*, r - A(y-x)\rangle + \\
&\quad \langle z^*, g(x) - s\rangle - f(r)\} \\
&= \sup_{\substack{x \in S, r \in Y \\ s \in -D}} \{\langle x^*, x\rangle + \langle p^*, r\rangle - \langle p^*, Ay\rangle + \\
&\quad \langle p^*, Ax\rangle + \langle z^*, g(x)\rangle + \langle -z^*, s\rangle - f(r)\} \\
&= \sup_{s \in -D} \{\langle -z^*, s\rangle\} + \sup_{r \in Y} \{\langle p^*, r\rangle - f(r)\} + \\
&\quad \sup_{x \in S} \{\langle x^* + A^* p^*, x\rangle - \langle -z^*, g(x)\rangle\} - \langle A^* p^*, y\rangle \\
&= \delta_{-D^*}(z^*) + f^*(p^*) + (-z^* g)^*_S(x^* + A^* p^*) - \langle A^* p^*, y\rangle.
\end{aligned}$$

As a consequence, the Fenchel-Lagrange dual problem is actually

$$(D^{C_{FL}}) \qquad \sup_{(p^*,z^*) \in Y^* \times D^*} \{\langle A^* p^*, y\rangle - f^*(p^*) - (z^* g)^*_S(A^* p^*)\}.$$

According to the general theory (cf. [4]) the weak duality is always fullfilled, i.e. $v(D^{C_L}) \leq v(P^C)$, $v(D^{C_F}) \leq v(P^C)$ and $v(D^{C_{FL}}) \leq v(P^C)$, where $v(P^C), v(D^{C_L}), v(D^{C_F})$ and $v(D^{C_{FL}})$ are the optimal objective values of $(P^C), (D^{C_L})$, (D^{C_F}) and $(D^{C_{FL}})$, respectively.
We consider now for any $x \in K$ the following optimization problem

$$(P_x^F) \qquad \inf_{y \in K} \Delta_C(\langle F(x), x - y\rangle),$$

where the ground set K is defined by

$$K = \{y \in S : g(y) \in -D\}.$$

It is easy to see that the optimal objective value $v(P_x^F) = -\gamma_\Delta^F(x) \leq 0,\ \forall x \in K$. By using the calculations we made above we get for the dual problems of (P_x^F):

$$(D_x^{F_L}) \qquad \sup_{z^* \in D^*} \inf_{y \in S} \{\Delta_C(\langle F(x), x - y\rangle) + \langle z^*, g(y)\rangle\},$$

$$(D_x^{F_F}) \qquad \sup_{p^* \in Y^*} \{\langle F(x)^* p^*, x\rangle - \Delta_C^*(p^*) - \delta_K^*(F(x)^* p^*)\}$$

(here, K may be any nonempty set) and

$$(D_x^{F_{FL}}) \qquad \sup_{p^* \in Y^*, z^* \in D^*} \{\langle F(x)^* p^*, x\rangle - (z^* g)_S^*(F(x)^* p^*) - \Delta_C^*(p^*)\}.$$

It is well known that for any set $A \subseteq X$ it holds that $\sigma_A(x^*) = \sup_{x \in A}\langle x^*, x\rangle = \sigma_{\mathrm{clco}A}(x^*)$, whereas $\mathrm{clco}A$ is the closed convex hull of the set A. Hence, for a convex cone C with $\mathrm{int}C \neq \emptyset$, by Corollary 2.1 it follows that $\Delta_C(y) = \sup_{y^* \in S(C^*)}\langle -y^*, y\rangle = \sup_{y^* \in S(-C^*)}\langle y^*, y\rangle = \sigma_{S(-C^*)}(y) = \sigma_{\mathrm{clco}S(-C^*)}(y)$, i.e. $\Delta_C(y) = \sigma_{\mathrm{clco}S(-C^*)}(y) = \delta^*_{\mathrm{clco}S(-C^*)}(y)$. Further, since $\mathrm{clco}S(-C^*)$ is a closed convex set we have by the Fenchel-Moreau Theorem (cf. [4, Theorem 2.3.6]) for the conjugate of the oriented distance function $\Delta_C^*(y^*) = \delta^{**}_{\mathrm{clco}S(-C^*)}(y^*) = \delta_{\mathrm{clco}S(-C^*)}(y^*)$. As a result, the Fenchel dual problem and the Fenchel-Lagrange dual problem can be written as

$$(D_x^{F_F}) \qquad \sup_{p^* \in \mathrm{clco}S(-C^*)} \{\langle F(x)^* p^* x\rangle - \delta_K^*(F(x)^* p^*)\}$$

and

$$(D_x^{F_{FL}}) \qquad \sup_{\substack{p^* \in \mathrm{clco}S(-C^*) \\ z^* \in D^*}} \{\langle F(x)^* p^*, x\rangle - (z^* g)_S^*(F(x)^* p^*)\}.$$

Example. Let $X = Y = \mathbb{R}^2$ be equipped with the Euclidean topology and $C = \mathbb{R}^2_+$, then we have $X^* = Y^* = \mathbb{R}^2$ also equipped with the Euclidean topology and $C^* = \mathbb{R}^2_+$. Let the ground set $K \subseteq X$ be a nonempty set and $F : K \to \mathcal{L}(X, Y)$ be a given mapping. For the set $\mathrm{clco}S(-C^*)$ we get

$$\begin{aligned}\mathrm{clco}S(-C^*) &= \{p^* = (p_1^*, p_2^*)^T \in -\mathbb{R}^2_+ \cap B(0,1) : p_1^* + p_2^* \leq -1\} \\ &= \{p^* \in \mathbb{R}^2 : \|p^*\| \leq 1, p_1^* + p_2^* \leq -1\},\end{aligned}$$

where $B(0,1) = \{p^* \in \mathbb{R}^2 : \|p^*\| \leq 1\}$. Therefore, the corresponding Fenchel dual problem looks like

$$(\widetilde{D}_x^F) \qquad \sup_{\substack{\|p^*\| \leq 1 \\ p_1^* + p_2^* \leq -1}} \{\langle F(x)^* p^*, x\rangle - \delta_K^*(F(x)^* p^*)\}.$$

Remark. If the convex set C is not a cone with $\operatorname{int} C \neq \emptyset$ we refer to [6] for the conjugate of the oriented distance function.

By using the duals $(D_x^{F_L}), (D_x^{F_F})$ and $(D_x^{F_{FL}})$ of the optimization problem (P_x^F), we introduce the following functions for $x \in K$:

$$\begin{aligned}\gamma_\Delta^{F_L}(x) :&= -\sup_{z^* \in D^*} \inf_{y \in S} \{\Delta_C(\langle F(x), x-y\rangle) + \langle z^*, g(y)\rangle\} \\ &= \inf_{z^* \in D^*} \sup_{y \in S} \{-\Delta_C(\langle F(x), x-y\rangle) - \langle z^*, g(y)\rangle\},\end{aligned}$$

$$\begin{aligned}\gamma_\Delta^{F_F}(x) :&= -\sup_{p^* \in \operatorname{clco} S(-C^*)} \{\langle F(x)^* p^*, x\rangle - \delta_K^*(F(x)^* p^*)\} \\ &= \inf_{p^* \in \operatorname{clco} S(-C^*)} \{\delta_K^*(F(x)^* p^*) - \langle F(x)^* p^*, x\rangle\}\end{aligned}$$

and

$$\begin{aligned}\gamma_\Delta^{F_{FL}}(x) :&= -\sup_{\substack{p^* \in \operatorname{clco} S(-C^*) \\ z^* \in D^*}} \{\langle F(x)^* p^*, x\rangle - (z^* g)_S^*(F(x)^* p^*)\} \\ &= \inf_{\substack{p^* \in \operatorname{clco} S(-C^*) \\ z^* \in D^*}} \{(z^* g)_S^*(F(x)^* p^*) - \langle F(x)^* p^*, x\rangle\}.\end{aligned}$$

Remark. A similar approach was introduced in [1] in order to construct a gap function for scalar variational inequalities.

Proposition 3.1. *It holds that*

$$\gamma_\Delta^{F_{FL}}(x) \geq \gamma_\Delta^{F_F}(x), \ \forall x \in K.$$

Proof. We fix $x \in K$ and $p^* \in Y^*$ and consider the following primal problem

$$(P^0) \qquad \inf_{y \in K} \langle -F(x)^* p^*, y\rangle,$$
$$K = \{y \in S : g(y) \in -D\}.$$

The corresponding Lagrange dual problem is

$$\begin{aligned}(D^0) \qquad &\sup_{z^* \in D^*} \{\inf_{y \in S} \langle -F(x)^* p^*, y\rangle + \langle z^*, g(y)\rangle\} \\ &= \sup_{z^* \in D^*} \inf_{y \in S} \{-[\langle F(x)^* p^*, y\rangle - \langle z^*, g(y)\rangle]\} \\ &= \sup_{z^* \in D^*} -\sup_{y \in S} \{\langle F(x)^* p^*, y\rangle - \langle z^*, g(y)\rangle\} \\ &= \sup_{z^* \in D^*} \{-(z^* g)_S^*(F(x)^* p^*)\}.\end{aligned}$$

By the weak duality it follows that

$$\sup_{z^*\in D^*}\{-(z^*g)_S^*(F(x)^*p^*)\} \le \inf_{y\in K}\{\langle -F(x)^*p^*, y\rangle\}$$

or, equivalently,

$$-\sup_{z^*\in D^*}\{-(z^*g)_S^*(F(x)^*p^*)\} + \delta_{\mathrm{clco}S(-C^*)}(p^*) - \langle F(x)^*p^*, x\rangle \ge$$

$$-\inf_{y\in K}\{\langle -F(x)^*p^*, y\rangle\} + \delta_{\mathrm{clco}S(-C^*)}(p^*) - \langle F(x)^*p^*, x\rangle.$$

Now we take the infimum over $p^* \in Y^*$ in both sides and get

$$\begin{aligned}\gamma_\Delta^{F_{FL}}(x) &= \inf_{\substack{p^*\in \mathrm{clco}S(-C^*)\\ z^*\in D^*}}\{(z^*g)_S^*(F(x)^*p^*) - \langle F(x)^*p^*, x\rangle\}\\ &\ge \inf_{p^*\in \mathrm{clco}S(-C^*)}\{\delta_K^*(F(x)^*p^*) - \langle F(x)^*p^*, x\rangle\} = \gamma_\Delta^{F_F}(x).\end{aligned}$$

□

Proposition 3.2. *It holds that*

$$\gamma_\Delta^{F_{FL}}(x) \ge \gamma_\Delta^{F_L}(x),\ \forall x \in K.$$

Proof. Let $z^* \in D^*$ be fixed. Since

$$\sup_{p^*\in Y^*}\{-\delta_{\mathrm{clco}S(-C^*)}(p^*) - (z^*g)_S^*(F(x)^*p^*) + \langle F(x)^*p^*, x\rangle\}$$

is the Fenchel dual problem of the primal problem (cf. Remark for the Fenchel dual problem)

$$\inf_{y\in X}\{\Delta_C(\langle F(x), x-y\rangle) + ((z^*g)+\delta_S)(y)\} =$$

$$\inf_{y\in S}\{\Delta_C(\langle F(x), x-y\rangle) + \langle z^*, g(y)\rangle\},$$

we get by the weak duality

$$\sup_{p^*\in Y^*}\{-\delta_{\mathrm{clco}S(-C^*)}(p^*) - (z^*g)_S^*(F(x)^*p^*) + \langle F(x)^*p^*, x\rangle\rangle\} \le$$

$$\inf_{y\in S}\{\Delta_C(\langle F(x), x-y\rangle) + \langle z^*, g(y)\rangle\}$$

or

$$\inf_{p^*\in Y^*}\{\delta_{\mathrm{clco}S(-C^*)}(p^*) + (z^*g)_S^*(F(x)^*p^*) - \langle F(x)^*p^*, x\rangle\rangle\} \ge$$

$$\sup_{y\in S}\{-\Delta_C(\langle F(x), x-y\rangle) - \langle z^*, g(y)\rangle\}.$$

Taking the infimum over $z^* \in D^*$ in both sides yields the desired result

$$\gamma_\Delta^{F_{FL}}(x) = \inf_{\substack{p^* \in \operatorname{clco} S(-C^*) \\ z^* \in D^*}} \{(z^* g)_S^*(F(x)^* p^*) - \langle F(x)^* p^*, x\rangle\} \geq$$

$$\inf_{z^* \in D^*} \sup_{y \in S} \{-\Delta_C(\langle F(x), x - y\rangle) - \langle z^*, g(y)\rangle\} = \gamma_\Delta^{F_L}(x).$$

□

Proposition 3.3. *It holds for all $x \in K$ that*

$$\gamma_\Delta^F(x) \leq \gamma_\Delta^{F_L}(x), \gamma_\Delta^F(x) \leq \gamma_\Delta^{F_F}(x) \text{ and } \gamma_\Delta^F(x) \leq \gamma_\Delta^{F_L}(x).$$

Proof. Let $x \in K$. Since $\gamma_\Delta^F(x) = -v(P_x^F), \gamma_\Delta^{F_L}(x) = -v(D_x^{F_L}), \gamma_\Delta^{F_F}(x) = -v(D_x^{F_F})$ and $\gamma_\Delta^{F_{FL}}(x) = -v(D_x^{F_{FL}})$ the assertions follow from the weak duality between (P_x^F) and the corresponding different dual problems. □

Remark. By the last three propositions we obtain the following relations between the introduced functions

$$\gamma_\Delta^{F_{FL}}(x) \geq \begin{matrix} \gamma_\Delta^{F_L}(x) \\ \gamma_\Delta^{F_F}(x) \end{matrix} \geq \gamma_\Delta^F(x) \; \forall x \in K,$$

which is equivalent to

$$v(P_x^F) \geq \begin{matrix} v(D_x^{F_L}) \\ v(D_x^{F_F}) \end{matrix} \geq v(D_x^{F_{FL}}) \; \forall x \in K.$$

Remark. The relations in the Remark above show that if strong duality for the pair $(P_x^F) - (D_x^{F_{FL}})$ holds, then strong duality holds also for the pairs $(P_x^F) - (D_x^{F_L})$ and $(P_x^F) - (D_x^{F_F})$.

Proposition 3.4. *$\gamma_\Delta^{F_L}, \gamma_\Delta^{F_F}$ and $\gamma_\Delta^{F_{FL}}$ are weak gap functions for the problem $(WVVI)$ where $K = \{y \in S : g(y) \in -D\} \neq \emptyset$. Concerning $\gamma_\Delta^{F_F}$, K may be any nonempty set.*

Proof. (i) By Theorem 3.1 and Propositions 3.1, 3.2 and 3.3 it holds that

$$\gamma_\Delta^{F_{FL}}(x) \geq \begin{matrix} \gamma_\Delta^{F_L}(x) \\ \gamma_\Delta^{F_F}(x) \end{matrix} \geq \gamma_\Delta^F(x) \geq 0 \; \forall x \in K.$$

(ii) Let $\gamma_\Delta^{F_L}(\bar{x}) = 0$ for some $\bar{x} \in K$. Then we obtain by (i) that $\gamma_\Delta^F(\bar{x}) = 0$. From Theorem 3.1 we have that $\bar{x}$ solves the problem $(WVVI)$.
For $\gamma_\Delta^{F_F}$ and $\gamma_\Delta^{F_{FL}}$ it follows analogously. □

In order to guarantee the strong duality between the primal problem (P^C) and the corresponding dual problems (D^{C_L}), (D^{C_F}) and $(D^{C_{FL}})$ we assume for the rest of this chapter that S is a convex set, f is a convex function and g is a D-convex function.

First, we state a strong duality proposition for the primal-dual pair $(P^C) - (D^{C_L})$, which is a direct conclusion of [4, Theorem 3.2.1].

Proposition 3.5. *If there exists $x' \in (y - A^{-1}(dom\, f)) \cap S$ such that $g(x') \in -int\, D$, then $v(P^C) = v(D^{C_L})$ and (D^{C_L}) has an optimal solution.*

In the case where $f = \Delta_C$ and $A = F(x)$, we have (notice that x and y are changed in (P_x^F) compared with (P^C))

$$\begin{aligned} & y' \in (x - F(x)^{-1}(\text{dom } \Delta_C)) \cap S \\ \Leftrightarrow\ & y' \in (x - F(x)^{-1}(Y)) \cap S \\ \Leftrightarrow\ & y' \in (x - X) \cap S \\ \Leftrightarrow\ & y' \in S. \end{aligned}$$

Therefore we have for the pair $(P_x^F) - (D_x^{F_L})$, $x \in K$, the following strong duality proposition.

Proposition 3.6. *If there exists $y' \in S$ such that $g(y') \in -int\, D$, then $v(P_x^F) = v(D_x^{F_L})$ and $(D_x^{F_L})$ has an optimal solution.*

Next, we give for any convex set $K \neq \emptyset$ a strong duality proposition for the primal-dual problems $(P^C) - (D^{C_F})$ by using [4, Theorem 3.2.1] again.

Proposition 3.7. *If there exists $x' \in (y - A^{-1}(dom\, f)) \cap K$ such that f is continuous at $A(y - x')$, then $v(P^C) = v(D^{C_F})$ and (D^{C_F}) has an optimal solution.*

Since Δ_C is a Lipschitz function, i.e. Δ_C is also continuous everywhere on Y, the Proposition 3.7 can be rewritten for the pairs $(P_x^F) - (D_x^{F_F})$, $x \in K$, as follows.

Proposition 3.8. *If $K \neq \emptyset$ is any convex set, then $v(P_x^F) = v(D_x^{F_F})$ and $(D_x^{F_F})$ has an optimal solution.*

Remark. Note that in this case there is no regularity condition needed.

Finally, we state a strong duality proposition for the primal-dual pair $(P^C) - (D^{C_{FL}})$ which also follows as a simple conclusion of [4, Theorem 3.2.1].

Proposition 3.9. *If there exists $x' \in (y - A^{-1}(dom\, f)) \cap S$ such that f is continuous at $A(y - x')$ and $g(x') \in -int\, D$, then $v(P^C) = v(D^{C_{FL}})$ and $(D^{C_{FL}})$ has an optimal solution.*

As application we can establish strong duality for (P_x^F) and $(D_x^{F_{FL}})$, $x \in K$.

Proposition 3.10. *If there exists $y' \in S$ such that $g(y') \in -int\, D$, then $v(P_x^F) = v(D_x^{F_{FL}})$ and $(D_x^{F_{FL}})$ has an optimal solution.*

Theorem 3.2. *(i)* $\gamma_\Delta^{F_F}$ *is a gap function for* $(WVVI)$ *for any convex set* $K \neq \emptyset$.
(ii) If there exists $y' \in S$ *such that* $g(y') \in -\operatorname{int} D$ *then* $\gamma_\Delta^{F_L}$ *and* $\gamma_\Delta^{F_{FL}}$ *are gap functions for* $(WVVI)$.

Proof. (i) By Proposition 3.4, it follows that $\gamma_\Delta^{F_F}$ is a weak gap function. For that reason, we need only to prove that if $\bar{x} \in K$ solves $(WVVI)$, then it holds that $\gamma_\Delta^{F_F}(\bar{x}) = 0$. According to Theorem 3.1, for some $\bar{x} \in K$ it holds that $\gamma_\Delta^F(\bar{x}) = 0$ if and only if $\bar{x}$ is a solution to $(WVVI)$. That means $v(P_{\bar{x}}^F) = -\gamma_\Delta^F(\bar{x}) = 0$. On the other hand, by Proposition 3.8 strong duality holds, i.e. if $\bar{x} \in K$ solves $(WVVI)$, then $\gamma_\Delta^{F_F}(\bar{x}) = -v(D_{\bar{x}}^{F_F}) = -v(P_{\bar{x}}^F) = 0$.
(ii) This can be proved in a similar way taking into account Proposition 3.6 and Proposition 3.10 instead of Proposition 3.8 as in the proof of i). □

4 Extension to set-valued problems

In this section we discuss how the presented approach can be extended to some variational inequalities with set-valued mappings investigated in the literature (see [11], [12], [13], [15] and [17]). Let us notice that in all mentioned works the space Y was supposed to be Euclidean one. Under compactness assumptions we will extend above results in Banach spaces.

4.1 Vector variational inequalities with set-valued mappings

Let X, Y be real Banach spaces, Y be partially ordered by a closed convex pointed cone C with $\operatorname{int} C \neq \emptyset$ and $\emptyset \neq K \subseteq X$ be a compact set. Further let $T : K \rightrightarrows \mathcal{L}(X,Y)$ be a set-valued mapping, where $\mathcal{L}(X,Y)$ is equipped with the usual operator norm, i.e. $(\mathcal{L}(X,Y), \|\cdot\|)$ is a Banach space. We consider the vector variational inequality with set-valued mapping which consists in finding $\bar{x} \in K$ such that

$$(SVVI) \qquad \exists \bar{t} \in T(\bar{x}) : \quad \langle \bar{t}, y - \bar{x} \rangle \not\prec 0, \ \forall y \in K.$$

Let us introduce the function

$$\begin{aligned} \gamma_S^T(x) &= -\sup_{t \in T(x)} \inf_{y \in K} \Delta_C(\langle t, x - y \rangle) \\ &= \inf_{t \in T(x)} \sup_{y \in K} \{-\Delta_C(\langle t, x - y \rangle)\}, \ x \in K. \end{aligned}$$

Theorem 4.1. *Assume that for each* $x \in K$, $T(x)$ *is nonempty and compact. Then* γ_S^T *is a gap function for* $(SVVI)$.

Proof: Let $x \in K$ and $t \in T(x)$. Then, from Proposition 2.3(ii) follows

$$\begin{aligned}\xi_{tK,C}(\langle t,x\rangle) : &= -\inf_{z\in tK} \Delta_C(\langle t,x\rangle - z) \ (\text{set } z := \langle t,y\rangle,\ y \in K)\\ &= -\inf_{y\in K} \Delta_C(\langle t,x\rangle - \langle t,y\rangle)\\ &= -\inf_{y\in K} \Delta_C(\langle t,x-y\rangle) \geq 0.\end{aligned}$$

Consequently, we have

$$\gamma_S^T(x) = \inf_{t\in T(x)} \xi_{tK,C}(\langle t,x\rangle) \geq 0.$$

Since K and $T(x)$ are compact and Δ_C is continuous, then by standard arguments (uniform continuity), we obtain that the function $\sup_{y\in K}\{-\Delta_C(\langle t,x-y\rangle)\}$ is continuous with respect to $t \in T(x)$. Moreover, the function γ_S^T is well defined and can be written as

$$\gamma_S^T(x) = \min_{t\in T(x)} \sup_{y\in K}\{-\Delta_C(\langle t,x-y\rangle)\}$$

(the infimum is attained). For some $\bar{x} \in K$ it holds $\gamma_S^T(\bar{x}) = 0$ if and only if $\exists \bar{t} \in T(\bar{x})$ such that

$$\sup_{y\in K}\{-\Delta_C(\langle \bar{t},\bar{x}-y\rangle)\} = 0$$

or

$$\Delta_C(\langle \bar{t},\bar{x}-y\rangle) \geq 0,\ \forall y \in K.$$

This equivalently means (cf. Proposition 2.1) that

$$\langle \bar{t},\bar{x}-y\rangle \notin \operatorname{int} C \ \Leftrightarrow \langle \bar{t},y-\bar{x}\rangle \not\prec 0,\ \forall y \in K,$$

i.e., $\bar{x}$ is a solution to $(SVVI)$. □

Example. If $Y = \mathbb{R}^m$, $C = \mathbb{R}^m_+$, then $Y^* = Y$, $C^* = \mathbb{R}^m_+$. Let $x,\ y \in K$. Then $T(x) = \prod_{i=1}^m T_i(x)$, $T_i : K \rightrightarrows X^*$. For any $t \in T$ it holds $t = (t_1,...,t_m)$ and

$$\langle t,x-y\rangle = (\langle t_1,x-y\rangle,...,\langle t_m,x-y\rangle).$$

According to Corollary 2.1 and Proposition 3(iv) in [14], we have

$$\Delta_C(\langle t,x-y\rangle) = \sup_{\substack{z\in\mathbb{R}^m_+\\ \|z\|=1}} \langle -z,\langle t,x-y\rangle\rangle = \max_{1\leq i\leq m} \langle t_i,y-x\rangle.$$

Consequently, we get

$$\gamma_S^T(x) = \inf_{t\in T(x)} \sup_{y\in K} \min_{1\le i\le m} \langle t_i, x-y\rangle$$

which is nothing else than the gap function for $(SVVI)$ investigated in [13] and [17].

4.2 Vector variational-like inequalities with set-valued mappings

Under the general assumptions of section 4.1 let $\eta : K\times K \to X$ be a vector-valued mapping such that $\eta(x,x) = 0,\ \forall x \in K$, which is continuous with respect to the first variable for any fixed second variable in K. Then the vector variational-like inequality with set-valued mapping consists in finding $\bar{x} \in K$ such that

$$(SVVLI) \qquad \exists \bar{t} \in T(\bar{x}): \quad \langle \bar{t}, \eta(y,\bar{x})\rangle \not< 0,\ \forall y \in K.$$

Let us introduce the function

$$\begin{aligned}\gamma_S^L(x) &= -\sup_{t\in T(x)} \inf_{y\in K} \Delta_C(-\langle t, \eta(y,x)\rangle)\\ &= \inf_{t\in T(x)} \sup_{y\in K}\{-\Delta_C(-\langle t, \eta(y,x)\rangle)\},\ x\in K,\end{aligned}$$

and verify the following assertion.

Theorem 4.2. *Assume that for each $x \in K$, $T(x)$ is nonempty and compact. Then γ_S^L is a gap function for $(SVVLI)$.*

Proof. First we prove that $\gamma_S^L(x) \ge 0\ \forall x \in K$. It holds $\eta(x,x) = 0\ \forall x \in K$ and hence $\langle t, \eta(x,x)\rangle = 0\ \forall x \in K, t \in T(x)$. Further we have by Corollary 2.1 that $\sup_{x^*\in S(C^*)}\langle -x^*, -\langle t, \eta(x,x)\rangle\rangle = 0\ \ \forall x \in K, t \in T(x)$, i.e. $\Delta_C(-\langle t, \eta(x,x)\rangle) = \Delta_C(0) = 0\ \ \forall x \in K, t \in T(x)$. By taking the infimum over $y \in K$ we get $\inf_{y\in K} \Delta_C(-\langle t, \eta(y,x)\rangle) \le 0\ \forall x \in K, t \in T(x)$, which is equivalent to $\sup_{y\in K}\{-\Delta_C(-\langle t, \eta(y,x)\rangle)\} \ge 0\ \forall x \in K, t \in T(x)$. Finally, it follows that

$$\gamma_S^L(x) = \inf_{t\in T(x)} \sup_{y\in K}\{-\Delta_C(-\langle t, \eta(y,x)\rangle)\} \ge 0\ \forall x \in K.$$

Next, we show that $\gamma_S^L(\bar{x}) = 0$ if and only if $\bar{x}$ solves $(SVVLI)$. As in the proof of Theorem 4.1 it follows that $\sup_{y\in K}\{-\Delta_C(-\langle t, \eta(y,x)\rangle)\}$ is a continuous function with respect to $t \in T(x)$. Moreover, from the assumption for $T(x)$, the function γ_S^L is well defined and can be formulated as

$$\gamma_S^L(x) = \min_{t\in T(x)} \sup_{y\in K}\{-\Delta_C(-\langle t, \eta(y,x)\rangle)\}.$$

Further, let $\bar{x} \in K$, then $\gamma_S^L(\bar{x}) = 0$ if and only if $\exists \bar{t} \in T(\bar{x})$ such that

$$\sup_{y \in K}\{-\Delta_C(-\langle \bar{t}, \eta(y,\bar{x})\rangle)\} = 0$$

and hence follows

$$\Delta_C(-\langle \bar{t}, \eta(y,\bar{x})\rangle) \geq 0 \quad \forall y \in K.$$

This implies

$$-\langle \bar{t}, \eta(y,\bar{x})\rangle \notin \operatorname{int} C \Leftrightarrow \langle \bar{t}, \eta(y,\bar{x})\rangle \not\prec 0, \quad \forall y \in K,$$

which means that $\bar{x}$ is a solution to $(SVVLI)$. □

Remark. As mentioned before, if $Y = \mathbb{R}^m$, $C = \mathbb{R}^m_+$, then it can be shown that γ_S^L reduces to the gap function investigated in [15].

4.3 Generalized vector variational-like inequalities with set-valued mappings

Under the general suppositions as given in section 4.1 let $\eta : K \times K \to X$ and $h : K \times K \to Y$ be two vector-valued mappings satisfying $\eta(x,x) = 0$ and $h(x,x) = 0$, $\forall x \in K$, which are continuous with respect to the first variable for any fixed second variable in K. Let us consider the generalized vector variational-like inequality with set-valued mapping which consists in finding $\bar{x} \in K$ such that

$$(SGVVLI) \qquad \exists \bar{t} \in T(\bar{x}) : \quad \langle \bar{t}, \eta(y,\bar{x})\rangle + h(y,\bar{x}) \not\prec 0, \ \forall y \in K$$

and introduce the function

$$\begin{aligned}\gamma_S^{GL}(x) &= -\sup_{t \in T(x)} \inf_{y \in K} \Delta_C(-\langle t, \eta(y,x)\rangle - h(y,x)) \\ &= \inf_{t \in T(x)} \sup_{y \in K}\{-\Delta_C(-\langle t, \eta(y,x)\rangle - h(y,x))\}, \ x \in K.\end{aligned}$$

Analogously, we can verify the following assertion.

Theorem 4.3. *Assume that for each $x \in K$, $T(x)$ is nonempty and compact. Then γ_S^{GL} is a gap function for $(SGVVLI)$.*

Proof. The proof is similary to the proof of Theorem 4.2. □

Remark. If $Y = \mathbb{R}^m$, $C = \mathbb{R}^m_+$, then it is easy to verify that γ_S^{GL} can be reduced to the gap function investigated in [12].

Remark. For the readers who are interested in the existence of solutions to vector variational inequalities, we refer to [12] and [17].

References

1. Altangerel, L.; Boţ, R.I.; Wanka, G. *On the construction of gap functions for variational inequalities via conjugate duality,* Asia-Pacific Journal of Operational Research 24, no. 3, 353–371, 2007.
2. Altangerel, L.; Boţ, R.I.; Wanka, G. *Conjugate duality in vector optimization and some applications to the vector variational inequality,* Journal of Mathematical Analysis and Applications 329, no. 2, 1010–1035, 2007.
3. Altangerel, L. *Scalarized gap functions for vector variational inequalities via conjugate duality,* Mongolian Mathematical Journal 11, 46–54, 2007.
4. Boţ, R.I.; Grad, S.-M.; Wanka, G. *Duality in vector optimization,* Vector Optimization. Springer-Verlag, Berlin, 2009.
5. Chen, G.-Y.; Goh, C.-J.; Yang, X.Q. *On gap functions for vector variational inequalities,* Vector variational inequalities and vector equilibria, 55–72, Nonconvex Optim. Appl., 38, Kluwer Acad. Publ., Dordrecht, 2000.
6. Coulibaly, A.; Crouzeix, J.-P. *Condition numbers and error bounds in convex programming,* Mathematical Programming 116, no. 1-2, Ser. B, 79–113, 2009.
7. Ginchev, I.; Hoffmann, A. *Approximation of set-valued functions by single-valued one,* Discussiones Mathematicae. Differential Inclusions, Control and Optimization 22, no. 1, 33–66, 2002.
8. Ginchev, I.; Guerraggio, A.; Rocca, M. *From scalar to vector optimization,* Applications of Mathematics 51, no. 1, 5–36, 2006.
9. Hiriart-Urruty, J.-B. *Tangent cones, generalized gradients and mathematical programming in Banach spaces,* Mathematics of Operations Research 4, no. 1, 79–97, 1979.
10. Hiriart-Urruty, J.-B. *New concepts in nondifferentiable programming,* Société Mathématique de France. Bulletin. Mémoire no. 60, 57–85, 1979.
11. Konnov, I.V. *A scalarization approach for vector variational inequalities with applications,* Journal of Global Optimization 32, no. 4, 517–527, 2005.
12. Li, J. ; He, Z.-Q. *Gap functions and existence of solutions to generalized vector variational inequalities,* Applied Mathematics Letters 18, no. 9, 989–1000, 2005.
13. Li, J.; Mastroeni, G. *Vector variational inequalities involving set-valued mappings via scalarization with applications to error bounds for gap functions,* Journal of Optimization Theory and Applications 145, no. 2, 355–372, 2010.
14. Liu, C.G.; Ng, K.F.; Yang, W.H. *Merit functions in vector optimization,* Mathematical Programming 119, no. 2, Ser. A, 215–237, 2009.
15. Mishra, S.K.; Wang, S.Y.; Lai, K.K. *Gap function for set-valued vector variational-like inequalities,* Journal of Optimization Theory and Applications 138, no. 1, 77–84, 2008.
16. Taa, A. *Subdifferentials of multifunctions and Lagrange multipliers for multiobjective optimization,* Journal of Mathematical Analysis and Applications 28, no. 2, 398–415, 2003.
17. Yang, X.Q.; Yao, J.C. *Gap functions and existence of solutions to set-valued vector variational inequalities,* Journal of Optimization Theory and Applications 115, no. 2, 407–417, 2002.
18. Zaffaroni, A. *Degrees of efficiency and degrees of minimality,* SIAM Journal on Control and Optimization 42, no. 3, 1071–1086, 2003

Optimal Inscribing of Two Balls into Polyhedral Set

Rentsen Enkhbat and Bazarragchaa Barsbold

Abstract In this chapter, we consider the problem for optimal inscribing of two balls into bounded polyhedral set, so that sum of their radiuses is maximized. We formulate this problem as a bilevel programming problem and investigated its some properties. The gradient-based method for solving it has been proposed. We illustrate our approach on some test problems.

Key words Chebyshev center • Continuity • Differentiability • Bilevel programming problem

1 Introduction

We consider a problem of optimal inscribing of two balls into a polyhedral set. Such problem can be found in many applications such as facility location problem, cluster analysis, data mining, machine learning, regression analysis of models with bounded noise. On the other hand, the above problem generalizes the Chebyshev center problem in the case of two balls.

In [3] the Chebyshev center is shown to be a maximum likelihood estimator for the center of a uniform distribution over a k-sphere and both unbiased and consistent for the multivariate spherical normal distribution and any spherical finite range distribution. In the field of parameter estimation, the Chebyshev center approach

R. Enkhbat
The School of Economics Studies, National University of Mongolia,Ulaanbaatar, Mongolia
e-mail: renkhbat46@yahoo.com

B. Barsbold (✉)
The School of Mathematics and Computer Science, National University of Mongolia, Ulaanbaatar, Mongolia
e-mail: barsboldb@yahoo.com

A. Chinchuluun et al. (eds.), *Optimization, Simulation, and Control*, Springer Optimization and Its Applications 76, DOI 10.1007/978-1-4614-5131-0_3,

tries to find an estimator $\hat{x}$ for all x from the given feasibility set Q, such that $\hat{x}$ minimizes the worst possible estimation error for x (e.g., best worst-case estimate). The other direction in the bounded error estimation is to compute a specific estimate in the membership set enjoying some optimality properties [2]. A well-known estimate is the Chebyshev center [5] of the set Ω^n

$$\theta_c = \arg \min_{\theta \in \Omega^n} \max_{\eta \in \Omega^n} \|\theta - \eta\|,$$

where $\|\cdot\|$ is the l_p norm. This is the best worst-case estimate of the true but unknown system parameter vector in the sense that it minimizes the maximum distance between and the unknown parameter vector that generated the data. With $p = \infty$ or 1 , the calculation of θ is basically a linear programming problem.

The problem of estimating a vector z in the regression model $B = Az + w$, where w is an unknown but bounded noise, has been considered in [1]. To estimate z, a relaxation of the Chebyshev center, which is the vector that minimizes the worst-case estimation error over all feasible vectors z, was considered in [1]. It is shown that the estimate can be viewed as a Tikhonov regularization with a special choice of parameter that can be found efficiently by solving a convex optimization problem with two variables or a semidefinite program with three variables, regardless of the problem size. When the norm constraint on z is a Euclidean one, the problem reduces to a single-variable convex minimization problem [1]. The chapter is organized as follows. In Sect. 2, we consider the problem of optimal inscribing of two balls into a polyhedral set. Section 3 is devoted to some properties of proposed auxiliary functions. In Sect. 4, some test problems have been solved numerically.

2 Optimal Inscribing of Two Balls

We formulate the problem of optimal subdivision of a bounded polyhedral set, so that sum of radiuses of inscribed balls is maximized. For this purpose, we need to introduce the following conventions. Let A be an $m \times n$ matrix, $b \in \mathbb{R}^m$, and $D = \{x \in \mathbb{R}^n : Ax \le b,\ x \ge 0\}$. For all $c \in R^n$, $c \ne 0$, and $z \in D$ hyperplane $c^T(x - z) = 0$ subdivides D into the following two parts:

$$D_1(c,z) = \{x \in \mathbb{R}^n : Ax \le b,\ c^T x \le c^T z,\ x \ge 0\}, \tag{1}$$

$$D_2(c,z) = \{x \in \mathbb{R}^n : Ax \le b,\ c^T x \ge c^T z,\ x \ge 0\}. \tag{2}$$

Let $v(A) = (\|A_1\|, \ldots, \|A_m\|)^T$,
where $\|A_i\| = \sqrt{a_{i1}^2 + \cdots + a_{in}^2}$ for all $i = 1, \ldots, m$ and

$$X_1(c,z) = \{(x,r) \in \mathbb{R}^{n+1} : Ax + v(A)r \le b,\ c^T x + \|c\| r \le c^T z,$$
$$r \ge 0,\ x \ge 0\}, \tag{3}$$

$$X_2(c,z) = \{(x,r) \in \mathbb{R}^{n+1} : Ax + v(A)r \le b,\ c^T x - \|c\| r \ge c^T z,$$
$$r \ge 0,\ x \ge 0\}. \tag{4}$$

Introduce the following auxiliary functions:

$$r_1(c,z) = \max_{(x,r_1)\in X_1(c,z)} r_1. \tag{5}$$

$$r_2(c,z) = \max_{(x,r_2)\in X_2(c,z)} r_2. \tag{6}$$

Then, the problem for optimally inscribing of two balls into a polyhedral set is formulated as follows:

$$\max\ \Gamma(c,z) = r_1(c,z) + r_2(c,z) \tag{7}$$

$$\text{s.t.} \begin{cases} \max\limits_{(x,r_1)\in X_1(c,z)} r_1, \\ \max\limits_{(x,r_2)\in X_2(c,z)} r_2, \end{cases} \tag{8}$$

where $c \in \mathbb{R}^n$, $c \ne 0$, and $z \in D$.
We denote by $Y_1(c)$ and $Y_2(c)$ constraints of dual problems to (5) and (6), respectively, i.e.,

$$Y_1(c) = \{y \in \mathbb{R}^{m+1} : a_{1j}y_1 + \cdots + a_{mj}y_m + c_j y_{m+1} \ge 0,\ \forall j = 1,\ldots,n$$
$$\|a_1\|y_1 + \cdots + \|a_m\|y_m + \|c\|y_{m+1} \ge 1,\ y_1,\ldots,y_{m+1} \ge 0\}, \tag{9}$$

$$Y_2(c) = \{y \in \mathbb{R}^{m+1} : a_{1j}y_1 + \cdots + a_{mj}y_m - c_j y_{m+1} \ge 0,\ \forall j = 1,\ldots,n$$
$$\|a_1\|y_1 + \cdots + \|a_m\|y_m + \|c\|y_{m+1} \ge 1,\ y_1,\ldots,y_{m+1} \ge 0\}. \tag{10}$$

Then dual problems to (5) and (6) are given by

$$\Omega_1(z) = \min\ b^T y + c^T z y_{m+1} \tag{11}$$

$$\text{s.t.} \begin{cases} A^T y + c y_{m+1} \ge 0, \\ v^T(A)y + \|c\| y_{m+1} \ge 1, \\ y \ge 0,\ y_{m+1} \ge 0 \end{cases} \tag{12}$$

and

$$\Omega_2(z) = \min\ b^T y - c^T z y_{m+1} \tag{13}$$

$$\text{s.t.} \begin{cases} A^T y - c y_{m+1} \ge 0, \\ v^T(A)y + \|c\| y_{m+1} \ge 1, \\ y \ge 0,\ y_{m+1} \ge 0. \end{cases} \tag{14}$$

Let us establish some basic properties of function Γ.

Proposition 1. *1. Function $\Gamma(\cdot,z):\mathbb{R}^n\to\mathbb{R}$ is homogeneous of degree zero.*
2. Functions $r_1(c,\cdot):D\to\mathbb{R}$ and $r_2(c,\cdot):D\to\mathbb{R}$ are concave on D for all $c\in\mathbb{R}^n\backslash\{0\}$.

Proof. 1. Clearly, $X_i(\alpha c,z)=X_i(c,z)$ hold for all $\alpha\in\mathbb{R}$, $\alpha\neq 0$, $c\in\mathbb{R}^n$, $c\neq 0$, *and* $i=1,2$. This implies that $\Gamma(\alpha c,z)=\Gamma(c,z)$ for all $\alpha\in\mathbb{R}$, $\alpha\neq 0$, and $c\in\mathbb{R}^n$, $c\neq 0$, which is our claim.

2. Now we show that $r_1(c,\cdot)$ is concave on D. Problem (11) is dual to (5). According to strong duality theorem [6], we have $r_1^*=b^Ty^*+c^Tzy_{m+1}^*$, where r_1^* is an optimal value of (5) and (y^*,y_{m+1}^*) is an optimal solution to (11). This follows that

$$r_1(c,z)=\Omega_1(z) \tag{15}$$

for all $c\in\mathbb{R}^n$, $c\neq 0$, and $z\in D$. By definition of $\Omega_1(z)$, it is concave as a minimum to family of linear functions over convex set. The latter and (15) imply that function $r_1(c,\cdot)$ is concave.

Concavity of $r_2(c,\cdot)$ can be shown easily by analogy to the above. By considering problem (13), we show that the function $\Omega_2(z)$ is concave as the minimum to family of linear functions. On the other hand, we have $\Omega_2(z)=r_2(c,z)$. Since both $r_1(c,\cdot)$ and $r_2(c,\cdot)$ are concave, their sum is concave, too. This completes the proof. □

It is well known [4] that concavity of a function implies its continuity on interior of its domain. Therefore, Proposition 1.2 implies continuity of functions $r_1(c,\cdot)$ and $r_2(c,\cdot)$ with respect to z for all $c\in\mathbb{R}^n\backslash\{0\}$. However, it does not guarantee continuity of the functions with respect to all pairs (c,z), where $c\in\mathbb{R}^n\backslash\{0\}$. In the next proposition, we establish continuity of the above function. First, we introduce some notations.

Let $\{c^k\}_{k=1}^{\infty}$ be a sequence convergent to c, i.e., $\lim\limits_{k\to\infty}c^k=c$, $\{z^k\}_{k=1}^{\infty}$ be convergent to z, $z\in X_1(c,z)$, and $z^k\in X_1(c^k,z^k)$ for all $k=1,2,\ldots$. We denote by (x^*,r_1^*) a solution to problem (5) and let a pair (y^*,y_{m+1}^*) be a solution to its dual problem given by (11). Similarly, we denote by

$$\begin{pmatrix}x^k\\ r_1^k\end{pmatrix}\quad\text{and}\quad\begin{pmatrix}y^k\\ y_{m+1}^k\end{pmatrix} \tag{16}$$

solutions to primal and dual problems defined by

$$\max_{(x,\,r_1)\in X_1(c^k,\,z^k)} r_1\ \text{and}\ \min_{(y,\,y_{m+1})\in Y_1(c^k)} b^Ty+(c^k)^Tz^ky_{m+1} \tag{17}$$

for all $k=1,2,\ldots$.

Lemma 1. *1. If D is nonempty and bounded, then $X_1(c,z)$ is nonempty and bounded for all $c\in\mathbb{R}^n\backslash\{0\}$ and $z\in D$.*

2. *If D is nonempty and bounded, then sequences $\{x^k\}_{k=0}^{\infty}$ and $\{r^k\}_{k=0}^{\infty}$ defined by (16) are bounded.*
3. *If $Y_1(c)$ is bounded, then sequence $\{y_{m+1}^k\}_{k=0}^{+\infty}$ defined by (16) is bounded.*

Proof. 1. Let the set D be nonempty and bounded. Then for all $c \in \mathbb{R}^n \backslash \{0\}$ and $z \in D$, there exists x_* providing the minimum to

$$\min_{x \in D} c^T x. \tag{18}$$

By definition of $D_1(c,z)$ given by (1), we have $x_* \in D_1(c,z)$. Therefore, $D_1(c,z)$ is nonempty. Since $D_1(c,z) \subseteq D$, it is also bounded. Since $D_1(c,z)$ is nonempty and bounded, problem (5) has a solution and all of them are bounded. Then as a constraint to (5), $X_1(c,z)$ is bounded and nonempty. This proves Part 1 of the lemma.

2. First, we show that $\{x^k\}_{k=0}^{\infty}$ is bounded. Since $x^k \in X_1(c^k, z^k)$, we have

$$a_i^T x^k + \|a_i\| r_1^k \le b_i, \ \text{ for all } i = 1, \dots, m. \tag{19}$$

This implies

$$a_i^T x^k \le b_i, \ \text{ for all } i = 1, \dots, m,$$

which means

$$x^k \in D, \quad \forall\, k = 0, 1, 2, \dots \ . \tag{20}$$

Boundedness and (20) imply that sequence $\{x^k\}_{k=0}^{\infty}$ is bounded. Now, let us prove that $\{r^k\}_{k=0}^{\infty}$ is bounded. Since $r^k \ge 0$, it is sufficient to show that the sequence is bounded above. On the contrary to this, we assume that there exists subsequence $\{k_j\}_{j=0}^{\infty}$ such that

$$\lim_{j \to +\infty} r^{k_j} = +\infty. \tag{21}$$

Since $\{x^k\}_{k=0}^{\infty}$ is bounded, then we have

$$\lim_{j \to +\infty} \left(a_i^T x^{k_j} + \|a_i\| r_1^{k_j} \right) = +\infty, \quad \forall i = 1, \dots, m. \tag{22}$$

Taking into account this and (19), we obtain $b_i = +\infty, \quad \forall i = 1, \dots, m$, which contradicts the boundedness of D. This proves Part 2 of the lemma.

3. Let denote by L a set of limit points of $\{y^k\}_{k=0}^{+\infty}$. Let $\bar{y} \in L$. Then there exists $\{k_i\}_{i=0}^{+\infty}$ such that $\lim\limits_{i \to +\infty} y^{k_i} = \bar{y}$, and from definition of $Y_1(c)$ given by (9), it follows that $\bar{y} \in Y_1(c)$. If $Y_1(c)$ is bounded then $\{y^k\}_{k=0}^{+\infty}$ has bounded limit set, so sequence $\{y_{m+1}^k\}_{k=0}^{+\infty}$ is bounded, too. This proves the lemma. □

Lemma 2.

$$y^*_{m+1}[c^T z-(c^k)^T z^k-(\|c\|-\|c^k\|)r^k_1-(c^T-(c^k)^T)x^k]$$
$$\le r^*_1-r^k_1 \tag{23}$$
$$\le y^k_{m+1}[c^T z-(c^k)^T z^k-(c^T-(c^k)^T)x^*-(\|c\|-\|c^k\|)r^*_1]. \tag{24}$$

Proof. Since $(x^*,r^*_1)\in X_1(c,z)$, $y^k\ge 0$, and $y^k_{m+1}\ge 0$,

$$\sum_{i=1}^{m} y^k_i(b_i-a^T_i x^*-\|a_i\|r^*_1)+y^k_{m+1}(c^T z-c^T x^*-\|c\|r^*_1)\ge 0 \tag{25}$$

hold for all $(y^k,y^k_{m+1})^T\in Y_1(c^k)$. Due to strong duality theorem [6], $r^k_1=\sum\limits_{i=1}^{m} y^k_i b_i+y^k_{m+1}(c^k)^T z^k$ takes place. Moreover, we have dual feasibility given by $\sum\limits_{i=1}^{m} y^k_i a_{ij}+c^k_j y^k_{m+1}\ge 0,\quad \forall j=1,\dots,n$, and $\sum\limits_{i=1}^{m}\|a_i\|y^k_i+\|c^k\|y^k_{m+1}\ge 1$. Applying these results to (25), we obtain the following estimation:

$$\begin{aligned}
r^*_1-r^k_1 &\le r^*_1-r^k_1+\sum_{i=1}^{m} y^k_i(b_i-a^T_i x^*-\|a_i\|r^*_1)\\
&\quad+y^k_{m+1}(c^T z-c^T x^*-r^*_1\|c\|)\\
&=r^*_1-r^k_1+\sum_{i=1}^{m} y^k_i(b_i-a^T_i x^*-\|a_i\|r^*_1)+y^k_{m+1}[c^T z-(c^k)^T z^k\\
&\quad+(c^k)^T z^k-c^T x^*-\|c\|r^*_1]\\
&=\left[\sum_{i=1}^{m} y^k_i b_i+y^k_{m+1}(c^k)^T z^k-r^k_1\right]+r^*_1-\sum_{i=1}^{m} y^k_i(a^T_i x^*+\|a_i\|r^*_1)\\
&\quad-y^k_{m+1}(c^T x^*+\|c\|r^*_1)+y^k_{m+1}(c^T z-(c^k)^T z^k)
\end{aligned} \tag{26}$$

$$\begin{aligned}
&=r^*_1-\sum_{i=1}^{m} y^k_i(a^T_i x^*+\|a_i\|r^*_1)-y^k_{m+1}(c^T x^*+\|c\|r^*_1)\\
&\quad+y^k_{m+1}[c^T z-(c^k)^T z^k]\\
&=r^*_1-\sum_{i=1}^{m} y^k_i(a^T_i x^*+\|a_i\|r^*_1)-y^k_{m+1}((c^k)^T x^*+\|c^k\|r^*_1)+y^k_{m+1}[c^T z\\
&\quad-(c^k)^T z^k-(c^T-(c^k)^T)x^*-(\|c\|-\|c^k\|)r^*_1]
\end{aligned}$$

$$
\begin{aligned}
&= r_1^* - \sum_{i=1}^{m}\sum_{j=1}^{n} y_i^k a_{ij} x_j^* - \sum_{i=1}^{m} \|a_i\| y_i^k r_1^* - \sum_{j=1}^{n} y_{m+1}^k c_j^k x_j^* - y_{m+1}^k \|c^k\| r_1^* \\
&\quad + y_{m+1}^k [c^T z - (c^k)^T z^k - (c^T - (c^k)^T) x^* - (\|c\| - \|c^k\|) r_1^*] \\
&= r_1^* - \sum_{j=1}^{n} \left(\sum_{i=1}^{m} y_i^k a_{ij} + c_j^k y_{m+1}^k \right) x_j^* - \left(\sum_{i=1}^{m} \|a_i\| y_i^k + \|c^k\| y_{m+1}^k \right) r_1^* \\
&\quad + y_{m+1}^k [c^T z - (c^k)^T z^k - (c^T - (c^k)^T) x^* - (\|c\| - \|c^k\|) r_1^*] \\
&\le r_1^* - r_1^* + y_{m+1}^k [c^T z - (c^k)^T z^k - (c^T - (c^k)^T) x^* - (\|c\| - \|c^k\|) \cdot r_1^*] \\
&= y_{m+1}^k [c^T z - (c^k)^T z^k - (c^T - (c^k)^T) x^* - (\|c\| - \|c^k\|) r_1^*]. \qquad (27)
\end{aligned}
$$

In similar way to (25), we have

$$
\sum_{i=1}^{m} y_i^* (b_i - a_i^T x^k - \|a_i\| r_1^k) + y_{m+1}^* ((c^k)^T z^k - (c^k)^T x^k - \|c^k\| r_1^k) \ge 0. \qquad (28)
$$

Due to above, we obtain

$$
\begin{aligned}
r_1^k - r_1^* \le r_1^k - r_1^* &+ \sum_{i=1}^{m} y_i^* (b_i - a_i^T x^k - \|a_i\| r_1^k) \\
&+ y_{m+1}^* ((c^k)^T z^k - (c^k)^T x^k - \|c^k\| r_1^k). \qquad (29)
\end{aligned}
$$

Further, in analogy to (26)–(27), we have the following estimation:

$$
r_1^k - r_1^* \le y_{m+1}^* [c^T z - (c^k)^T z^k + ((c^k)^T - c^T) x^k + (\|c^k\| - \|c\|) r_1^k]. \qquad (30)
$$

Based on (26), (27) and (30), we have (23)–(24). □

3 Continuity and Differentiability of Auxiliary Functions

Proposition 2 (Continuity of r_1 and r_2). *Let D be nonempty and bounded and $Y_1(c)$ be bounded. Then functions $r_1 : \mathbb{R}^n \times [0;1] \to \mathbb{R}$ and $r_2 : \mathbb{R}^n \times [0;1] \to \mathbb{R}$ given by (5) and (6) are continuous for all $c \in \mathbb{R}^n \backslash 0$ and $z \in D$.*

Proof. It is sufficient to prove continuity of r_1. Continuity of r_2 can be shown by analogy. We need to prove that $\lim\limits_{k \to +\infty} r_1^k = r_1^*$. Based on (23)–(24) we have

$$y^*_{m+1}[c^T z-(c^k)^T z^k-(\|c\|-\|c^k\|)r^k_1-(c^T-(c^k)^T)x^k]$$
$$\leq r^*_1-r^k_1 \tag{31}$$
$$\leq y^k_{m+1}[c^T z-(c^k)^T z^k-(c^T-(c^k)^T)x^*-(\|c\|-\|c^k\|)r^*_1]. \tag{32}$$

Since D and $Y_1(c)$ are bounded, from Lemma 1 points 2 to 3 it implies that $\{(x^k,r^k_1)\}^{\infty}_{k=1}$ and $\{y^k_{m+1}\}^{\infty}_{k=1}$ are bounded. Taking into account this and letting $k\to\infty$ in (23)–(24), we obtain

$$\lim_{k\to\infty}(r^*_1-r^k_1)=0. \tag{33}$$

This completes the proof. □

Let $c,h,v\in\mathbb{R}^n, z\in D$, and $c(t)=c+th$, $z(t)=z+tv$ for all sufficiently small $t\in\mathbb{R}$ satisfying $z(t)\in D$. Vectors

$$\left[\begin{pmatrix}x(t)\\ r_1(t)\end{pmatrix},\begin{pmatrix}y(t)\\ y_{m+1}(t)\end{pmatrix}\right]$$

denote primal and dual solutions to problem

$$\max\ r_1$$
$$\begin{cases}Ax+v(A)r_1\leq b\\ c^T(t)x+\|c(t)\|r_1\leq c^T(t)z(t)\\ r_1\geq 0\end{cases} \tag{34}$$

for given $t\in\mathbb{R}$ with $z(t)\in D$.

Proposition 3 (Directional Differentiability). *If primal and dual problems (5) and (11) have a unique solution*[1] *for $c\in\mathbb{R}^n\backslash 0$ and $z\in\mathbb{R}^n$, then there exists directional derivative of r_1 along $\tilde{h}=(h^T,v^T)^T$, and it is given by*

$$\frac{\partial r_1}{\partial\tilde{h}}(c,z)=y^*_{m+1}[c^T v+z^T h-\frac{c^T h}{\|c\|}r^*_1-h^T x^*]. \tag{35}$$

Proof. Due to Lemma 2, we have

$$y_{m+1}(t)[c^T(t)z(t)-c^T z-(\|c(t)\|-\|c\|)r^*_1-th^T x^*]$$
$$\leq r_1(t)-r^*_1 \tag{36}$$
$$\leq y^*_{m+1}[c^T(t)z(t)-c^T z-(\|c(t)\|-\|c\|)r_1(t)-th^T x(t)]. \tag{37}$$

[1]This means that primary and dual problems (5) and (11) are nondegenerated.

It holds

$$c^T(t)z(t) - c^T z = tc^T v + tz^T h + t^2 h^T v. \tag{38}$$

We substitute it into (36)–(37) and obtain

$$\begin{aligned} & y_{m+1}(t)[tc^T v + tz^T h + t^2 h^T v - (\|c(t)\| - \|c\|)r_1^* - th^T x^*] \\ & \quad \le r_1(t) - r_1^* \qquad (39) \\ & \quad \le y_{m+1}^*[tc^T v + tz^T h + t^2 h^T v - (\|c(t)\| - \|c\|)r_1(t) - th^T x(t)]. \qquad (40) \end{aligned}$$

If problems (5) and (11) have a unique solution, then

$$\lim_{t\to 0} x(t) = x^* \quad \text{and} \quad \lim_{t\to 0} y_{m+1}(t) = y_{m+1}^*. \tag{41}$$

Taking into account this, dividing (39)–(40) by t and letting $t \to 0$, we obtain (35). □

From (35) it implies that directional derivative is continuous for all (c,z), unless $c = 0$. Then we have the following formula for the gradient of r_1:

Corollary 1 (Gradient Formula for r_1).

$$\nabla r_1(c,z) = y_{m+1}^* \begin{bmatrix} z - \frac{c}{\|c\|} r_1^* - x^* \\ c \end{bmatrix}. \tag{42}$$

We denote solutions to primal and dual problems (6) and (13) by

$$\begin{pmatrix} \bar{x}^* \\ r_2^* \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \bar{y}^* \\ \bar{y}_{m+1}^* \end{pmatrix},$$

respectively. In order to derive gradient formula for function $r_2(c,z)$ given by (6), we consider the following problem:

$$\begin{array}{l} \max\ r_2 \\ \begin{cases} Ax + v(A)r_2 \le b \\ c^T(t)x - \|c(t)\|r_2 \le c^T(t)z(t) \\ r_2 \ge 0 \end{cases} \end{array} \tag{43}$$

for $c(t) = c - th$ and $z(t) = z + tv$, where $t \in \mathbb{R}$ is given so that it holds $z(t) \in D$ and $c, h, z, v \in \mathbb{R}^n$. Let

$$\begin{pmatrix} \bar{x}(t) \\ r_2(t) \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \bar{y}(t) \\ \bar{y}_{m+1}(t) \end{pmatrix}$$

be primal and dual solutions.

Proposition 4 (Directional Differentiability of r_2). *If primal and dual problems (6) and (13) have unique primal and dual solutions for all $c \in \mathbb{R}^n \backslash 0$ and $z \in \mathbb{R}^n$, then there exists directional derivative of r_2 along $\tilde{h}$, where $\tilde{h} = (h^T, v^T)^T$, and it is given by*

$$\frac{\partial r_2}{\partial \tilde{h}}(c,z) = -\bar{y}^*_{m+1}\left[c^T v + z^T h + \frac{c^T h}{\|c\|} r_1^* - h^T x^*(c,z)\right]. \tag{44}$$

Proof. By analogy to the proof of Proposition 3. □

The directional derivative is continuous for all (c,z), whenever $c \neq 0$. Then we have the following formula for the gradient of r_2.

Corollary 2 (Gradient Formula for r_2).

$$\nabla_{\tilde{h}} r_2(c,z) = -\bar{y}^*_{m+1}\begin{bmatrix} z + \frac{c}{\|c\|} r_2^* - \bar{x}^* \\ c \end{bmatrix}. \tag{45}$$

We note that assumptions on uniqueness of solutions x^*, y_{m+1} and $\bar{x}, \bar{y}_{m+1}$ in Propositions 3 and 4 are necessary.

4 Numerical Examples

Example 1. Consider a rectangle defined by

$$D = \{x \in \mathbb{R}^2 : 0 \le x_1 \le 4,\ 0 \le x_2 \le 2\}. \tag{46}$$

Obviously, we can enclose two equal balls into it with radiuses $r_1 = r_2 = 1$. This solves problems (7)–(8) over the rectangle providing the maximum to the objective function setting $r_1^* + r_2^* = 2$. On the other hand, we can also, enclose two equal balls with radiuses $r_1 = r_2 = 0.5$ which provides the minimum to the objective function for $r_1 + r_2 = 1$. These solutions were obtained for $c^* = (1,0)^T$, $z^* = (2,1)^T$ for the maximum and for $c_* = (0,1)^T$, $z_* = (2,1)^T$ in the case of the minimum. In neighborhood of c_* functions r_1 and r_2 are nondifferentiable, because we find

$$x^* = t\left(0, \frac{1}{2}\right)^T + (1-t)\left(4, \frac{1}{2}\right)^T \tag{47}$$

$$\bar{x} = t\left(0, \frac{3}{2}\right)^T + (1-t)\left(4, \frac{3}{2}\right)^T, \tag{48}$$

Table 1 Results of numerical experiment with accuracy $\varepsilon = 10^{-5}$

n	r_c	r_1^0	r_2^0	$\nabla f(c^0, z^0)$	r_1^*	r_2^*	$f(c^*, z^*)$	$\nabla f(c^*, z^*)$	Time
10.0	4.7917	3.6405	4.3576	0.49258	4.7833	4.4303	9.2136	0.000000011147	16.739
15.0	4.3712	3.4742	4.0768	0.33946	4.3050	4.1504	8.4554	0.000000092872	14.258
20.0	4.1143	3.3624	3.8890	0.26124	4.1111	3.9537	8.0648	0.000000024780	35.350
25.0	3.9346	3.2788	3.7509	0.21364	3.8949	3.8111	7.7060	0.00000064499	27.378
30.0	3.7989	3.2124	3.6429	0.18152	3.7658	3.6981	7.4639	0.000000018056	29.406
35.0	3.6912	3.1575	3.5552	0.15834	3.6628	3.6062	7.2690	0.0000000049867	36.863
40.0	3.6027	3.1108	3.4817	0.14078	3.5778	3.5293	7.1071	0.000000028025	46.441
45.0	3.5280	3.0703	3.4189	0.12700	3.5084	3.5083	7.0167	0.000000056647	75.021
50.0	3.4638	3.0346	3.3641	0.11589	3.4439	3.4062	6.8501	0.00000016694	74.709

where $t \in [0, 1]$ and there is no guarantee that

$$\lim_{k\to 0} x(t) = x^* \text{ and } \lim_{k\to 0} \bar{x}(t) = \bar{x}. \tag{49}$$

Example 2. Consider a triangle defined by

$$D = \{x \in \mathbb{R}^2 : x_1 + 2x_2 \le 10,\ x_1, x_2 \ge 0\}. \tag{50}$$

A ball inscribed into it has radius given by $r = 1.9098$ and center at $x = (1.9098, 1.9098)^T$. Based on gradient formulas (42) and (45) we employed quasi-Newton algorithm to solve problems (7)–(8) over D given by (50). Starting with $c = (1, 1)^T$, the algorithm has found a globally optimal subdivision of the triangle. Balls inscribed into the optimal subdivision are given by $x^* = (1.9098, 1.9098)^T$, $r_1^* = 1.9098$, $\bar{x}^* = (5.3507, 1.0976)$, and $r_2^* = 1.0976$, and sum of their radiuses satisfies $r_1^* + r_2^* = 3.0074$.

Example 3. We define D as a simplex set given by

$$D = \left\{x \in \mathbb{R}^n : \frac{x_1}{20} + \frac{x_2}{40} + \cdots + \frac{x_n}{20n} \le 1,\ x_1, x_2, \ldots, x_n \ge 0\right\} \tag{51}$$

for given $n \in \mathbb{R}$. In this case, we had solved numerically the optimal inscribing problem for $n = 10, \ldots, 200$. Numerical experiments were made on PC with Intel Core 2 Duo, 2.93 GHz processor and 2,038 MB RAM using Matlab implementation of quasi-Newton method. We continued iteration until decreasing of the gradient norm below given precision ε. Results of the experiments are presented in Tables 1 and 2, where n stands for problem size, r_c is radius of the ball inscribed into D, r_1^0 and r_2^0 are radiuses of the initially inscribed balls into D, and $\nabla f(c^0, z^0)$ is a norm of the gradient at the initial subdivision. Columns labeled by r_1^*, r_2^*, $f(c^*, z^*)$, and $\nabla f(c^*, z^*)$ contain approximations to radiuses of the optimally inscribed balls, optimal value and gradient of the objective function at the optimal subdivision. In the last column we listed computation time measured in seconds.

Table 2 Results of numerical experiment with accuracy $\varepsilon = 10^{-4}$

n	r_c	r_1^0	r_2^0	$\nabla f(c^0, z^0)$	r_1^*	r_2^*	$f(c^*, z^*)$	$\nabla f(c^*, z^*)$	Time
10	4.7917	3.6405	4.3576	0.49258	4.6916	4.4366	9.1282	0.000040117	9.7345
15	4.3712	3.4742	4.0768	0.33946	4.3050	4.1504	8.4554	0.000000092872	14.274
20	4.1143	3.3624	3.8890	0.26124	4.0647	3.9554	8.0201	0.000017258	19.485
25	3.9346	3.2788	3.7509	0.21364	3.8949	3.8111	7.7060	0.00000064499	26.770
30	3.7989	3.2124	3.6429	0.18152	3.7658	3.6981	7.4639	0.000000018056	28.330
35	3.6912	3.1575	3.5552	0.15834	3.6628	3.6062	7.2690	0.0000000049867	36.785
40	3.6027	3.1108	3.4817	0.14078	3.5778	3.5293	7.1071	0.000000028025	46.317
45	3.5280	3.0703	3.4189	0.12700	3.5084	3.5083	7.0167	0.000000056647	74.912
50	3.4638	3.0346	3.3641	0.11589	3.4439	3.4062	6.8501	0.00000016694	74.319
55	3.4077	3.0028	3.3158	0.10672	3.3895	3.3558	6.7453	0.0000046109	75.536
60	3.3580	2.9740	3.2727	0.099018	3.3413	3.3107	6.6520	0.0000000028635	108.75
65	3.3135	2.9479	3.2339	0.092454	3.2995	3.2979	6.5974	0.00000040210	2911.8
70	3.2734	2.9239	3.1986	0.086788	3.2591	3.2334	6.4925	0.0000051336	120.50
75	3.2369	2.9018	3.1663	0.081843	3.2270	3.2270	6.4540	0.0000015405	166.87
80	3.2035	2.8814	3.1366	0.077489	3.1909	3.1688	6.3597	0.00000033370	173.61
85	3.1727	2.8623	3.1091	0.073622	3.1609	3.1402	6.3011	0.0000000014511	220.16
90	3.1443	2.8444	3.0836	0.070163	3.1331	3.1136	6.2467	0.0000000032898	225.48
95	3.1178	2.8277	3.0598	0.067050	3.1072	3.0889	6.1960	0.0000017390	272.42
100	3.0931	2.8119	3.0375	0.064232	3.0830	3.0657	6.1487	0.00000030525	284.66
105	3.0699	2.7970	3.0166	0.061669	3.0603	3.0439	6.1043	0.000000018861	344.82
110	3.0482	2.7829	2.9968	0.059325	3.0390	3.0234	6.0624	0.000000014225	413.06
115	3.0277	2.7695	2.9782	0.057174	3.0189	3.0041	6.0230	0.000000023488	456.19
120	3.0083	2.7567	2.9605	0.055192	2.9999	2.9858	5.9856	0.000000083348	502.93
125	2.9900	2.7445	2.9437	0.053360	2.9819	2.9684	5.9503	0.000014221	538.92
130	2.9726	2.7329	2.9278	0.051660	2.9663	2.9662	5.9326	0.0000018209	618.78
135	2.9560	2.7218	2.9125	0.050078	2.9485	2.9361	5.8845	0.000014718	790.63
140	2.9402	2.7111	2.8980	0.048603	2.9330	2.9210	5.8540	0.00000037802	862.19
145	2.9251	2.7009	2.8841	0.047223	2.9209	2.9209	5.8418	0.0000053246	1176.6
150	2.9107	2.6910	2.8708	0.045929	2.9039	2.8929	5.7968	0.00000054939	1226.9
155	2.8969	2.6815	2.8580	0.044714	2.8903	2.8797	5.7700	0.0000014864	1249.9
160	2.8837	2.6724	2.8458	0.043569	2.8800	2.8800	5.7599	0.00000023966	1318.1
165	2.8709	2.6636	2.8340	0.042490	2.8647	2.8548	5.7195	0.00000099447	1080.5
170	2.8587	2.6551	2.8226	0.041470	2.8527	2.8430	5.6957	0.00000000038773	1422.5
175	2.8469	2.6468	2.8116	0.040504	2.8410	2.8317	5.6727	0.000000025851	1563.4
180	2.8355	2.6388	2.8010	0.039588	2.8298	2.8208	5.6506	0.00000060155	1636.1
185	2.8246	2.6311	2.7908	0.038718	2.8204	2.8204	5.6408	0.0000058924	2496.3
190	2.8140	2.6236	2.7809	0.037891	2.8086	2.8000	5.6086	0.00000000038547	1656.0
195	2.8037	2.6164	2.7714	0.037104	2.7985	2.7902	5.5886	0.0000000065744	2328.8
200	2.7938	2.6093	2.7621	0.036353	2.7887	2.7806	5.5693	0.0000016114	2433.4

References

1. Amir Beck and Yonina C. Eldar, *Regularization in Regression with Bounded Noise: A Chebyshev Center Approach* SIAM. J. Matrix Anal. and Appl. Volume 29, Issue 2, pp. 606–625 (2007) Published May 1, 2007

2. Er-Wei Bai, Yinyu Ye, and Roberto Tempo *Bounded Error Parameter Estimation: A Sequential Analytic Center Approach* IEEE Transactions on Automatic Control, Vol. 44, No. 6, pp. 1107–1117 June 1999
3. Halteman, Edward J. *The Chebyshev center: A multidimensional estimate of location.* J. Stat. Plann. Inference 13, 389–394 (1986).
4. Tyrrell R. Rockafellar. *Convex analysis*. Number 28 in Princeton Mathematical Series. Princeton University Press, Princeton, N.J., 1970.
5. J. F. Traub, G. Wasikowski, and H. Wozniakowski, *Information-Based Complexity.* New York: Academic, 1988.
6. Yudin DB, Gol'shtein EG, *Linear programming*, Israel Program of Scientific Translations, Jerusalem, 1965

Mathematical Programs with Equilibrium Constraints: A Brief Survey of Methods and Optimality Conditions

Ider Tseveendorj

Abstract This chapter provides a short survey of the research for an important class of constrained optimization problems for which their constraints are defined in part by a variational inequality. Such problems are known as mathematical programs with equilibrium constraints (MPEC). MPEC arise naturally in different areas and play an important role, for example, in the pricing of telecommunication and transportation networks, in economic modeling, in computational mechanics in many other fields of modern optimization, and have been the subject of a number of recent studies. We present a general formulation of MPEC, describe the main characteristics of MPEC, and review the main properties and theoretical results for these problems. The short survey mainly concentrates on the review of the available solution methodology.

Key words Mathematical programs • Equilibrium constraints • Variational inequality • Bilevel optimization • Nonconvex optimization • Nondifferentiable optimization

Introduction

Mathematical programming is a field extensively studied by many researchers. Due to its potential for application in real-world problems it has prospered over the last few decades. Great strides have been made recently in the solution of large-scale mathematical programming problems arising in different practical areas, particularly in telecommunication. It is well known that the "classical" telecommunication problems like the minimum cost flow, the multicommodity flow,

I. Tseveendorj (✉)
Laboratory PRISM, Computer Science Department, University of Versailles, St.-Quentin-en-Yvelines, 45 avenue des États-Unis, 78035 Versailles Cedex, France
e-mail: ider.tseveendorj@prism.uvsq.fr

A. Chinchuluun et al. (eds.), *Optimization, Simulation, and Control*, Springer Optimization and Its Applications 76, DOI 10.1007/978-1-4614-5131-0_4,

and network design, to name a few, can be formulated as mathematical programming problems, and for solving these problems, several efficient algorithms have already been proposed [2].

But as far as a real-world decision making is concerned, the decision maker often has to deal with a reaction of other decision makers. For example, in problem of pricing in telecommunication, a reaction of clients plays an important role in the total revenue. Analysts have found, however, that standard mathematical programming models are often inadequate in such situations because more than a single objective function and single decision maker are involved. Such kind of problems are formulated as a mathematical program with equilibrium constraints.

We consider the MPEC in the following statement:

$$\begin{cases} \text{minimize } \varphi(x,y) \\ \text{subject to } x \in \Omega, \\ \qquad\qquad y \in S(x), \end{cases} \tag{MPEC}$$

where $\varphi : R^{n+m} \to R$, $\Omega \subset R^n$, and $S(x)$ is the solution set of the reactions of the other decision makers (an equilibrium constraint), that is usually formulated in terms of a variational inequality.

We recall some definitions and results from the theory of the variational inequality problems in Sect. 1. The general MPEC is a highly nonconvex, nondifferentiable optimization problem that encompasses certain combinatorial features in its constraints. As such, it is computationally very difficult to solve, especially if one wishes to compute a global optimal solution.

MPEC arise naturally in different areas and play an important role, for example, in the design of telecommunication and transportation networks, in economic modeling, in computational mechanics in many other fields of modern optimization, and have been the subject of a number of recent studies. They also include, as a special case, the bilevel optimization problem, where some variables are restricted to be in the solution set of another parametric optimization problem [4].

An extensive bibliography on this topic and its application can be found in the monographs [13, 19]. In this chapter we intend to give an overview over the literature in the field: we are interested only in basic ideas of methods and optimality conditions for MPEC, and therefore our list of the literature is not exhaustive.

In the following sections, after Sect. 1 of variational inequality, we present short review of the basic approaches and optimality conditions for MPEC.

1 Variational Inequality Problem

The variational inequality problem is a general problem formulation that encompasses a wide range of problems, including, among others, optimization problems, complementarity problems, fixed point problems, and network equilibrium problems. In this section, we briefly review several important moments for solving variational inequality problems.

Mathematical programmers' interest in the variational inequality problem stems primarily from the recognition that the equilibrium conditions for network equilibrium problems can be formulated in a natural way as a variational inequality problems, and secondly it includes as special cases virtually all of classical problems of mathematical programming: the first-order optimality conditions, linear and nonlinear complementarity problems, fixed point problems, and minimax problems.

Let an operator $F : R^{n+m} \to R^m$ and a set-valued map $C : R^n \to R^m$ be given. Then variational inequality problem, VI(F,C), is defined as follows:

$$\begin{cases} \text{find} & y \in C(x) \\ \text{such that} & \langle F(x,y), u-y\rangle \geq 0, \quad \text{for all } u \in C(x), \end{cases} \tag{1}$$

and its solution set is denoted by $S(x)$.

Here we will recall some definitions and basic results from [7, 13, 15]. For simplicity, in the remainder of the current section, we fix variable x and give basic definitions and results w.r.t. second variable y.

In other words, we have $C \subset R^m$, $F : R^m \to R^m$ and consider a version of the variational inequality problem without parameter x, which is usually referred to as the variational inequality problem, VI(F,C):

$$\begin{cases} \text{find} & y \in C \\ \text{such that} & \langle F(y), u-y\rangle \geq 0, \quad \text{for all } u \in C. \end{cases} \tag{2}$$

Definition 1. An operator $F : R^m \to R^m$ is said to be

- Monotone on C if for all $u, v \in C$

$$\langle F(u) - F(v), u - v\rangle \geq 0$$

- Strictly monotone on C if for all $u, v \in C, u \neq v$:

$$\langle F(u) - F(v), u - v\rangle > 0$$

- Strongly monotone on C if there exists a constant $\kappa > 0$, such that for all $u, v \in C$

$$\langle F(u) - F(v), u - v\rangle \geq \kappa \parallel u - v \parallel$$

- Antimonotone on C if $-F(\cdot)$ is monotone; i.e.,

$$\text{for all } u, v \in C \ \ \langle F(u) - F(v), u - v\rangle \leq 0;$$

- Nonmonotone on C if it is not monotone on C; (*notice that antimonotone operator is nonmonotone too*)

In the literature $C \subset R^m$ is generally assumed to be convex and compact (or, in some cases , convex and closed), and $F : R^m \to R^m$ is generally assumed to be

- Either continuous, hemicontinuous, or continuously differentiable
- Either
  - Monotone on C
  - Strictly monotone on C
  - Strongly monotone on C

1.1 Existence and Convexity of the Solution Set of VIP

The following results specify conditions ensuring that VI(F,C) has a solution:

Theorem 1 ([10]). *If $C \subset R^m$ is compact and convex, and $F(\cdot)$ is continuous, then the variational inequality VI(F,C) has a solution.*

Theorem 2 ([10]). *Suppose that C is closed convex, and $F(\cdot)$ is continuous and satisfies the following condition:*
There exists an $x^0 \in C$ such that

$$\lim_{\|x\|\to\infty, x\in C} \frac{\langle F(x)-F(x^0), x-x^0\rangle}{\|x-x^0\|} = +\infty.$$

Then, the variational inequality problem VI(F,C) has a solution.

Theorem 3 ([1]). *Suppose that C is closed and convex, and $F(\cdot)$ is monotone, hemicontinuous, and satisfies the following coercivity condition on C:*
There exists an $x^0 \in C$ and a scalar $\gamma > 0$ such that

$$\text{if } x \in C \text{ and } \|x\| > \gamma, \text{ then } \langle F(x), x-x^0\rangle > 0.$$

Then, the variational inequality problem VI(F,C) has a solution.

Theorem 4 ([1]). *If C is closed and convex, and $F(\cdot)$ is strongly monotone and hemicontinuous on C, then, the variational inequality problem VI(F,C) has a solution.*

The following theorem specifies conditions on the operator (or mapping) $F(\cdot)$ which gives an idea of structures (uniqueness and convexity) of the solution set of the variational inequalities:

Theorem 5 ([10]). *Let C be a closed convex set in R^m and $F(\cdot)$ be a continuous mapping. Let S denote the (possibly empty) solution set of the VI(F,C).*

- *If $F(\cdot)$ is monotone on C, then S, if nonempty, is a closed convex set.*
- *If $F(\cdot)$ is strictly monotone on C, then S consists of at most one element.*
- *If $F(\cdot)$ is strongly monotone on C, then S consists of exactly one element.*

1.2 Relationship to Other Problems

Let us discuss briefly the relationship between the variational inequality problems and other optimization-related classical problems:

(i) *Optimization and VIP* : The operator $F(\cdot)$ is a gradient mapping on C if there exists a Gateaux differentiable functional $\varphi(\cdot)$ such that $F(x) = \nabla\varphi(x)$ for every x in C.

If $F(\cdot)$ is a gradient mapping, namely the gradient of the continuously differentiable functional $\varphi(\cdot)$, then y solves VI(F,C) precisely when y minimizes locally functional $\varphi(\cdot)$ over C.

So, whenever $F(\cdot)$ satisfies the properties of gradient mapping, we can solve variational inequality problem using any algorithm that will solve the equivalent optimization problem, if the latter is solvable.

(ii) *Complementarity and VIP* : If $C = R^m_+$ then variational inequality problem (2) is equivalent to the problem of

$$\begin{cases} \text{find} & y \geq 0 \\ \text{such that } F(y) \geq 0 \text{ and } \langle F(y), y\rangle = 0. \end{cases} \tag{3}$$

(iii) *Fixed point and VIP* : Let $F(x) = x - \Omega(x)$ for every $x \in C$. Then VI(F,C) and fixed point problem

$$\text{find } y \in C \quad \text{such that} \quad y = \Omega(y)$$

have precisely the same solutions, if any.

1.3 Traffic Equilibrium

We consider the traffic equilibrium problem.

Let $G = (N,A)$ be a telecommunication network consisting of a set N of nodes and a set of A of directed arcs. Let W be a set of origin-destination (OD) node pairs. For each $w \in W$, let P_w be the set of directed paths joining the OD pair w. We assume a fixed demand, d_w, for sending from the origin node to the destination node of OD pair w. Let h_p be the flow variable on path $p \in P_w$, where $w \in W$. Thus

$$d_w = \Sigma_{p \in P_w} h_p, \text{ for all } w \in W.$$

We group together the path flows h_p into vector $h \in R^m$ (where m is the number of paths in the network). Let C be the set of all feasible path flow vectors:

$$C = \{h \mid h_p \geq 0, \Sigma_{p \in P_w} h_p = d_w \text{ for every } p \in P_w, \text{ and } w \in W\}.$$

Let $F_p(h)$ represent the marginal cost of a unit of flow on path p as a known smooth function of the flow h on the network.

According to the Wardrop equilibrium principle,

for a given OD pair w, if a path $p \in P_w$ is used ($h_p > 0$), then the marginal cost $F_p(h)$ on that path must be minimal among the marginal costs on all paths joining OD pair w.

This equilibrium principle can be stated mathematically as follows:

$$\begin{cases} h^* \text{ is an equilibrium flow} \\ \text{if for each } w \in W \text{ and each } p \in P_w, \text{ it satisfies:} \\ \text{if } h_p^* > 0, \text{ then } F_p(h^*) = \min\{F_q(h^*) \mid q \in P_w\}. \end{cases}$$

Following [23], we now know that these equilibrium conditions can be reformulated as the following variational inequality problem:

$$\text{find } h^* \in C \text{ such that } \langle F(h^*), h - h^* \rangle \geq 0 \text{ for every flow } h \in C.$$

2 Mathematical Programs with Equilibrium Constraints

In the remainder we consider MPEC as the following:

$$\begin{cases} \text{minimize } \varphi(x,y) \\ \text{subject to } x \in \Omega, \\ \qquad\qquad y \in S(x), \end{cases} \tag{MPEC}$$

where $\varphi(x,y)$ is continuously differentiable and

$$S(x) = \{y \in R^m \mid y \in C(x) \text{ and } \langle F(x,y), u - y \rangle \geq 0 \ \ \forall u \text{ s.t. } u \in C(x)\}.$$

We assume that $C(x)$ is defined by

$$C(x) = \{y \in R^m \mid g_i(x,y) \geq 0, i = 1, \ldots, l\}$$

with $g : R^{n+m} \to R^l$ twice continuously differentiable and concave in the second variable. We indicate by $I(x,y)$ the set of active constraints, i.e.

$$I(x,y) = \{i \mid g_i(x,y) = 0\}.$$

We make the following blanket assumptions:

A1: $C(x) \neq \emptyset$ for all $x \in \bar{\Omega}$, where $\bar{\Omega}$ is an open set containing Ω.
A2: $C(x)$ is uniformly compact on $\bar{\Omega}$.
A3: F is strongly monotone with respect to y.

A4: Ω is compact.
A5: At each $x \in \Omega$ and $y \in S(x)$, the partial gradients $\nabla_y g_i(x,y)$, for $i \in I(x,y)$ are linearly independent.

Then by A1, A2, and A3, for every $x \in \Omega$, there exists one and only one solution to the lower-level variational inequality. Furthermore by A5, every solution must satisfy the KKT conditions for the optimization problem:

$$\min\{\langle F(x,y),u\rangle \mid g(x,u) \geq 0\}$$

for a unique $\lambda \in R^l$

$$\begin{cases} F(x,y) - \nabla_y g(x,y)\lambda = 0, \\ \lambda \geq 0, g(x,y) \geq 0, \\ \langle \lambda, g(x,y)\rangle = 0. \end{cases} \tag{KKT}$$

Therefore, under our assumptions, the MPEC can be reformulated as the following nonlinear programming problem:

$$\begin{cases} \text{minimize } \varphi(x,y) \\ \text{subject to } x \in \Omega, \\ \qquad F(x,y) - \nabla_y g(x,y)\lambda = 0, \\ \qquad \lambda \geq 0, g(x,y) \geq 0, \\ \qquad \langle \lambda, g(x,y)\rangle = 0. \end{cases} \tag{MPEC'}$$

This formulation has been used frequently in the literature for designing algorithms and for establishing optimality conditions for MPEC.

3 Methods for Solving the MPEC

This section reviews recent algorithmic research on MPEC. Few successful numerical methods have been proposed to date. We divide the algorithms into four general categories:

- Penalty technique
- Nondifferential optimization
- Smoothing methods
- Heuristic methods

Here we focus our discussion on three papers, one from each categories except the last; for further references interested reader is referred to [4, 13]. For heuristic methods see, e.g., [8, 24].

3.1 Penalty Techniques

The principle of exact penalization [21]: The exact penalization approach toward constrained optimization problems

$$\min\{f(x) \mid x \in C\}$$

goes back to Eremin [5]. It aims at replacing the constrained problem by an equivalent unconstrained problem by augmenting the objective function f through the addition of a term which penalizes infeasibility. From a geometric point of view, infeasibility is most naturally measured in terms of the distance

$$d_C(y) = \min\{\|x-y\| \mid x \in C\}$$

of the point y to the closed set C.

Theorem 6. *Let $x \in S \subset R^n$ and let $C \subset S$ be nonempty and closed. Suppose $f : S \to R$ is Lipschitz of rank K on S and let $\bar{K} > K$. Then x is a global minimizer of f over C if and only if x is a global minimizer of the function $f + \bar{K}d_C$ over S.*

Corollary 1. *Let $x \in S \subset R^n$, $f : S \longrightarrow R$ be Lipschitz of rank K in a ball around x which intersects a closed and nonempty set $C \subset R^n$. If $\bar{K} > K$, then x is a local minimizer of f over C if and only if x is an unconstrained local minimizer of $f + \bar{K}d_C$.*

Although the foregoing results are theoretically very appealing, they are only of limited practical value since the mere evaluation of the penalty function involves the solution of a constrained optimization problem. Thus, nothing is won in passing from the constrained $\min\{f(x) \mid x \in C\}$ to the unconstrained problem $\min\{f(x) + \bar{K}d_C \mid x \in R^n\}$. One is therefore interested in finding upper bounds for the distance function in terms of functions which are easier to evaluate. Such majorants can again be used as penalization terms as pointed out in the following corollary.

Corollary 2. *If the assumptions of Theorem 6 hold and if $\psi : S \to R$ is a function such that*

1. $\psi(y) \geq d_C(y)$ *for every* $y \in S$
2. $\psi(y) = d_C(y)$ *for every* $y \in C$

then x is a global minimizer of f over C if and only if x is a global minimizer of the function $f + \bar{K}\psi$ over S.

The authors of the paper propose the MPEC penalty function of the type:

$$f(x) = g(x) + p(h(x)),$$

where g is the C^1 objective function, p is piecewise affine, and h is a vector-valued function. Such functions are locally Lipschitz and B-differentiable. Suitable tool for the minimization of the penalty functions f is the bundle-trust-region method of Schramm and Zowe [22].

The authors propose a different trust-region method which is applicable to the nonsmooth MPEC penalty functions and is designed to find a B-stationary point.

3.2 Nondifferential Optimization

In the case of bilevel problems, the key assumption is that the map, which assigns to the upper-level variable the solutions of the lower-level problem, is single-valued and locally Lipschitz. Then the problem can be converted into the minimization of a locally Lipschitz objective which depends on the upper-level variable only. This map is nondifferentiable and thus one has to refer to methods from nondifferentiable optimization.

The authors of the paper [18] extended this approach to (MPEC): the assumptions similar to (A1)–(A5) are required. This implies that for each x the variational inequality possesses exactly one solution y, and thus (VI) defines an operator S assigning to x this unique $y = S(x)$. So, in this case one can write (MPEC) as the following:

$$\begin{cases} \text{minimize } \varphi(x,y) \\ \text{subject to } x \in \Omega, \\ \qquad\qquad y = S(x). \end{cases} \tag{4}$$

Another way to write (4) is

$$\begin{cases} \text{minimize } \Theta(x) = \varphi(x, S(x)) \\ \text{subject to } x \in \Omega. \end{cases} \tag{5}$$

As S and thus also Θ are nondifferentiable in general, standard optimization methods cannot be applied, and we have to refer to methods from nondifferentiable optimization. Among them an important place occupies so-called bundle methods [9] which construct and update during the iteration process piecewise affine local models of the objective. These local models are based on the objective values and subgradients at the single iteration points. They are enriched in a finite number of so-called null steps in such a way that a descent direction for the objective can be computed. For nonconvex nondifferentiable problems the convergence of bundle method has been proved in [22] under the basic assumptions that the objectives are locally Lipschitz and directionally differentiable. In this paper the authors showed that the composite objective Θ satisfies these requirements, and moreover, they are able to compute at each iteration a subgradient of Θ needed for building up the mentioned local model.

3.3 Smoothing Methods

It is easy to see that, in general, problem (MPEC) does not satisfy standard constraint qualifications because of the complementarity-type constraint of the paper [6]. The authors reformulate the (MPEC) as the following nonsmooth equivalent problem:

$$\begin{cases} \text{minimize } \varphi(x,y) \\ \text{subject to } x \in \Omega, \\ \qquad F(x,y) - \nabla_y g(x,y)\lambda = 0, \\ \qquad g(x,y) - z = 0, \\ \qquad -2\min\{\lambda, z\} = 0, \end{cases} \tag{6}$$

where $z \in R^l$ and the min operator is applied componentwise to the vectors λ and z. Introducing the function $H_0 : R^{n+m+l+l} \to R^{m+l+l}$, defined as

$$H_0(w) = H_0(x,y,z,\lambda) = \begin{pmatrix} F(x,y) - \nabla_y g(x,y)\lambda \\ g(x,y) - z \\ -2\min\{\lambda, z\} \end{pmatrix},$$

the authors rewrite the above problem more compactly as the following:

$$\begin{cases} \text{minimize } \varphi(x,y) \\ \text{subject to } x \in \Omega, \\ \qquad H_0(x,y,z,\lambda) = 0. \end{cases} \tag{P}$$

With respect to Problem (6) the new variable z has been added, which at feasible points is always equal to $g(x,y)$.

Proposition 1. *(x^*, y^*) is the global (a local) solution of the (MPEC) if and only if there exists a vector (z^*, λ^*) such that $(x^*, y^*, z^*, \lambda^*)$ is the global (a local) solution to problem (P).*

The strategy of so-called smoothing method is to solve a sequence of smooth, regular one-level problems which progressively approximate problem (P).

Smoothing problem (P)

Let μ be a parameter. Define the function $\phi_\mu : R^2 \to R$ by

$$\phi_\mu(a,b) = \sqrt{(a-b)^2 + 4\mu^2} - (a+b).$$

Proposition 2. *For every μ we have*

$$\phi_\mu(a,b) = 0 \text{ if and only if } a \geq 0, b \geq 0, ab = \mu^2.$$

Note also that for $\mu = 0$, $\phi_\mu(a,b) = -2\min\{a,b\}$ while for every $\mu \neq 0, \phi_\mu$ is smooth. Therefore the function ϕ_μ is a smooth perturbation of the min function. So let us introduce

$$H_\mu(w) = H_\mu(x,y,z,\lambda) = \begin{pmatrix} F(x,y) - \nabla_y g(x,y)\lambda \\ g(x,y) - z \\ \Phi_\mu(\lambda,z) \end{pmatrix},$$

where

$$\Phi_\mu(\lambda,z) = (\phi_\mu(\lambda_1,z_1),\ldots,\phi_\mu(\lambda_l,z_l))^\top \in R^l.$$

Then, for every $\mu \neq 0$, one may define an optimization problem

$$\begin{cases} \text{minimize } \varphi(x,y) \\ \text{subject to } x \in \Omega, \\ \qquad\qquad H_\mu(x,y,z,\lambda) = 0. \end{cases} \tag{P_μ}$$

The introduction of the smoothing parameter μ has three consequences:

- Nonsmooth problems are transformed into smooth problems, except when $\mu = 0$.
- Well-posedness can be improved in the sense that feasibility and constraint qualifications, hence stability, are often more likely to be satisfied for all values of μ.
- Solvability of quadratic approximation problems is improved.

Algorithm G

Step 0: Let $\{\mu_k\}$ be a sequence on nonzeros $\lim_{k\to\infty} \mu_k = 0$.
Choose $w^0 = (x^0, y^0, z^0, \lambda^0)$, and set $k = 1$.
Step 1: Find a global solution w^k to problem (P_{μ_k}).
Step 2: Set $k := k+1$, and go to Step 1.

Theorem 7. *The sequence $\{w^k\}$ generated by algorithm G is contained in a compact set, and each of its limit points is a global solution of problem (P).*

4 Optimality Conditions for MPEC

In this section we reduce our attention just to optimality conditions for MPECs; we recognize in the recent works the following approaches:

- In [13, 14] the authors compute under the so-called "basic constraint qualification" a tangent cone approximating the equilibrium constraint. This leads directly to a primal version of optimality conditions. Via dualization, one gets then a finite family of optimality conditions in the dual, KKT form.

- In [27] an error bound is constructed for the equilibrium constraint in a bilevel program using the value function of the lower-level problem. Under the assumption of so-called partial calmness KKT conditions have been obtained. This idea is further developed and extended to MPECs in [28].
- In [12, 16] only the strongly regular case is investigated; cf. Robinson [20]. Then, close to the solution, the equilibrium constraint defines a Lipschitz implicit function assigning the parameters the (unique) solutions of the corresponding VI (complementarity problem). This implicit function is described by means of the generalized Jacobians; cf. Clarke [3], and the generalized differential calculus of F.H. Clarke leads then to optimality conditions, again in the KKT form.
- In [26,29] the generalized differential calculus of B. Mordukhovich is employed. Zhang and Treiman [29] deal with bilevel programs, [26] with a general MPEC. In [26, 29] the equilibrium constraint is augmented to the objective by an exact penalty, whereas in the lower-level problem is replaced by the Mordukhovich stationarity conditions.

The above list is not exhaustive; further references can be found in [13]. We would like to underline the following papers [17, 25] devoted to optimality conditions for MPEC:

- In [17] using the generalized differential calculus for nonsmooth and set-valued mappings due to B. Mordukhovich, the author derives first-order necessary optimality conditions. The imposed constraint qualification is studied in detail and compared with other conditions arising in this context.
- Under mild constraint qualification, in [25] the author derives some necessary and sufficient optimality conditions involving the proximal coderivatives.

Acknowledgements This research was partly supported by the Agence Nationale de la Recherche under a grant HORUS and partly by the DIGITEO under a grant 2009-55D "ARM."

References

1. A. Auslender (1976) *Optimisation: Methodes Numeriques Masson*, Paris.
2. D.P.Bertsekas, (1991) *Linear Network Optimization:* MIT Press, Cambridge.
3. F.H.Clarke (1983) *Optimization and Nonsmooth Analysis* Wiley, NewYork.
4. S. Dempe(2003) *Annotated Bibliography on Bilevel Programming and Mathematical Programs with Equilibrium Constraints*, Optimization, 52, 333–359.
5. I.I. Eremin, (1966) *The penalty method in convex programming*, Soviet Math. Dokl., 8, 459–462.
6. F. Facchinei, H. Jiang and L. Qi (1999) *A smoothing method for mathematical programs with equilibrium constraints*, Mathematical Programming, 85A:107–134.
7. D. Fortin and I. Tsevendorj (2000) *Nonmonotone VIP: a bundle type approach*, Rapport de Recherche N4062, INRIA.
8. T.L.Frisz, R.L.Tobin, H.-J. Cho and N.J.Mehta (1990) *Sensitivity analysis based heuristic algorithms for mathematical programs with variational inequality constraints*, Mathematical Programming 48, 265–284.

9. J.-B. Hiriart-Urruty and C. Lemarechal.(1993) *Convex Analysis and Minimization Algorithms II*, volume 306 of Grundlehren der mathematischen Wissenschaften. Springer, Berlin, Heidelberg.
10. D. Kinderlehrer and Stampacchia (1980) *An introduction to variational inequalities and applications*, Academic Press, New York, NY.
11. M. Kocvara, J.V.Outrata (1995) *On the solution of optimum design problems with variational inequalities* in Du, Qu and Womersley eds. Recent Advances in Nonsmooth Optimization, 172–192.
12. M. Kocvara, J.V.Outrata (1997) *A nonsmooth approach to optimization problems with equilibrium constraints* in Ferris and Pang eds. Complementarity and variational problems, SIAM, Philadelphia, PA, 148–164.
13. Z.Q. Luo, J.S.Pang, and D.Ralph,(1996) *Mathematical Programs with Equilibrium Constraints*, Cambridge University Press, Cambridge, UK.
14. Z.Q. Luo, J.S.Pang, D.Ralph,S.-Q.Wu (1996) *Exact penalization and stationary conditions of methematical programs with equilibrium constraints*, Mathematical Programming, 75, 19–76.
15. J.M.Ortega and W.C. Rheinboldt (1970) *Iterative solution of nonlinear equations in several variables*, Academic Press, New York-London
16. J.V. Outrata (1994) *On optimization problems with variational inequality constraints*, SIAM Journal on optimization, 4, 340–357.
17. J.V. Outrata, (1999) *Optimality Conditions for a class of mathematical programs with equilibrium constraints*, Mathematics of operations research, vol 24, No. 3, p.627–644.
18. J.Outrata, J. Zowe (1995) *A numerical approach to optimization problems with variational inequality constraints*, Mathematical Programming, 68, 105–130.
19. J.Outrata, M. Kocvara, J. Zowe, (1998) *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints* , Kluwer, Dordrecht
20. S.M.Robinson (1980) *Strongly regular generalized equations* Math.oper. Res., 5, 43–62.
21. S. Scholtes, M. Stoer (1999) *Exact Penalization of Mathematical Programs with Equilibrium Constraints* , SIAM J. Control Optim., vol. 37, No 2, 617–652.
22. H. Schramm, J. Zowe,(1992) *A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results*, SIAM Journal on optimization, 2, 121–152.
23. M. Smith, (1979) *The Existence, Uniqueness and Stability of Traffic Equilibria*, Transportation Research B, 17B:4, 291–303.
24. C. Suwansirikul, T.L. Friesz and R.L. Tobin (1987) *Equilibrium decomposed optimization: A heuristic for the continuous equilibrium network design problem*, Transportation Sciences, 21, 254–263.
25. J.J. Ye, Optimality Conditions for optimization problems with complementarity constraints, SIAM J. Optim. vol.9, No. 2, 374–387.
26. J.J. Ye, X.Y. Ye (1997) *Necessary Optimality Conditions for Optimization Problems with Variational Inequality Constraints*, Math. Oper. Res. 22 977–997.
27. J.J. Ye, D.L. Zhu (1995) *Optimality conditions for bilevel programming problems*, Optimization 33, 9–27.
28. J.J. Ye, D.L. Zhu , Q.J. Zhu (1997) *Exact penalization and necessary optimality conditions for generalized bilevel programming problems*, SIAM Journal on optimization, 7, 481–507.
29. R. Zhang (1995) *Problems of hierarchical optimization in finite dimensions*, SIAM J. Optimization, 7, 521–536.

Linear Programming with Interval Data: A Two-Level Programming Approach

Chiang Kao and Shiang-Tai Liu

Abstract Linear programming has been widely applied to solving real world problems. The conventional linear programming model requires the parameters to be known constants. In the real world, however, the parameters are seldom known exactly and have to be estimated. This chapter discusses the general interval linear programming problems where all the parameters, including the cost coefficients, requirement coefficients, and technological coefficients, are represented by interval data. Since the parameters are interval-valued, the objective value is interval-valued as well. A pair of two-level mathematical programs is formulated to calculate the lower bound and upper bound of the objective values of the interval linear program. The two-level mathematical programs are then transformed into one-level nonlinear programs. Solving the pair of nonlinear programs produces the interval of the objective values of the problem. An example illustrates the whole idea and sheds some light on interval linear programming.

Key words Linear programming • Interval parameter • Two-level programming

1 Introduction

Linear programming is a mathematical modeling technique designed to optimize the usage of limited resources. It has been widely used to solve problems in military, industries, agriculture, economics, and even behavioral and social sciences.

C. Kao (✉)
Department of Industrial and Information Management,
National Cheng Kung University, Tainan, Taiwan
e-mail: ckao@mail.ncku.edu.tw

S.-T. Liu
Graduate School of Business and Management, Vanung University, Tao-Yuan, Taiwan
e-mail: stliu@vnu.edu.tw

A. Chinchuluun et al. (eds.), *Optimization, Simulation, and Control*, Springer Optimization and Its Applications 76, DOI 10.1007/978-1-4614-5131-0_5,

Several surveys (see, e.g., Hartley [6], Lane et al. [9]) indicate that linear programming is the most frequently used technique in solving real world problems among all operations research techniques. Numerous textbooks have been written about linear programming. Most textbooks of operations research spend the largest number of pages discussing this topic. Linear programming has become the most important technique and the fundamental for studying other optimization techniques in operations research.

Any linear programming problem can be expressed by the following model:

$$\begin{aligned} \text{Min } Z &= \boldsymbol{cx} \\ \text{s.t. } \boldsymbol{Ax} &= \boldsymbol{b} \\ \boldsymbol{x} &\geq \boldsymbol{0}, \end{aligned} \tag{1}$$

where $\boldsymbol{x} = (x_j, j = 1, \ldots, n)$ is the vector of decision variables to be determined. The other variables are the parameters given by the problem: $\boldsymbol{c} = (c_j, j = 1, \ldots, n)$ is vector of cost coefficients, $\boldsymbol{b} = (b_i, i = 1, \ldots, m)$ is vector of requirement coefficients, and $\boldsymbol{A} = ||a_{ij}||$ is the matrix of technological coefficients. The problem is to determine the values of the decision variables under the constraints which minimize the objective function. The optimal values of the decision variables $x_j, j = 1, \ldots, n$ are functions of the parameters a_{ij}, b_i, and $c_j, i = 1, \ldots, m, j = 1, \ldots, n$. When the value of one or more of the parameters is changed, the optimal values of the decision variables and the objective function will in general change accordingly.

Linear programming makes several assumptions regarding the parameters. The major one is that the value assigned to each parameter is a known constant. However, in real world applications, this assumption is seldom satisfied because linear programming models are usually formulated to find some future course of action. The parameter values used would be based on a prediction of future conditions which inevitably introduces some degree of uncertainty. There are also situations where the data cannot be collected without error. In the literature, the approaches for solving this problem are typified by post-optimality analysis [5]. As implied by its name, post-optimality analysis concerns how the optimal solution changes when the value of one or more parameters is changed. It is an ex post facto analysis after the optimal solution for a set of known parameters is solved. The technique which deals with changing one parameter at a time is called sensitivity analysis and the one dealing with changing several parameters simultaneously is called parametric programming [10, 12, 14]. Another approach in this category is the tolerance approach which focuses on simultaneous and independent variations of the requirement coefficients and cost coefficients without affecting the optimality of the given basis [4, 13, 15–18]. The primary objective is to find the range of the parameters within which the current solution is still optimal.

In contrast to post-optimality analysis, which is conducted after an optimal solution is obtained, this chapter deals with the problem of finding the optimal solution for the linear programming problem whose imprecise parameters are expressed by intervals in an a priori manner. One approach for dealing with uncertainty in

parameters is via stochastic programming, in which the parameters are treated as random variables. The standard procedure is to optimize the expected value of the objective function. Dantzig [3] discusses the case where random variables appear only in the requirements, and Charnes et al. [1] discuss the case of random costs. The problem becomes very complicated when all a_{ij}, b_i, and c_j are random variables. Another way to represent imprecise parameters in real world applications is by intervals [2, 7]. The associated linear program is an interval linear program. When the parameters have interval values, the objective function will also have an interval value; that is, it lies in a range. Serafini [11] proposed a two-phase approach for solving the linear program where the requirement coefficients are represented by intervals. The method only gives a point value for the objective function. In this chapter, we construct a pair of two-level mathematical programming models, based on which the lower bound and upper bound of the objective values are obtained. In other words, an interval value for the objective function of the interval linear programming problem is derived. This result should provide the decision maker with more information for making better decisions.

In the next section, we shall discuss the nature of interval linear programming, followed with a two-level mathematical programming formulation for finding the bounds of the interval objective values. Section 3 describes how to transform the two-level mathematical program into the conventional one-level program. We then use an example to illustrate how to apply the concept of this chapter to solve the interval linear programming problem. Finally, we draw a conclusion and suggest some directions for future study.

2 Problem Formulation

Before we get into the details of this chapter, a simple example helps clarify the nature of linear programming problems with interval parameters. Consider the following interval linear program:

$$\text{Min} \quad Z = 4x_1 + 3x_2 \tag{2}$$

$$\text{s.t.} \quad x_1 + [1,2]x_2 = 4 \tag{3}$$

$$[2,3]x_1 + x + 2 \geq 6 \tag{4}$$

$$x_1, x_2 \geq 0,$$

where the parameters a_{12} and a_{21} are imprecise and are represented by intervals [1–3], respectively. As a_{12} varies from the lower bound 1 to upper bound 2, the feasible region defined by Constraint (3) and the nonnegativity conditions is a line segment moving counterclockwise from $\overline{AF}$ to $\overline{AG}$ as depicted in Fig. 1.

For the second Constraint (4), as parameter a_{21} changes from its lower bound 2 to upper bound 3, the boundary of the feasible region represented by this constraint

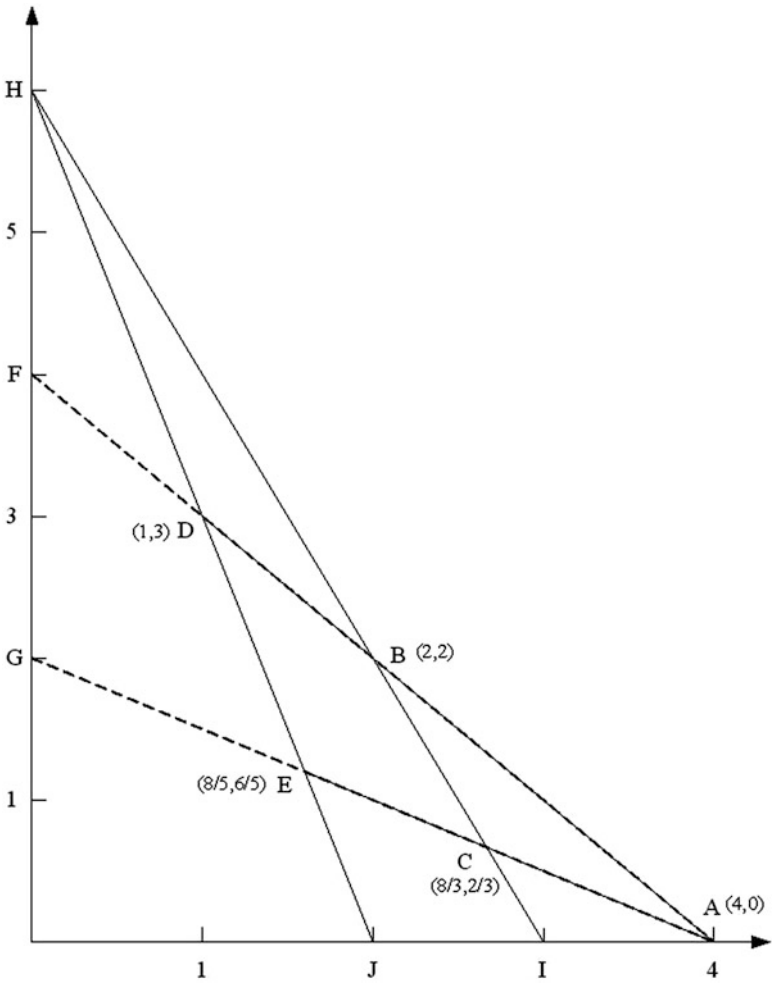


Fig. 1 Graphical solution of the example

swings clockwise from $\overline{HI}$ to $\overline{HJ}$. Clearly, the feasible region defined by this constraint becomes larger when a_{21} increases in its value. In other words, the smallest feasible region occurs at $a_{21} = 2$ and the largest at $a_{21} = 3$. For the former, when Constraint (3) is also considered, the feasible region is the line segment moving continuously from $\overline{AB}$ to $\overline{AC}$. If it is $\overline{AB}$, then, graphically, the minimal value of the objective function $Z = 4x_1 + 3x_2$ occurs at $B = (2,2)$, with an objective value of 14. As the feasible region moves to $\overline{AC}$, the minimal value decreases to $\frac{38}{3}$ which occurs at $C = (\frac{8}{3}, \frac{2}{3})$. Similarly, for the latter case of largest feasible region, the feasible region is the line segment $\overline{AD}$ moving continuously to $\overline{AE}$. The minimal value for $\overline{AD}$ is 13, occurring at $D = (1,3)$, and for $\overline{AE}$ it is 10, occurring

at $E = (\frac{8}{5}, \frac{6}{5})$. Combining these results together, we conclude that the lower bound of the optimal objective values is 10 and the upper bound is 14. The optimal value lies in the range of [10, 14].

This example shows that if the constraint coefficients are interval-valued, then the objective value will lie in a range. The graphical solution method helps derive the lower bound and upper bound of the objective values of the problem. The lower bound is obtained in the largest feasible region of the triangle *ADE* while the upper bound is obtained in the smallest feasible region of the triangle *ABC*. This example is so simple that a visual inspection suffices to find the solution. For general problems, we need to rely on some systematic solution method.

For the conventional linear program of Model (1), if one or more parameters have interval values, then we have an interval linear program. Without loss of generality, we assume all parameters are interval-valued since a constant can be represented by a degenerated interval where the lower bound of the interval coincides with its upper bound. As opposed to the conventional linear program where an unconstrained variable can be expressed by the difference of two nonnegative variables, an unconstrained variable in an interval linear program cannot be transformed in this way. The reason will be clear later in the derivation of the solution method. Therefore, the variables are separated into two groups, one nonnegative and the other unconstrained in sign. To be consistent with the dual problem formulation, the constraints are also separated into two groups, one of inequality type and the other of equality type, so that the corresponding dual variables will be nonnegative and unconstrained in sign, respectively. In this chapter, the interval linear program is formulated as:

$$\begin{aligned}
\text{Min} \quad & Z = \sum_{j=1}^{n} \hat{c}_j x_j \\
\text{s.t.} \quad & \sum_{j=1}^{n} \hat{a}_{ij} x_j \geq \hat{b}_i, \quad i = 1, \ldots, p \\
& \sum_{j=1}^{n} \hat{a}_{ij} x_j = \hat{b}_i, \quad i = p+1, \ldots, m \\
& x_j \geq 0, \quad j = 1, \ldots, q; \\
& x_j \text{ unconstrained in sign}, \quad j = q+1, \ldots, n,
\end{aligned} \tag{5}$$

where $\hat{c}_j \in [C_j^L, C_j^U]$, $\hat{b}_i \in [B_i^L, B_i^U]$, and $\hat{a}_{ij} \in [A_{ij}^L, A_{ij}^U]$ are the interval counterparts of c_j, b_i, and a_{ij}, respectively. The inequality constraint of the "$\leq$" form can be transformed to the form of "$\geq$" by multiplying the terms on both sides by "-1." If the objective function is "Max," then it can be changed to "$-$Min $-$ Z" to conform to Model (5). Hence, (5) is a generic interval linear programming model.

Clearly, different values of $\hat{c}_j, \hat{b}_i$, and $\hat{a}_{ij}$ produce different objective values. To find the interval of the objective values, it suffices to find the lower bound and upper

bound of the objective values of Model (5). Denote $\boldsymbol{S} = \{\hat{\boldsymbol{c}}, \hat{\boldsymbol{b}}, \hat{\boldsymbol{a}}) | C_j^L \le \hat{c}_j \le C_j^U, B_i^L \le \hat{b}_i \le B_i^U, A_{ij}^L \le \hat{a}_{ij} \le A_{ij}^U, i = 1, \ldots, m, j = 1, 2, \ldots, n\}$. The values of $\hat{c}_j, \hat{b}_i$, and $\hat{a}_{ij}$ that attain the smallest value for Z can be determined from the following two-level mathematical programming model:

$$
\begin{aligned}
Z^L = \text{Min}_{(\hat{c},\hat{b},\hat{a}) \in S} \, \text{Min}_x \;\; & Z = \sum_{j=1}^{n} \hat{c}_j x_j \\
\text{s.t.} \;\; & \sum_{j=1}^{n} \hat{a}_{ij} x_j \ge \hat{b}_i, \;\; i = 1, \ldots, p \\
& \sum_{j=1}^{n} \hat{a}_{ij} x_j = \hat{b}_i, \;\; i = p+1, \ldots, m \\
& x_j \ge 0, \;\; j = 1, \ldots, q; \\
& x_j \text{ unconstrained in sign}, \;\; j = q+1, \ldots, n, \qquad (6)
\end{aligned}
$$

where the inner program calculates the objective value for each $\hat{c}_j, \hat{b}_i$, and $\hat{a}_{ij}$ specified by the outer program, while the outer program determines the values of $\hat{c}_j, \hat{b}_i$, and $\hat{a}_{ij}$ that produces the smallest objective value. The objective value is the lower bound of the objective values for Model (5).

By the same token, to find the values of $\hat{c}_j, \hat{b}_i$, and $\hat{a}_{ij}$ that produce the largest objective value for Z, a two-level mathematical program is formulated by replacing the outer program of Model (6) from "Min" to "Max":

$$
\begin{aligned}
Z^U = \text{Max}_{(\hat{c},\hat{b},\hat{a}) \in S} \, \text{Min}_x \;\; & Z = \sum_{j=1}^{n} \hat{c}_j x_j \\
\text{s.t.} \;\; & \sum_{j=1}^{n} \hat{a}_{ij} x_j \ge \hat{b}_i, \;\; i = 1, \ldots, p \\
& \sum_{j=1}^{n} \hat{a}_{ij} x_j = \hat{b}_i, \;\; i = p+1, \ldots, m \\
& x_j \ge 0, \;\; j = 1, \ldots, q; \\
& x_j \text{ unconstrained in sign} \;\; j = q+1, \ldots, n. \qquad (7)
\end{aligned}
$$

The objective value Z^U is the upper bound of the objective values for Model (5).

When the interval data $\hat{c}_j, \hat{b}_i$, and $\hat{a}_{ij}$ degenerate to point data c_j, b_i, and a_{ij}, respectively, the outer program of Models (6) and (7) vanishes, and Models (6) and (7) boil down to the same conventional linear program. This shows that the two-level mathematical program formulation of the interval linear program developed here is a generalization of the conventional constant-parameter linear program. The pair of two-level mathematical programs in (6) and (7) clearly express the bounds of

the objective values. However, they are not solvable in the current form. In the next section, we discuss how to transform the two-level program into the conventional one-level program. With the pair of one-level programs, the interval of the objective values of the interval linear program can be obtained.

3 One-Level Transformation

3.1 Lower Bound

The previous section showed that to find the lower bound of the objective values of an interval linear programming problem of Model (5), it suffices to solve the two-level mathematical program of Model (6). Since both the inner program and outer program of (6) have the same minimization operation, they can be combined into a conventional one-level program with the constraints of the two programs considered at the same time.

$$
\begin{aligned}
Z^L = \text{Min} \quad & Z = \sum_{j=1}^{n} \hat{c}_j x_j \\
\text{s.t.} \quad & \sum_{j=1}^{n} \hat{a}_{ij} x_j \geq \hat{b}_i, \quad i = 1, \ldots, p \\
& \sum_{j=1}^{n} \hat{a}_{ij} x_j = \hat{b}_i, \quad i = p+1, \ldots, m \\
& C_j^L \leq \hat{c}_j \leq C_j^U, \quad j = 1, \ldots, n \\
& B_i^L \leq \hat{b}_i \leq B_i^U, \quad i = 1, \ldots, m \\
& A_{ij}^L \leq \hat{a}_{ij} \leq A_{ij}^U, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n \\
& x_j \geq 0, \quad j = 1, \ldots, q; \\
& x_j \text{ unconstrained in sign}, \quad j = q+1, \ldots, n
\end{aligned} \tag{8}
$$

This model is a nonlinear program. By separating the decision variables into nonnegative ones and unconstrained-in-sign ones, it can be rewritten as:

$$
Z^L = \text{Min} \quad Z = \sum_{j=1}^{q} \hat{c}_j x_j + \sum_{j=q+1}^{n} \hat{c}_j x_j \tag{9}
$$

$$
\text{s.t.} \quad \sum_{j=1}^{q} \hat{a}_{ij} x_j + \sum_{j=q+1}^{n} \hat{a}_{ij} x_j \geq \hat{b}_i, \quad i = 1, \ldots, p \tag{10}
$$

$$\sum_{j=1}^{q} \hat{a}_{ij}x_j + \sum_{j=q+1}^{n} \hat{a}_{ij}x_j = \hat{b}_i, \quad i = p+1,\ldots,m \tag{11}$$

$$C_j^L \le \hat{c}_j \le C_j^U, \quad j = 1,\ldots,n \tag{12}$$

$$B_i^L \le \hat{b}_i \le B_i^U, \quad i = 1,\ldots,m \tag{13}$$

$$A_{ij}^L \le \hat{a}_{ij} \le A_{ij}^U, \quad i = 1,\ldots,m, \quad j = 1,\ldots,n \tag{14}$$

$$x_j \ge 0, \quad j = 1,\ldots,q;$$

$$x_j \text{ unconstrained in sign}, \quad j = q+1,\ldots,n \tag{15}$$

For nonnegative x_j we have $C_j^L x_j \le \hat{c}_j x_j \le C_j^U x_j$ as is manifested from (12). In searching for the minimal value of the objective function, the interval parameter $\hat{c}_j, j = 1,\ldots,q$, must reach its lower bound. Consequently, we have

$$\text{Min } Z = \sum_{j=1}^{q} C_j^L x_j + \sum_{j=q+1}^{n} \hat{c}_j x_j.$$

The largest feasible region defined by the inequality constraint $\sum_{j=1}^{n} \hat{a}_{ij}x_j \ge \hat{b}_i$ in Models (9)–(15) appears when the interval parameter $\hat{b}_i$ is equal to its lower bound B_i^L. We can reduce the number of nonlinear terms by using a variable transformation technique, that is, multiplying Constraint (14) by nonnegative x_j and substituting $\hat{a}_{ij}x_j$ by r_{ij}. Models (9)–(15) then become

$$\begin{aligned}
Z^L = \text{Min} \quad & Z = \sum_{j=1}^{q} C_j^L x_j + \sum_{j=q+1}^{n} \hat{c}_j x_j \\
\text{s.t.} \quad & \sum_{j=1}^{q} r_{ij} + \sum_{j=q+1}^{n} \hat{a}_{ij}x_j \ge B_i^L, \quad i = 1,\ldots,p \\
& \sum_{j=1}^{q} r_{ij} + \sum_{j=q+1}^{n} \hat{a}_{ij}x_j = \hat{b}_i, \quad i = p+1,\ldots,m \\
& C_j^L \le \hat{c}_j \le C_j^U, \quad j = q+1,\ldots,n \\
& B_i^L \le \hat{b}_i \le B_i^U, \quad i = p+1,\ldots,m \\
& A_{ij}^L x_j \le r_{ij} \le A_{ij}^U x_j, \quad i = 1,\ldots,m, \quad j = 1,\ldots,n \\
& A_{ij}^L \le \hat{a}_{ij} \le A_{ij}^U, \quad i = 1,\ldots,m, \quad j = q+1,\ldots,n \\
& x_j \ge 0, \quad j = 1,\ldots,q; \\
& x_j \text{ unconstrained in sign}, \quad j = q+1,\ldots,n.
\end{aligned} \tag{16}$$

The lower bound of the objective value, Z^L, is obtained by solving this mathematical program.

3.2 Upper Bound

Conceptually, the upper bound of the objective value of the interval linear program of Model (5) can be calculated from the two-level program of Model (7). However, solving Model (7) is not as straightforward as solving Model (6) because the outer program and inner program have different directions for optimization, viz., one for maximization and the other for minimization. They cannot be combined into a one-level program directly. Based on the duality theorem, the dual of a linear program has the same optimal objective value as its primal when an optimal solution exists. Hence, we can replace the inner program of Model (7) by its dual to form a maximization problem:

$$
\begin{aligned}
Z^U = \mathrm{Max}_{(\hat{c},\hat{b},\hat{a})\in S}\, \mathrm{Max}_y \quad & Z = \sum_{i=1}^{m} \hat{b}_i y_i \\
\text{s.t.} \quad & \sum_{i=1}^{m} \hat{a}_{ij} y_i \le \hat{c}_j, \quad j = 1,\ldots,q \\
& \sum_{i=1}^{m} \hat{a}_{ij} y_i = \hat{c}_j, \quad j = q+1,\ldots,n \\
& y_i \ge 0, \quad i = 1,\ldots,p; \\
& y_i \text{ unconstrained in sign}, \quad i = p+1,\ldots,m
\end{aligned} \tag{17}
$$

Now that both the inner program and outer program have the same maximization operation, they can be merged into a one-level program with the constraints at the two levels considered at the same time:

$$
\begin{aligned}
Z^U = \mathrm{Max} \quad & Z = \sum_{i=1}^{m} \hat{b}_i y_i \\
\text{s.t.} \quad & \sum_{i=1}^{m} \hat{a}_{ij} y_i \le \hat{c}_j, \quad j = 1,\ldots,q \\
& \sum_{i=1}^{m} \hat{a}_{ij} y_i = \hat{c}_j, \quad j = q+1,\ldots,n \\
& C_j^L \le \hat{c}_j \le C_j^U, \quad j = 1,\ldots,n \\
& B_i^L \le \hat{b}_i \le B_i^U, \quad i = 1,\ldots,m \\
& A_{ij}^L \le \hat{a}_{ij} \le A_{ij}^U, \quad i = 1,\ldots,m, \quad j = 1,\ldots,n \\
& y_i \ge 0, \quad i = 1,\ldots,p; \\
& y_i \text{ unconstrained in sign}, \quad i = p+1,\ldots,m
\end{aligned} \tag{18}
$$

Similar to the case of lower bound, we separate the decision variables y_i into two parts, those of nonnegative ones and unconstrained-in-sign ones:

$$Z^U = \text{Max} \quad Z = \sum_{i=1}^{p} \hat{b}_i y_i + \sum_{i=p+1}^{m} \hat{b}_i y_i \tag{19}$$

$$\text{s.t.} \quad \sum_{i=1}^{p} \hat{a}_{ij} y_i + \sum_{i=p+1}^{m} \hat{a}_{ij} y_i \le \hat{c}_j, \quad j = 1, \ldots, q \tag{20}$$

$$\sum_{i=1}^{p} \hat{a}_{ij} y_i + \sum_{i=p+1}^{m} \hat{a}_{ij} y_i = \hat{c}_j, \quad j = q+1, \ldots, n \tag{21}$$

$$C_j^L \le \hat{c}_j \le C_j^U, \quad j = 1, \ldots, n \tag{22}$$

$$B_i^L \le \hat{b}_i \le B_i^U, \quad i = 1, \ldots, m \tag{23}$$

$$A_{ij}^L \le \hat{a}_{ij} \le A_{ij}^U, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n \tag{24}$$

$$y_i \ge 0, \quad i = 1, \ldots, p;$$

$$y_i \text{ unconstrained in sign}, \quad i = p+1, \ldots, m \tag{25}$$

Regarding the objective function, the interval parameters associated with positive variables must be set to the upper bound to attain the maximal value. In other words, the objective function of (19)–(25) can be replaced by

$$\text{Max } Z = \sum_{i=1}^{p} B_i^U y_i + \sum_{i=p+1}^{m} \hat{b}_i y_i.$$

To find the upper bound Z^U of Models (19)–(25), the interval parameters $\hat{c}_j$ must be set to the values which will generate the largest feasible region. For the inequality constraint $\sum_{i=1}^{p} \hat{a}_{ij} y_i + \sum_{i=p+1}^{m} \hat{a}_{ij} y_i \le \hat{c}_j$, different values of $\hat{c}_j$ define a series of parallel hyperplanes. Obviously, the largest feasible region appears when $\hat{c}_j$ is set to its upper bound C_j^U. Thus, we have $\sum_{i=1}^{p} \hat{a}_{ij} y_i + \sum_{i=p+1}^{m} \hat{a}_{ij} y_i \le C_j^U, j = 1, \ldots, q$. The variable transformation technique, which is utilized in (9)–(15), can also be applied to the nonlinear term $\hat{a}_{ij} y_i$ with positive y_i. One can multiply Constraint (24) by y_i for $i = 1, \ldots, p$ and substitute $\hat{a}_{ij} y_i$ by s_{ij} to reduce the number of nonlinear terms.

Via the dual formulation, bound value assignment, and variable transformation, the two-level mathematical program of Model (7) is transformed into the following nonlinear program:

$$Z^U = \text{Max} \quad Z = \sum_{i=1}^{p} B_i^U y_i + \sum_{i=p+1}^{m} \hat{b}_i y_i$$

$$\text{s.t.} \quad \sum_{i=1}^{p} s_{ij} + \sum_{i=p+1}^{m} \hat{a}_{ij} y_i \le C_j^U, \quad j = 1, \ldots, q$$

$$\sum_{i=1}^{p} s_{ij} + \sum_{i=p+1}^{m} \hat{a}_{ij} y_i = \hat{c}_j, \quad j = q+1, \ldots, n$$

$$C_j^L \le \hat{c}_j \le C_j^U, \quad j = q+1, \ldots, n$$

$$B_i^L \le \hat{b}_i \le B_i^U, \quad i = p+1, \ldots, m$$

$$A_{ij}^L y_i \le s_{ij} \le A_{ij}^U y_i, \quad i = 1, \ldots, p, \quad j = 1, \ldots, n$$

$$A_{ij}^L \le \hat{a}_{ij} \le A_{ij}^U, \quad i = p+1, \ldots, m, \quad j = 1, \ldots, n$$

$$y_i \ge 0, \quad i = 1, \ldots, p$$

$$y_i \text{ unconstrained in sign}, \quad i = p+1, \ldots, m \tag{26}$$

The optimal solution Z^U is the upper bound of the objective values of the interval linear program. Together with Z^L solved from Sect. 3.1, $[Z^L, Z^U]$ constitutes the interval on which the objective values of the interval linear program lie.

3.3 Special Case

If a linear program has only inequality constraints and nonnegative decision variables, then Model (5) is of the following form:

$$\begin{aligned} \text{Min} \quad & Z = \sum_{j=1}^{n} \hat{c}_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} \hat{a}_{ij} x_j \ge \hat{b}_i, \quad i = 1, \ldots, m \\ & x_j \ge 0, j = 1, \ldots, n \end{aligned} \tag{27}$$

Models (16) and (26) for calculating the lower bound and upper bound, respectively, of the objective value are simplified to the following forms:

$$\begin{aligned} Z^L = \text{Min} \quad & Z = \sum_{j=1}^{n} C_j^L x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} r_{ij} \ge B_i^L, \quad i = 1, \ldots, m \\ & A_{ij}^L x_j \le r_{ij} \le A_{ij}^U x_j, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n \\ & x_j \ge 0, j = 1, \ldots, n \end{aligned} \tag{28}$$

$$
\begin{aligned}
Z^U = \text{Max} \quad & Z = \sum_{i=1}^{m} B_i^U y_i \\
\text{s.t.} \quad & \sum_{i=1}^{m} s_{ij} \le C_j^U, \quad j = 1, \ldots, n \\
& A_{ij}^L y_i \le s_{ij} \le A_{ij}^U y_i, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n \\
& y_i \ge 0, \quad i = 1, \ldots, m
\end{aligned} \tag{29}
$$

Since (28) and (29) are linear programs, one can calculate the lower and upper bounds of the objective values easily.

4 An Example

Consider the following interval linear programming problem.

$$
\begin{aligned}
\text{Min} \quad & Z = (7,10)x_1 + (7,9)x_3 - 2x_4 + (-2,-1)x_5 + (-3,-1)x_6 - 10x_7 \\
\text{s.t.} \quad & x_1 + 2x_2 - 2x_3 + (1,4)x_4 + (3,5)x_6 + (1,2)x_7 = (-6,-4) \\
& -2x_1 + (1,3)x_2 - x_3 + (2,3)x_4 + (1,2)x_5 + (1,2)x_6 + 2x_7 = (-1,2) \\
& (1,3)x_1 + 2x_3 + 2x_4 + (2,4)x_5 + 2x_6 - 2x_7 = (6,10) \\
& x_1, x_2, x_4, x_5, x_6 \ge 0; x_3, x_7 \text{ unconstrained in sign}
\end{aligned}
$$

Based on Model (16), the lower bound of the objective value Z^L can be formulated as:

$$
\begin{aligned}
Z^L = \text{Min} \quad & 7x_1 + \hat{c}_3 x_3 - 2x_4 - 2x_5 - 3x_6 - 10x_7 \\
\text{s.t.} \quad & x_1 + 2x_2 - 2x_3 + p_{14} + p_{16} + \hat{a}_{17} x_7 = \hat{b}_1 \\
& -2x_1 + p_{22} - x_3 + p_{24} + p_{25} + p_{26} + 2x_7 = \hat{b}_2 \\
& p_{31} + 2x_3 + 2x_4 + p_{35} + 2x_6 - 2x_7 = \hat{b}_3 \\
& 7 \le \hat{c}_3 \le 9 \\
& -6 \le \hat{b}_1 \le -4, \quad -1 \le \hat{b}_2 \le 2, \quad 6 \le \hat{b}_3 \le 10 \\
& x_4 \le p_{14} \le 4x_4, \quad 3x_6 \le p_{16} \le 5x_6, \quad x_2 \le p_{22} \le 3x_2, \quad 2x_4 \le p_{24} \le 3x_4 \\
& x_5 \le p_{25} \le 2x_5, \quad x_6 \le p_{26} \le 2x_6, \quad x_1 \le p_{31} \le 3x_1, \quad 2x_5 \le p_{35} \le 4x_5 \\
& 1 \le \hat{a}_{17} \le 2 \\
& x_1, x_2, x_4, x_5, x_6 \ge 0; x_3, x_7 \text{ unconstrained in sign}
\end{aligned}
$$

This model is a nonlinear program. By using the nonlinear programming solver LINGO (LINDO Systems 2005), we derive $Z^L = -12,\ x_1^* = 26,\ x_3^* = 38,\ x_7^* = 46,\ x_2^* = x_4^* = x_5^* = x_6^* = 0,\ \hat{c}_3 = 7,\ \hat{b}_1 = -4,\ \hat{b}_2 = 2,\ \hat{b}_3 = 10$, and $\hat{a}_{17} = 1$.

The upper bound of the objective value Z^U, according to Model (26), can be formulated as

$$
\begin{aligned}
Z^U = \text{Max}\ & \hat{b}_1 y_1 + \hat{b}_2 y_2 + \hat{b}_3 y_3 \\
\text{s.t.}\ & y_1 - 2y_2 + \hat{a}_{31} y_3 \le 10 \\
& 2y_1 + \hat{a}_{22} y_2 \le 0 \\
& -2y_1 - y_2 + 2y_3 = \hat{c}_3 \\
& \hat{a}_{14} y_1 + \hat{a}_{24} y_2 + 2y_3 \le -2 \\
& \hat{a}_{25} y_2 + \hat{a}_{35} y_3 \le -1 \\
& \hat{a}_{16} y_1 + \hat{a}_{26} y_2 + 2y_3 \le -1 \\
& \hat{a}_{17} y_1 + 2y_2 - 2y_3 = -10 \\
& 7 \le \hat{c}_3 \le 9 \\
& -6 \le \hat{b}_1 \le -4,\ -1 \le \hat{b}_2 \le 2,\ 6 \le \hat{b}_3 \le 10 \\
& 1 \le \hat{a}_{31} \le 3,\ 1 \le \hat{a}_{22} \le 3,\ 1 \le \hat{a}_{14} \le 4,\ 2 \le \hat{a}_{24} \le 3,\ 1 \le \hat{a}_{25} \le 2 \\
& 2 \le \hat{a}_{35} \le 4,\ 3 \le \hat{a}_{16} \le 5,\ 1 \le \hat{a}_{26} \le 2,\ 1 \le \hat{a}_{17} \le 2 \\
& y_1, y_2, y_3 \text{ unconstrained in sign}
\end{aligned}
$$

By employing LINGO, we obtain $Z^U = 29.8$, which occurs at $y_1^* = -1.4,\ y_2^* = -2.4,\ y_3^* = 1.9,\ \hat{c}_3 = 9,\ \hat{b}_1 = -6,\ \hat{b}_2 = -1$, and $\hat{b}_3 = 10$. The corresponding primal solution is $x_3^* = 2.6,\ x_5^* = 1.6,\ x_7^* = -0.8$, and $x_1^* = x_2^* = x_4^* = x_6^* = 0$.

Combining these two results, we conclude that the objective values of this interval linear program lie in the range of $[-12, 29.8]$.

5 Conclusion

Linear programming has been considered as the most powerful technique for improving the efficiency and increasing the productivity of companies and public organizations. To further expand its applicability, more general models are continually being developed. This chapter generalizes the conventional linear programming of constant parameters to interval parameters. As opposed to the post-optimality analysis which conducts an ex post facto analysis after the optimal solution for a set of constant parameters is obtained, the interval linear programming discusses the range of optimal objective values produced from the interval parameters, including cost, requirement, and technology, in an a priori manner.

The idea is to find the lower bound and upper bound of the range by employing the two-level mathematical programming technique. Following the duality theorem, the two-level mathematical programs are transformed into a pair of one-level mathematical programs so that the numerical solution method can be applied. When all interval parameters degenerate to constant parameters, the two-level mathematical programs boil down to the conventional linear program. An example illustrates that the proposed idea is indeed able to find the range of the objective values of the interval linear programming problem.

For general interval linear programming problems, it is very probable that for some range of interval parameters the problem is infeasible. Our method ignores those infeasible values and finds the lower bound and upper bound of the feasible solutions. It does not identify the range of values which cause infeasibility.

While this chapter develops a pair of mathematical programs which are able to find the lower bound and upper bound of the objective values, the mathematical programs are nonlinear which may be difficult to solve for large-scale problems. In the future, a solution method which only involves linear program formulation is desired to assure solvability.

Finally, interval linear programming is not just a topic for theoretical discussion. It does have real world applications. Kao and Liu [7] used forecasted financial data, represented in intervals, to predict the performance of Taiwan commercial banks. Since the problem has a special structure, it can be solved easily by relying on the linear programming technique. In a later study, Kao and Liu [8] found that the interval data approach produces an interval objective value which is too wide to provide useful information. The values close to the bounds, both the lower bound and upper bound, have very small probability of occurrence. If the distributions of the interval data are known, then the distribution of the objective values, which is more informative for making subsequent decisions, can be obtained. Therefore, another direction for future study is to derive the distribution of the objective values based on the distributions of the parameters.

Acknowledgements This research is supported by the National Science Council of the Republic of China (Taiwan) under Contract NSC 89-2418-H-006-001.

References

1. Charnes, A., W.W. Cooper, G.H. Symonds.: Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. Management Science. 4, 235–263 (1958).
2. Cooper, W.W., K.S. Park, G. Yu.: An illustrative application of IDEA (Imprecise Data Envelopment Analysis) to a Korean mobile telecommunication company. Operations Research. 49, 807–820 (2001).
3. Dantzig, G.B.: Linear programming under uncertainty. Management Science. 1, 197–206 (1955).
4. Filippi, C.: A fresh view on the tolerance approach to sensitivity analysis in linear programming. European Journal of Operational Research. 167, 1–19 (2005).

5. Gal, T.: Postoptimal Analyses, Parametric Programming, and Related Topics. 2nd ed. Walter de Gruyter, Berlin (1995).
6. Hartley, D.S., III.: Military operations research: presentation at ORSA/TIMS meetings. Operations Research. 40, 640–646 (1992).
7. Kao, C., S.T. Liu.: Predicting bank performance with financial forecasts: a case of Taiwan commercial banks. Journal of Banking and Finance. 28, 2353–2368 (2004).
8. Kao, C., S.T. Liu.: Stochastic data envelopment analysis in measuring the efficiency of Taiwan commercial banks. European J. Operational Research. 196, 312–322 (2009).
9. Lane, M.S., A.H. Mansour, J.L. Harpell.: Operations research techniques: a longitudinal update 1973–1988. Interfaces. 23, 63–68 (1993).
10. Schenkerman, S.: Sensitivity of linear programs to related changes in multiple inputs. Decision Sciences. 24, 879–891 (1993).
11. Serafini, P.: Linear programming with variable matrix entries. Operations Research Letters. 33, 165–170 (2005).
12. Wagner, H.M.: Global sensitivity analysis. Operations Research. 43, 948–969 (1995).
13. Wang, H.F., C.S. Huang.: Multi-parametric analysis of the maximum tolerance in a linear programming problem. European Journal of Operational Research. 67, 75–87 (1993).
14. Ward, J.E., R.E. Wendell.: Approaches to sensitivity analysis in linear programming. Annals of Operations Research. 27, 3–38 (1990).
15. Wendell, R.E.: Using bounds on the data in linear programming: The tolerance approach to sensitivity analysis. Mathematical Programming. 28, 304–322 (1984).
16. Wendell, R.E.: The tolerance approach to sensitivity analysis in linear programming. Management Science. 31, 564–578 (1985).
17. Wendell, R.E.: Tolerance sensitivity and optimality bounds in linear programming. Management Science. 50, 797–803 (2004).
18. Wondolowski, F.R. Jr.: A generalization of Wendell's tolerance approach to sensitivity analysis in linear programming. Decision Sciences. 22, 792–810 (1991).

Quantifying Retardation in Simulation Based Optimization

Andreas Griewank, Adel Hamdi, and Emre Özkaya

Abstract In many applications one wishes to optimize designs on the basis of an established simulation tool. We consider the situation where "simulation" means solving a system of state equations by a fixed point iteration. "Optimization" may then be performed by appending an adjoint solver and an iteration step on the design variables. The main mathematical goal of this chapter is to quantify and estimate the *retardation factor*, i.e., the complexity of an optimization run compared to that of a single simulation, measured in terms of contraction rates. It is generally believed that the retardation factor should be bounded by a reasonably small number irrespective of discretization widths and other incidental quantities. We show that this is indeed the case for a simple elliptic control problem, when the state equations are solved by Jacobi or a multigrid V-cycle. Moreover, there is strong dependence on a regularization term. This is also shown to be true when the state equation is solved by Newton's method and the projected Hessian is explicitly available

A. Griewank (✉)
Department of Mathematics, Humboldt University Berlin, Unter den Linden 6, 10099 Berlin, Germany
e-mail: griewank@mathematik.hu-berlin.de

A. Hamdi
Laboratoire de mathématiques LMI, Institut National des Sciences Appliques de Rouen Avenue de l'Universit 76801 Saint-Etienne-du-Rouvray Cedex, France
e-mail: adel.hamdi@insa-rouen.fr

E. Özkaya
Computational Mathematics Group, CCES, RWTH Aachen University, Schinkelstr. 2, 52062 Aachen, Germany
e-mail: ozkaya@mathcces.rwth-aachen.de

A. Chinchuluun et al. (eds.), *Optimization, Simulation, and Control*, Springer Optimization and Its Applications 76, DOI 10.1007/978-1-4614-5131-0_6,

1 Introduction

For many actual or potential users of optimization technology the transition from an existing simulation model or tool to a corresponding optimization method is anything but simple. Often the prospect of redesigning and reimplementing their models to interface with a classical NLP package is so daunting that the idea of employing calculus-based optimization methods is abandoned altogether. Especially for discretization of PDEs the specifications of Jacobian sparsity patterns and the provision of partial derivative values can be extremely laborious. Moreover, due to sheer size and lack of structure that effort may not even lead to an efficient solver for the purpose of *system simulation*, i.e., the resolution of a nonlinear *state equation*

$$c(y,u) \;=\; 0 \quad \text{with} \quad c : Y \times U \to Y \;. \tag{1}$$

Here $u \in U$ is a design vector, which is kept fixed as $c(y,u) = 0$ is solved for the corresponding *state vector* $y = y_*(u) \in Y$. In aerodynamics u may represent a parameterization of a wing shape, which together with appropriate free stream boundary conditions determine the flow field y around the wing. In climatological studies u is a vector of model parameters and y is a vector of prognostic variables, i.e., ocean and atmosphere flow velocities and temperature.

In the right function space setting one may assume that the linearized operator $c_y \equiv \nabla_y c$ has a bounded inverse, but often the Jacobian obtained for a suitable discretization is so unwieldy that no Newton-like solver can be realized. In this chapter we address also the situation where one has to make do with a fixed point iteration

$$y_{k+1} \;=\; G(y_k,u) \;\text{ with }\; G(y,u) = y \;\Leftrightarrow\; c(y,u) = 0, \tag{2}$$

which frequently may be interpreted as pseudo-time stepping on an underlying instationary version of the state equation. For example, in aerodynamics one uses quasi-unsteady formulations which are solved by explicit central finite volume schemes stabilized by artificial dissipation and Runge–Kutta time integration [14]. In our days, these schemes are most efficient in combination with geometric multigrid [13, 16].

There is some steady progress in simulation models and of course computing power. Nevertheless, we have to assume that in many application areas a single state equation solved to full accuracy takes several hours or even days on a single machine. Compared to the effort of gaining feasibility for a given u in this way, the evaluation of an objective

$$f(y,u) : Y \times U \to \mathbb{R} \tag{3}$$

which may represent a fitting functional or other performance indices is usually a cheap by-product. Hence the transition from simulation to optimization may appear at first quite simple. Consequently there are many software tools that implement

assorted direct search strategies based on computing solutions y_k with $c(y_k, u_k) \approx 0$ and then $f(y_k, u_k)$ at a cloud of sampling points u_k in the design domain U.

Disregarding occasional claims of global convergence to global minima on nondifferentiable problems, one can expect that local minima will be approximately located by Nelder Mead type algorithms [15] if $c(y,u)$ and $f(y,u)$ are at least once continuously differentiable. Instead of the linear models on which Nelder Mead is based, one may of course fit other *surrogate objective* functions through the points evaluated at any stage. To construct a reasonable *response surface* of that kind or to approximate a single gradient by differences one needs at the very least $\dim(u)$ evaluations. Hence there is no hope to achieve what might be called the principle of bounded deterioration of optimal design:

$$\text{Cost Optimization} \sim \text{Cost Simulation}.$$

This goal has been achieved in several projects of the DFG-sponsored priority program 1253 (e.g., see [10, 11]), but a general theoretical statement is as yet not even on the horizon. It is not difficult to construct an example where, depending on a parameter d in the objective function, the distance between a given initial state $y_0 \approx y(u_0)$ and the optimized state $y_* = y(u_*)$ grows linearly with d. One then must expect that reaching y_* from y_0 while staying reasonably feasible and gradually changing u takes a number of solver steps that are also proportional to d. We assume here that the problem at hand is so nonlinear or otherwise difficult that this continuation-like approach cannot be avoided by jumping more or less directly from u_0 and y_0 into the vicinity of u_* and y_*. Then the constant in the above proportionality claim must also grow with d, which one might view as quantifying the difficulty of the optimization task relative to the simulation task for given starting design u_0 and state y_0. For an elliptic PDE example also used by Kunisch and Schulz [12], where d represents the reciprocal of a regularization parameter in the objective is given in Sect. 4 on page 9. In that case we were able to explicitly compute the optimal retardation factor for our currently preferred design space preconditioner, which turns out to be a multiple of I, in this special situation.

This chapter is organized as follows: In Sect. 2 we consider the general design optimization problem and its solution in a one-shot fashion using a design-space preconditioner $B \succ 0$. We discuss in particular the key characteristics that determine the algorithmic performance. In Sect. 3 we consider a model scenario, where the state equation is solved by Newton's method, the adjoint is separable, and the projected Hessian is evaluated exactly. This analysis applies similarly to hierarchical approaches, where the state equation is resolved rather accurately after each design change. Here the one-shot approach yields a retardation factor that is proportional a parameter γ that depends on the size of the Lagrange Hessian with respect to the design alone relative to the full projected Hessian. In Sects. 4 and 5 we analyze the minimal retardation for Jacobi and multigrid on the test problem of Kunisch and Schulz in 1D. In both cases, we obtain factors that are essentially mesh independent.

2 One-Shot Optimization and Problem Characteristics

In the remainder, we consider the following equality-constrained optimization problem:

$$\min_{(y,u)} \; f(u,y) \;\; \text{s.t.} \;\; y = G(u,y), \tag{4}$$

where G is contractive with respect to a norm $\|\cdot\|$. At least theoretically we may assume without loss of generality that $\|v\|^2 = v^\top v$ in the Euclidean norm, and then we obtain the Lagrangian function

$$L(u,y,\bar{y}) = f(y,u) + \bar{y}^\top (G(y,u) - y) \tag{5}$$

with $\bar{y}$ denoting the adjoint state vector, or co-state. Then the KKT conditions for a stationary point $(y_*, \bar{y}_*, u_*)$ of the problem (4) are

$$\begin{aligned} 0 &= G(y_*,u_*) - y_* \\ 0 &= L_y(y_*,\bar{y}_*,u_*) \equiv f_y(y_*,u_*) + \bar{y}_*^\top G_y(y_*,u_*) - \bar{y}_* \\ 0 &= L_u(y_*,\bar{y}_*,u_*) \equiv f_u(y_*,u_*) + \bar{y}_*^\top G_u(y_*,u_*). \end{aligned} \tag{6}$$

This coupled system of equations naturally leads to the fixed point iteration

$$\begin{bmatrix} y_{k+1} \\ \bar{y}_{k+1} \\ u_{k+1} \end{bmatrix} = \begin{bmatrix} G(y_k,u_k) \\ \bar{y}_k + L_y(y_k,\bar{y}_k,u_k) \\ u_k - B_k^{-1} L_u(y_k,\bar{y}_k,u_k) \end{bmatrix}. \tag{7}$$

Here B_k is a suitable design space preconditioner that is crucial for the success of the method and will be analyzed in the remainder. The other essential ingredient for the efficiency of the approach is the ability to evaluate the full gradient $(L_y, L_u) \in \mathbb{R}^{n+m}$ at a fixed multiple of the cost of evaluating (f,G) and thus L by itself. This can always be achieved by automatic differentiation in the reverse mode as described for example in [6].

First differentiating the new iterate $(y_{k+1}, \bar{y}_{k+1}, u_{k+1})$ with respect to the old $(y_k, \bar{y}_k, u_k)$, then dropping the iteration counter k, and finally evaluating at $(y_*, \bar{y}_*, u_*)$, we obtain the coupled Jacobian

$$J_* = J\Big|_{(y_*,\bar{y}_*,u_*)} = \begin{bmatrix} G_y & 0 & G_u \\ L_{yy} & G_y^\top & L_{yu} \\ -B^{-1}L_{uy} & -B^{-1}G_u^\top & (I - B^{-1}L_{uu}) \end{bmatrix} \in \mathbb{R}^{(2n+m)\times(2n+m)}. \tag{8}$$

The asymptotic rate of convergence will be determined by the spectral radius $\rho_* = \rho(J_*)$, the maximal modulus of any eigenvalue λ in the spectrum of J_*. It was

first observed in [7] that block elimination yields for $\lambda \in \mathrm{spect}(J_*) \setminus \mathrm{spect}(G_y)$ the characterization

$$\det[P(\lambda)] = 0 \quad \text{with} \quad P(\lambda) = (\lambda - 1)B + H(\lambda), \tag{9}$$

where

$$H(\lambda) = Z(\lambda)^\top \begin{bmatrix} L_{yy} & L_{yu} \\ L_{uy} & L_{uu} \end{bmatrix} Z(\lambda) \quad \text{for} \quad Z(\lambda) \equiv \begin{bmatrix} (\lambda I - G_y)^{-1} G_u \\ I \end{bmatrix}. \tag{10}$$

We observe that $H(\lambda)$ is a projection of the Lagrangian Hessian $\nabla^2 L$ onto the range of the matrix $Z(\lambda) \in \mathbb{R}^{(n+m)\times m}$. It is easy to see that the columns of $Z(1)$ span the tangent space on the feasible set $\{G(y,u_*) = y\}$ and that $H(1)$ is thus the reduced Hessian of our constrained optimization problem. For a suitable choice of B we will try to predict and minimize the

$$\text{Retardation factor}: \quad r \equiv \frac{(1-\rho(J_*))}{(1-\rho(G_y))} \approx \frac{\ln(\rho(J_*))}{\ln(\rho(G_y))}.$$

Naturally this ratio is only a somewhat idealized measure of the slow-down in going from simulation to optimization. Not only are initial conditions neglected but also the fact that each execution of the coupled iteration will be some five times as expensive as that of G by itself is not accounted for. Now we may characterize the design optimization problem and its one-shot solution in terms of the following quantities:

- The Jacobian G_y and the Hilbert norm with respect to which it is contractive such that $\|G_y\| \leq \rho$. These objects form the linchpin of the whole one-shot framework. In case of the Newton iteration $G(y,u) = y - c_y(y,u)^{-1}c(y,u)$, we obtain $G_y = 0$ and $\rho(G_y) = 0$ at all feasible points. This situation will be considered as limiting scenario for methods that are rapidly converging such as full multigrid.
- The partial Hessian L_{yy} represents both the nonlinearity of the fixed point solver and the sensitivity of the dual w.r.t. to the primal. The norm $p \equiv \|L_{yy}\|$ may be viewed as a measure of the coupling between the two.
- The Jacobian G_u represents the sensitivity of the primal state equation w.r.t. design changes. We may assume that G_u has full rank and then reparameterize the design space such that $G_u^\top G_u = I$, at least theoretically.
- The mixed derivative L_{yu} represents the sensitivity of adjoint equation with respect to design. Often one has a separable adjoint in that $L_{yu} = 0$. Generally, we may use the ratio $q \equiv \max_v \|L_{yu}v\|/\|G_u v\|$ as measure of (non-) separability. When G_u is orthogonal we have simply $q \equiv \|L_{yu}\|$.
- The positive definiteness condition $H(1) \succ 0$ for $(y,u) \approx (y_*,u_*)$ represents second-order sufficiency, a mild condition, which is completely independent of the chosen iteration function G.

- The global definiteness condition $\nabla^2_{y,u} L \succ 0$ for $(y,u) \approx (y_*, u_*)$ on the full Lagrange Hessian implies $H(\lambda) \succ 0$ for all λ. These matrices are for $\lambda \neq 1$ very much dependent on G, and the condition seems unreasonably strong especially if $\dim(y) \gg \dim(u)$ as will typically be the case.
- The partial Hessian $L_{uu} \succ 0$ need not be positive definite for second-order sufficiency $H(1) \succ 0$. However, this property is often guaranteed by a regularization of the design vector u and when L_{uu} is gradually scaled up to infinity we have likely $u_* \to 0$ and $r \to 0$. Conversely one might have $\|u_*\| \to \infty$ and $r \to \infty$ as L_{uu} becomes small and $H(1)$ nearly singular.

For our analysis of Newton method in the separable case we have to consider the generalized problem $G_u^\top L_{yy} G_u v = \gamma H(1) v$ assuming of course that $H(1) = G_u^\top L_{yy} G_u + L_{uu}$ is positive definite. If the partial Hessian L_{uu} is positive semidefinite we see immediately that the eigenvalue γ cannot be larger than 1.

We will denote the diagonalization of $G_u^\top L_{yy} G_u$ with respect to $H(1)$ as $\Gamma = \mathrm{diag}(\gamma_i)_{i=1}^n$ and use $\gamma \equiv \max\{|\gamma_1|, |\gamma_n|\} \equiv \|\Gamma\|$ as a measure of irregularity. If L_{uu} completely dominates $G_u^\top L_{yy} G_u$ the parameter γ tends to zero, so that $\Gamma = 0$ represents a maximally regularized solution.

At the end of this introductory section, we derive a suitable preconditioner B for general separable problems. That is the result given by the following proposition:

Proposition 1 (Preconditioner for general separable case). *If $L_{yu} = 0$ the choice*

$$B \equiv \alpha G_u G_u^\top + L_{uu} \;\; \textit{where} \;\; \alpha = \|L_{yy}\| / (1 - \|G_y\|)^2$$

ensures that the matrix $P(\lambda)$ introduced in (9) cannot be singular for $\lambda \leq -1$ or $\lambda = 1$.

Proof. In view of (10), we get for all v in $\mathbb{C}^n$ and λ in $\mathbb{C}$ such that $|\lambda| \geq 1$,

$$\begin{aligned} \overline{v}^\top H(\lambda) v - \overline{v}^\top L_{uu} v &= \overline{v}^\top G_u (\lambda I - G_y)^{-1} L_{yy} (\lambda I - G_y)^{-1} G_u v \\ &\leq \|L_{yy}\| \, \|(\lambda I - G_y)^{-1} G_u v\|^2 \leq \|L_{yy}\| \|G_u v\|^2 / (1-\rho)^2 \end{aligned}$$

where $\rho = \|G_y\|$. Then the given choice of B ensures

$$\overline{v}^\top H(\lambda) v \leq \overline{v}^\top B v, \quad \text{for all } \; v \in \mathbb{C}^n \;\text{ and }\; \lambda \in \mathbb{C} \text{ with } \; |\lambda| \geq 1$$

Therefore, using (9), we find for all real numbers λ such that $\lambda \leq -1$,

$$P(\lambda) = (\lambda - 1) B + H(\lambda) \preceq \lambda B \prec 0$$

which implies that $P(\lambda)$ cannot be singular for $\lambda \leq -1$; thus, it cannot be an eigenvalue of J_*. Furthermore, since the second-order sufficiency ensures that $P(1) = H(1) \succ 0$, then $\lambda = 1$ cannot be also an eigenvalue of J_*. □

So far the efforts to derive a B that excludes any eigenvalues outside the unit ball have not been successful. Of course very large positive definite B will do trick, but the resulting convergence is likely excruciatingly slow. The proposed B need not be evaluated explicitly but can be approximated by BFGS updates [9]. Note that in the nonseparable case, the preconditioner B must contain a second term $\beta L_{uy} L_{yu}$, where β is a suitable penalty parameter as discussed in [8]. Loosely speaking the term $\alpha G_u G_u^\top$ cautions against rapid changes in the design variable components that have a strong effect on the primal equations, and $\beta L_{uy} L_{yu}$ would do the same with respect to $L_y = 0$, the adjoint equation. All this is relative to the convergence rate ρ which quantifies the ability of the given fixed point solver to recover feasibility. Before applying this kind of preconditioner, we will first examine the use of what most people would consider a more natural choice, namely $B = H(1)$.

3 The Newton Scenario for Separable Adjoints

In this section, we analyze the very nice situation when we have not only $G_y = 0$ and thus $G_u = dy/du$ but also $L_{yu} = 0$. We expect the resulting observations also to apply when G represents an inner iteration like several multigrid cycles that resolve the state equation up to higher-order terms before the design variables are changed once more. One can easily check that separability implies $H(-1) = H(1)$ and that the eigenvalues $\lambda \in \mathrm{spect}(J_*) \setminus \mathrm{spect}(G_y)$ are characterized now by singularity of the matrix

$$\begin{aligned} P(\lambda) &= (\lambda - 1)B + G_u^\top L_{yy} G_u / \lambda^2 + L_{uu} \\ &= (\lambda - 1)B + G_u^\top L_{yy} G_u (1/\lambda^2 - 1) + H(1) \end{aligned}$$

which is a special case of (9). Under the second-order sufficiency condition $H(1) \succ 0$, we may transform $P(\lambda)$ as discussed above to

$$\tilde{P}(\lambda) = (\lambda - 1)\tilde{B} + (1/\lambda^2 - 1)\Gamma + I.$$

Since B can be selected freely, we will assume that it is given by $B = H(1)/\eta$, i.e., the projected Hessian itself scaled by the reciprocal of a step multiplier $\eta > 0$. For this seemingly ideal choice, we obtain $\tilde{B} = I/\eta$ and thus the completely diagonal matrix

$$\tilde{P}(\lambda) = \big[(\lambda - 1)/\eta + 1\big] I + (1/\lambda^2 - 1)\Gamma.$$

It is singular if one of its diagonal elements vanishes, which leads to the set of rational equations

$$\frac{(\lambda - 1)/\eta + 1}{1 - 1/\lambda^2} = \gamma_i \in \mathbb{R} \text{ for some } i \in \{1, \ldots, n\}. \tag{11}$$

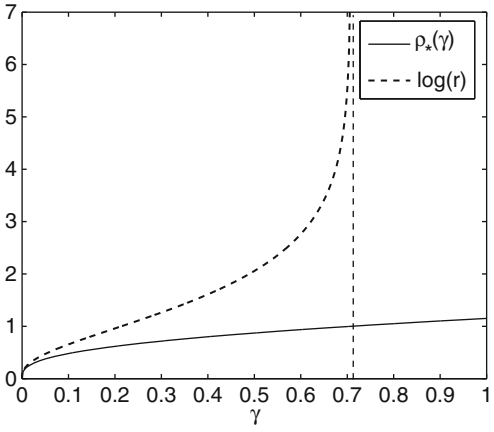


Fig. 1 Dependence of the convergence rate ρ_* and the logarithm of the retardation factor r on the cross-term size γ

Proposition 2 (Full step convergence). *For $\eta = 1$ the maximal modulus ρ_* of any solution to (11) is less than 1 if and only if $\gamma = \|\Gamma\| < 1/\sqrt{2}$, and we have then $\rho_* < \sqrt[3]{2\gamma}$.*

Proof. Taking the reciprocals, we find for $\hat{\lambda} = 1/\lambda$ the cubic equation $1/\gamma_i = \hat{\lambda} - \hat{\lambda}^3$. The elementary examination of its graph shows that this equation has more than one real root exactly if $|\gamma_i| \geq \frac{3}{2}\sqrt[3]{3} \approx 2.6$. Then one or two of the three roots lie in the interval $(-1,1)$ so that the reciprocal $\hat{\lambda} = 1/\lambda$ will be larger than 1 in modulus, and we must have $\rho > 1$. If $\hat{\lambda} = (\cos\varphi + i\sin\varphi)/\rho$ is a complex root with $\sin\varphi \neq 0$, one obtains after some elementary manipulations that $|\lambda| = 1/|\hat{\lambda}| = \rho = 1 + 2\cos(2\varphi)$. Hence $\rho < 1$ can only happen if $\varphi \in \left(\frac{\pi}{4}, \frac{3\pi}{4}\right) \cup \left(\frac{5\pi}{4}, \frac{7\pi}{4}\right)$. In fact we have for $\varphi = \pm\frac{\pi}{4}$ exactly $\hat{\lambda} = \frac{1}{\sqrt{2}}(1 \pm i) = \frac{1}{\sqrt{2}}(1 \mp i)^{-1} = \lambda^{-1}$ with

$$\frac{\lambda^3}{\lambda^2 - 1} = \frac{(-1 \mp i)\dfrac{1}{\sqrt{2}}}{-1 \mp i} = \frac{1}{\sqrt{2}} = \frac{-(-\lambda)^3}{(-\lambda)^2 - 1}.$$

Thus, we need $\gamma < 1/\sqrt{2}$ in order to obtain convergence. The final assertion follows from

$$\frac{|\lambda|^3}{2} \leq \frac{|\lambda|^3}{1 + |\lambda|^2} \leq \left|\frac{\lambda^3}{\lambda^2 - 1}\right| = |\gamma_i| \leq \gamma. \qquad \square$$

According to the proposition, the problem must be regular enough such that γ is less than $1/\sqrt{2}$. The actual retardation factor is given by $r = 1/(1 - \rho_*(\gamma))$, which is plotted in Fig. 1 together with ρ_*. Note that the numerator is simply 1 since we apply Newton's method. Only for rather small γ do we obtain rapid convergence, which is

a little surprising. When the generalized eigenvalues γ_i lie outside $\left(-1/\sqrt{2}, 1/\sqrt{2}\right)$ but are bounded above by 1 then convergence can be ensured with the help of a step multiplier of size $1/(\gamma+1)$.

Proposition 3 (Convergence with step size control). *If $B = H(1)/(1+\gamma)$ with $\gamma = \|\Gamma\|$ and $L_{uu} \succ 0$, then the spectral radius ρ_* is always contained in the interval $\left[\gamma/(\gamma+1), 1\right)$. Moreover, we conjecture that exactly*

$$\rho_* = \sqrt[3]{\frac{\gamma}{1+\gamma}} \approx 1 - \frac{1}{3\gamma}.$$

Proof. Substituting $\eta = 1+\gamma$ into (11), we obtain the family of equations

$$Q_i(\lambda) = \lambda + \frac{\gamma_i}{(1+\gamma)\lambda^2} - \frac{\gamma_i+\gamma}{1+\gamma} = 0 \text{ for some } i = 1,..,n.$$

Let us add an equation $Q_0(\lambda)$ with $\gamma_0 \equiv -\gamma$ if $\max\{|\gamma_i|\}$ is attained for $\gamma_i > 0$. Hence, we may order the γ_i as $-\gamma = \gamma_0 \le \gamma_1 \le \gamma_i \le \ldots \le \gamma_n \le \gamma$. We have for all $i \ge 0$,

$$Q_i(-1) = -1 - \frac{\gamma}{1+\gamma} = \frac{-(1+2\gamma)}{1+\gamma} < \ 0 \ < \ Q_i(1) = 1 - \frac{\gamma}{1+\gamma} = \frac{1}{1+\gamma}L.$$

Let us denote by $P_i(\lambda) = \lambda^2 Q_i(\lambda)$ the corresponding cubic polynomial. If $\gamma_i = 0$, we have a double root at $\lambda = 0$ and a nontrivial root at $\lambda_i = \gamma/(1+\gamma) \in (0,1)$. For all $\gamma_i \neq 0$, we find $Q_i(0) = \infty sign(\gamma_i)$. So that by the mean value theorem and our observation regarding $Q_i(\pm 1)$, there exist real roots

$$\lambda_i \in (-1,0), \text{ if } \gamma_i > 0 \quad \text{and} \quad \lambda_i \in (0,1), \text{ if } \gamma_i < 0.$$

In particular, we have $Q_0(\lambda_0) = \lambda_0 - \gamma/(1+\gamma)\lambda_0^2$ and thus, $\lambda_0 = \sqrt[3]{\gamma/(1+\gamma)}$. Furthermore, if $\gamma_i > 0$, then we have

$$Q_i\left(\frac{-\gamma_i}{1+\gamma}\right) = \frac{-(2\gamma_i+\gamma)}{1+\gamma} + \frac{1+\gamma}{\gamma_i} = \frac{1+\gamma^2+2\gamma-2\gamma_i^2-\gamma_i\gamma}{(1+\gamma)\gamma_i} \ge \frac{1+2\gamma(1-\gamma)}{(1+\gamma)\gamma_i} > 0.$$

Besides, we have

$$Q_i(-\lambda_0) = -\lambda_0 + \frac{\gamma_i}{(1+\gamma)\lambda_0^2} - \frac{\gamma_i+\gamma}{1+\gamma} = \frac{\gamma_i-\gamma}{(1+\gamma)\lambda_0^2} - \frac{\gamma_i+\gamma}{1+\gamma} \le 0.$$

Hence, we conclude that

$$\lambda_i \in [-\lambda_0, -\gamma_i/(1+\gamma)), \qquad \text{if } \gamma_i > 0.$$

Similarly, we derive for $\gamma_i < 0$ that

$$Q_i\left(\frac{\gamma}{1+\gamma}\right) = \frac{\gamma_i(1+\gamma)}{\gamma^2} - \frac{\gamma_i}{1+\gamma} = \frac{\gamma_i(1+2\gamma)}{\gamma^2(1+\gamma)} < 0$$

and

$$\begin{aligned} Q_i(\lambda_0) &= \lambda_0 + \frac{\gamma_i}{(1+\gamma)\lambda_0^2} - \frac{\gamma_i+\gamma}{1+\gamma} \\ &= \frac{\gamma+\gamma_i}{(1+\gamma)\lambda_0^2} - \frac{\gamma_i+\gamma}{1+\gamma} = \frac{\gamma_i+\gamma}{1+\gamma}\left(\frac{1}{\lambda_0^2} - 1\right) \geq 0 \end{aligned}$$

when we have used $\lambda_0 \in (0,1)$ for the last inequality. Therefore, we obtain

$$\lambda_i \in (\gamma/(1+\gamma), \lambda_0], \qquad \text{if } \gamma_i < 0.$$

Thus the modulus $|\lambda_i|$ of the real roots λ_i is bounded above by $\lambda_0 = \sqrt[3]{\gamma/(1+\gamma)}$, which motivates our conjecture. However, each cubic polynomial $P_i(\lambda)$ has another pair of roots $\lambda^{\pm}$ which must satisfy

$$|\lambda_i^+||\lambda_i^-||\lambda_i| = |P_i(0)| = |\gamma_i|/(1+\gamma) \leq \gamma/(1+\gamma)$$

and hence using the common lower-bound $|\gamma_i|/(1+\gamma)$ on $|\lambda_i|$, we find $\min(|\lambda_i^-||\lambda_i^+|) < 1$. Finally, we observe from our sign conditions that if one of the two is real, the other must be too, and both must be smaller than 1 in size. The same follows if they form a complex conjugate pair so that also $|\lambda_i^-| = |\lambda_i^+| < 1$. □

Even when our conjecture is valid the convergence rate $\sqrt[3]{\gamma/(1+\gamma)}$ is obviously not very good unless $\gamma \approx 0$, which indicates a rather large L_{uu}. If the Hessian $H(1) = G_u^\top L_{yy} G_u + L_{uu}$ is only just positive definite with the negative curvature of the first term being just balanced by the second, then $\gamma = -\lambda_1$ can be arbitrarily large and the same holds for the retardation factor

$$r = 1/(1-\rho_*) = 1/(1-\sqrt[3]{\gamma/(1+\gamma)}) \approx \gamma/3.$$

This means that the effort in resolving the state equation rather accurately at each inner loop of the optimization calculation does not pay off, unless we have strong regularization. We conclude that, contrary to what one might have expected, the seemingly natural optimization step $-H(1)^{-1}L_u$ will be too large and lead to blow up unless the cross-term γ is quite small. As we have shown, one can remedy the situation by cutting the step size back by a factor of order $1/(1+\gamma)$, but the resulting retardation factor grows proportional to γ. Hence, solving the state equation quite accurately at each optimization step does not really pay off even if the projected Hessian $H(1)$ is evaluated or approximated at a reasonable cost. It is also remarkable that we need the condition that L_{uu} is positive semi-definite so that the elements of

Γ are no greater than 1. Hence, in this sense the results apply only to regularized problems. For the full step method, our convergence conditions are of course sharp, but probably and even more conservative step-size control would still work when there are generalized eigenvalues greater than 1.

4 Jacobi Method on an Elliptic Problem

Now let us consider a much slower iterative solver, namely the Jacobi method applied to the standard elliptic regulator problem. Here, we find that the preconditioner should be a multiple of the identity and its optimal scaling can be found by solving a system of three cubic polynomials, which can be reduced to a single polynomial in the convergence factor ρ_*. We use the 1D model optimization problem of tracking type:

$$f(y,u) = \frac{1}{2}\int_0^1 (y(t)-z(t))^2(t)\mathrm{d}t + \frac{\mu}{2}\int_0^1 u^2(t)\mathrm{d}t,$$

where the state y and the control u are linked by the state equation

$$-y''(t) = u(t), \text{ for } t \in [0,1] \text{ with } y(0) = 0,\ y(1) = 0.$$

Here μ is the regularization parameter that is a strictly positive, and z denotes the desired target state. We discretize the Laplacian term using central finite differences on an equidistant mesh with mesh size $h = 1/(n+1)$ where $n \in \mathbb{N}$. Given $z \in \mathbb{R}^n$ we obtain as for example in [5] the following discretized optimization problem:

$$\min_{(y,u)\in\mathbb{R}^{2n}} f(y,u) = \frac{h}{2}\|y-z\|^2 + \frac{\mu h}{2}\|u\|^2 \quad \text{s.t. } Cy = u.$$

Here C is the tridiagonal matrix defined by $C = -\text{tridiag}(1,-2,1)/h^2$. For solving $Cy = u$, we obtain the Jacobi iteration

$$y = G(y,u) \equiv Jy + \frac{h^2}{2}u.$$

It is well known that the eigenvalues of the matrix J are given by

$$c_i \equiv -\cos(i\pi h), \quad \text{for } i = 1,..,n. \tag{12}$$

Hence, the spectral radius of the symmetric matrix J is, given by $\rho(J) = \cos(\pi h) \approx 1 - \frac{1}{2}h^2\pi^2 < 1$. The Lagrangian defined in (5) becomes

$$L(y,\bar{y},u) = \frac{h}{2}\|y-z\|^2 + \bar{y}^T Jy + \frac{h^2}{2}\bar{y}^T u + \frac{\mu h}{2}\|u\|^2 - \bar{y}^\top y.$$

Therefore, the fixed point iteration (7) takes the form

$$\begin{aligned} y_{k+1} &= Jy_k + 0.5h^2 u_k \\ \bar{y}_{k+1} &= hy_k + J\bar{y}_k \\ u_{k+1} &= u_k - B_k^{-1}(\mu h u_k + 0.5h^2 \bar{y}_k). \end{aligned}$$

Hence the characteristic quantities discussed in Sect. 2 are given by

$$G_u = 0.5h^2 I,\ G_y = J,\ L_{yy} = hI,\ L_{uu} = \mu h I,\ L_{yu} = 0,\ q = 0 \tag{13}$$

and the projected Hessian takes the form

$$H(\lambda) = \mu h I + (\lambda I - J)^{-1} h^5/4$$

It should be noted that as a discretization effect, $H(1)$ stays positive definite even when μ tends to zero. This numerical regularization will also be observed for the retardation factor. Since both G_u and L_{uu} are multipliers of the identity the same is true for the B introduced in Proposition 1. Rather than using the conservative scaling by $p/(1-\rho)$ suggested there, we set $B = Ih/\eta$ and determine the scaling η that yields the optimal convergence rate. The extended iteration now takes the form

$$\begin{aligned} y_{k+1} &= Jy_k + 0.5h^2 u_k \\ \bar{y}_{k+1} &= h(y_k - z) + J\bar{y}_k \\ u_{k+1} &= -0.5\eta h \bar{y}_k + (1 - \eta\mu) u_k \end{aligned}$$

And the extended Jacobian is given by

$$J_* = \begin{bmatrix} J & 0 & \frac{h^2}{2} I \\ hI & J & 0 \\ 0 & -\frac{\eta h}{2} I & (1-\eta\mu) I \end{bmatrix}. \tag{14}$$

Now, $P(\lambda) = (\lambda - 1)B + H(\lambda)$ from (9) can be diagonalized by the eigenvectors of J yielding the matrix

$$\tilde{P}(\lambda) = (\lambda - 1)h/\eta I + 0.25h^5 \mathrm{diag}(1/(\lambda - c_i)^2)_{i=1}^n + \mu h I.$$

Hence, the eigenvalues of J_* satisfy one of the equations

$$P_i(\lambda) = (\lambda + \eta\mu - 1)(\lambda - c_i)^2 + h^4 \eta/4 = 0, \quad \text{for } i = 1, .., n,$$

where the c_i are the eigenvalues of J as given in (12). This equation can be rewritten as

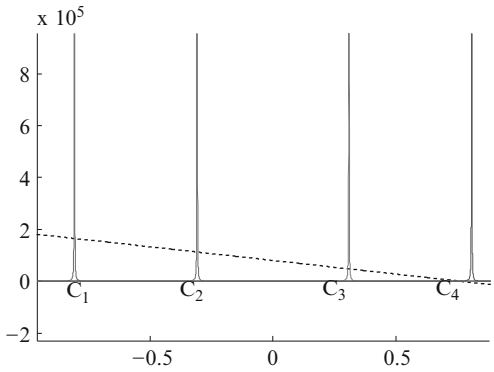


Fig. 2 Cubic equations for $n = 4$ and $\eta = 0.1$

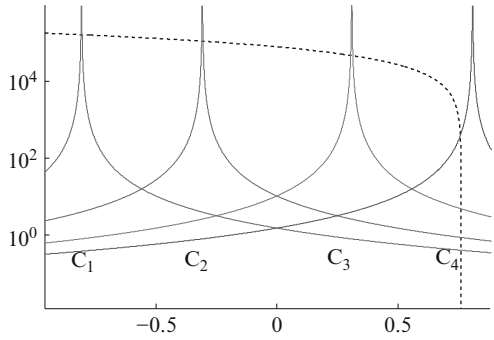


Fig. 3 Cubic equations in log scale

$$\frac{1}{(\lambda - c_i)^2} = \frac{\lambda + \eta\mu - 1}{-h^4\eta/4} \tag{15}$$

The left-hand sides have a quadratic pole at $\lambda = c_i$, and the common right-hand side is linear with respect to λ. For $n = 4$ and $\eta = 0.1$ the situation is depicted in Figs. 2 and 3 with the latter being scaled logarithmically.

$$\rho_* \equiv \max_{1 \le i \le n} \{|\lambda_i|, |\lambda_i^-|, |\lambda_i^+|\} \tag{16}$$

Proposition 4 (Algebraic characterization of optimal step size). *The optimal step size η, the resulting minimal convergence factor ρ_*, and an auxiliary eigenvalue λ can be computed by solving the following system of three cubic polynomials:*

$$\begin{aligned}
(-\rho_* + 1 - \Delta c)^2(-\rho_* - 1 + \eta\mu) + \frac{1}{4}h^4\eta &= 0 \\
(\lambda - 1 + \Delta c)^2(\lambda - 1 + \eta\mu) + \frac{1}{4}h^4\eta &= 0 \\
-(1 - \Delta c)^2 + \lambda\rho_*^2 + \eta(\frac{1}{4}h^4 - \mu(1 - \Delta c)^2) &= 0
\end{aligned} \tag{17}$$

where $\Delta c = 1 - c_n = 1 + c_1 \le 0.5\pi^2 h^2$.

Proof. For $\eta = 0$, we have $\rho_* = 1$, and for $\eta \to \infty$ the products $\lambda_i \lambda_i^- \lambda_i^+ = -P_i(0)$ go to infinity so that ρ_* becomes very large. Hence, by continuity a minimizer $\eta_* \geq 0$ must exist. Furthermore, the optimum can only be attained where there is a tie between at least two moduli. When all roots are real, then they are contained in the interval formed by the smallest and largest root of P_1. However, one can easily check that these cannot have the same modulus so that we must have a tie between a real eigenvalue of P_1 and the complex pair $\lambda_n^{\pm}$ of P_n. Rather than computing $\lambda_n^{\pm}$ directly, we impose the following conditions:

$$-P_n(0) = \lambda_n \lambda_n^- \lambda_n^+ = \lambda \rho^2 \quad \text{and} \quad P_n(\lambda_n) = 0.$$

This, together with the equation $P_1(-\rho_*) = 0$, gives the system of three cubic equations listed above. □

Due to the linearity with respect to η, we may rewrite the system of three cubic polynomials introduced in Proposition 4 as follows:

$$a_{11} + \eta a_{12} = 0 \ , a_{21} + \eta a_{22} = 0 \ , \ a_{31} + \eta a_{32} = 0$$

The existence of a solution $\eta \in \mathbb{R}$ requires that the three vectors (a_{1j}, a_{2j}) for $j = 1,2,3$ are pairwise linearly dependent so that we obtain equivalently the system of two equations $a_{11}a_{32} = a_{12}a_{31}$ and $a_{11}a_{22} = a_{12}a_{21}$ in λ and ρ_*. Since the first determinant equation is linear in λ, we can use it to express λ in terms of ρ_* and then substitute it into the second equation that yields a polynomial ρ that can be solved by standard software. In Fig. 4 we have plotted the resulting retardation factors as a function of the reciprocal $1/\mu$ of the regularization parameter μ and for $n = 32, 64, 128$. As one can see, the retardation factor is very small until $1/\mu$ is about 10^2, then grows quite rapidly until it becomes a linear function of $1/\mu$, and finally for very large $1/\mu$, it becomes constant. The plots in Fig. 4 were verified by computing and optimizing the spectral radius of J_* directly as a function of the scaling η. To understand better what is going on we can perform an asymptotic analysis. The optimal configuration obtained from the cubic system introduced in Proposition 4 always contains exactly one pair of complex conjugate eigenvalues as roots of P_n. Since their argument was observed to be rather small we probably do not lose much by picking η as the value for which P_n has $\lambda = \lambda_n^+ = \lambda_n^-$ as real root so that all $3n$ eigenvalues are in fact real. Moreover, by inspection of Figs. 1 and 2, we see that all of them must be contained in the interval $\left[-\lambda_1^+, \lambda_1^+\right]$ where $P_1(\lambda_1^+) = 0$. Hence, it follows that we must have contraction with $\rho_* = \lambda_1^+$. More specifically we find

Proposition 5 (Upper retardation bound). *P_n has a double root at*

$$\lambda = \frac{2}{3}(1 - \mu\eta) + \frac{1}{3}c_n$$

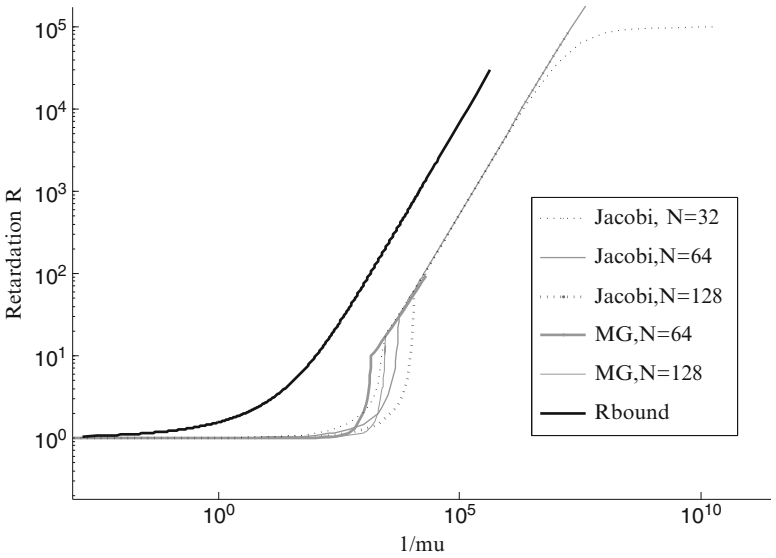


Fig. 4 Retardation factor for Jacobi and multigrid methods

when $\tilde{\eta} = \eta/\Delta c$ *satisfies the cubic equation:*

$$(1-\mu\tilde{\eta})^3 = \tilde{\eta}\frac{27h^4}{16(\Delta c)^2} \geq \tilde{\eta}\frac{27}{4\pi^4}$$

which yields the retardation bound

$$r = \frac{\Delta c}{(1-\rho_*)} \;\leq\; \frac{1}{\tilde{\eta}(\mu + h^4/16)} \;\leq\; \bar{r} \equiv \frac{1}{\tilde{\eta}\mu}$$

Proof. By differentiating $P_n(\lambda)$, we obtain

$$P_n'(\lambda) = 2(\lambda - c_n)(\lambda - 1 + \mu\eta) + (\lambda - c_n)^2$$

Hence, a double root $\lambda \neq c_n$ of $P_n(\lambda)$ must satisfy

$$0 = 2(\lambda - 1 + \eta\mu) + \lambda - c_n = 3\lambda - 2 + 2\eta\mu - c_n$$

which yields immediately the first assertion. Substituting this value into $P_n(\lambda)$, we find after some elementary manipulation

$$P_n(\lambda) = \frac{-4}{27}(\Delta c - \mu\eta)^3 + \frac{h^4}{4}\eta = 0.$$

Division by Δc then yields the second assertion. Since $\Delta c/h^2 = \pi^2/2 - O(h^2)$ the relation between μ and $\tilde{\eta}$ is essentially independent of h for all sufficiently small h. Moreover, we find for the resulting $\rho_* = \lambda_1^+ \in (c_n, 1)$ that $0 = (\rho - c_1)^2(\rho + \eta\mu - 1) + h^4/4$ and hence with $\rho - c_1 \leq 2$

$$1 - \rho_* = \mu\eta + \frac{h^4\eta/4}{(\rho_* - c_1)^2} \geq \eta\mu + \frac{h^4\eta}{16}.$$

Multiplication of the reciprocal by Δc yields the last assertion. □

The last inequality in the preposition will be almost an equality as long as $\mu \gg h^4/16$ as we would normally assume. By solving for μ the cubic equation introduced in Proposition 5, we obtain the expression

$$\mu(\tilde{\eta}) \equiv \left(1 - \frac{3}{\sqrt[3]{4\pi^4}}\sqrt[3]{\tilde{\eta}}\right)/\tilde{\eta} \text{ for } \tilde{\eta} \in (0, 4\pi^4/27). \tag{18}$$

The bold top line in Fig. 4 was obtained by plotting the curve $(1/\mu(\tilde{\eta}), \bar{r}(\tilde{\eta}))$ parameterized by $\tilde{\eta}$. As one can see $\bar{r}$ is indeed an upper bound on the retardation and almost proportional to $1/\mu$ in a medium range where the fully optimized step parameter η yields a similar slope. However, the optimized version is always faster by a factor of a little more than 10. And depending on the mesh with $h = 1/(n+1)$ there is an extra gain when the regularization parameter μ is comparatively large.

When $h^4/4$ dominates μ the retardation factor reaches the constant given by $\Delta c/(1 - \rho_*)$ with $\tilde{\eta} = 4\pi^4/27$ and the ρ_* the largest root of $(\rho_* - 1)(\rho - c_1)^2 + h^4\tilde{\eta}/(4\Delta c)$. This contraction ratio is of size $1 - O(h^2)$ as one would expect for Jacobi method.

5 Multigrid Method

In this final section, we employ a standard V-cycle multigrid algorithm, see [1–4], to perform primal and dual iterations. Here, we aim to study the behavior of the already established retardation factor when we solve the same optimization problem using the multigrid algorithm with Jacobi smoother. By employing a standard V-cycle multigrid algorithm, the primal iterations take the form

$$y_{k+1} = G(y_k, u_k) \equiv C_{MG}y_k + Ku_k \tag{19}$$

when C_{MG} and K are two matrices. Then, the corresponding adjoint iteration is

$$\bar{y}_{k+1} = hy_k + C_{MG}^{\top}\bar{y}_k. \tag{20}$$

Since the state equation is linear all second derivatives of the Jacobian depend only on the objective and are therefore exactly the same as in Jacobian method,

see (13). As preconditioner we use the one proposed in Proposition 1 scaled by step multiplier η.

$$B = \frac{1}{\eta}\left(\frac{\alpha}{2}K^\top K + \mu h\right) \quad \text{where } K = G_u \text{ and } \alpha = h/(1-\rho(C_{MG}))$$

The Jacobian associated to the coupled full step iteration takes the form:

$$J_* = \begin{bmatrix} C_{GM} & 0 & K \\ hI & C_{GM}^\top & 0 \\ 0 & -B^{-1}K & 1 - B^{-1}\mu h \end{bmatrix}. \tag{21}$$

The spectral radius of this matrix can be computed for small-scale problems easily with computer packages.

On the same problem considered in the previous section, we computed the matrices C_{MG} and K for the coarse grid and fine grid pairs $(1/32, 1/64)$ and $(1/64, 1/128)$. Then, we computed the spectral radius of the coupled Jacobian for a range of choices of η to determine the minimal ρ_* approximately. The resulting retardation factors are also plotted in Fig. 4. As one can see, the dependence on the regularization coefficient μ is almost identical to the one observed for the Jacobi method. Of course the multigrid solver and the resulting optimization solver are much faster, but the ratio between their contraction factors seems to be pretty much the same and is of course again largely independent of the mesh size.

6 Summary and Conclusion

For several comparatively simple model scenarios we have examined the retardation factor in the transition from simulation to optimization by a one-shot method. For Jacobi and V-cycle multigrid on an elliptic model problem we observed a reasonable retardation that is proportional to the reciprocal of a regularization parameter but largely mesh independent. The results on a variant of Newton method is a little troubling. The simple-minded one-shot approach does not seem to make good use of a fast corrector and an accurate projected Hessian. We are currently looking for a modification that does not require the accurate solution of the linearized KKT system. Also one needs to extend the methodology to the nonseparable case.

References

1. A. Brandt, Multi-Level Adaptive Solutions to Boundary-Value Problems, Math. Comp. 31 p. 333–390, (1977).
2. A. Brandt, Multigrid Techniques: Guide with Applications to Fluid Dynamics, GMD-Studien. no 85, St. Augustin, Germany, (1984).

3. A. Brandt and N. Dinar, Multigrid Solutions to Elliptic Flow Problems In: S.V. Parter (ed.), Numerical Methods for Partial Differential Equations, Academic Press, New York, (1979)
4. A. Brandt, S. McCormick and J. Ruge, Multigrid Methods for Differential Eigenproblems, SIAM J. Sci. Stat. Comput. 4(2):244–260, (1983).
5. G.H. Golub and J. M. Ortega, Scientific Computing And Differential Equations: An Introduction To Numerical Methods, Academic Press, Boston, (1991).
6. A. Griewank, Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, Society for Industrial and Applied Mathematics, Philadelphia-USA, (2000)
7. A. Griewank, Projected Hessians for Preconditioning in One-Step One-Shot Design Optimization, Large Scale Nonlinear Optimization, p. 151–171,(2006).
8. A. Hamdi and A. Griewank, Properties of an Augmented Lagrangian for Design Optimization, Optimization Methods and Software, 25(4):645–664, (2009).
9. A. Hamdi and A. Griewank, Reduced Quasi-Newton Method for Simultaneous Design and Optimization, Computational Optimization and Applications, Springer Netherlands, (2009).
10. M. Hinze, M. Köster, S. Turek: A Space-Time Multigrid Solver for Distributed Control of the Time-dependent Navier-Stokes System, Priority Programme 1253, Preprint-Nr.: SPP1253-16-02 (2008).
11. R.H.W. Hoppe, Multilevel Based All-at-once Methods in PDE constrai Optimization with Applications to Shape Optimization of Active Microfluidic Biochips DFG SPP 1253 Annual Meeting, Kloster Banz, 21–23.09.2008, (2008).
12. K. Ito, K. Kunisch, I. Gherman, V. Schulz, Approximate Nullspace Iterations for KKT Systems in Model Based Optimization, SIAM Journal on Matrix Analysis and Applications, 31:1835–1847 (2010)
13. A. Jameson. Multigrid Algorithms for Compressible Flow Calculations. In W. Hackbusch and U. Trottenberg, editors, Multigrid Methods II, volume 1228 of Lecture Notes in Mathematics, pages 166–201. Springer, 1986.
14. A. Jameson, W. Schmidt, and E. Turkel, Numerical solutions of the Euler Equation by Fnite Volume Methods Using Runge-Kutta time-stepping schemes. AIAA 81–1259, 1981.
15. J. A. Nelder and R. A. Mead. A Simplex Method for Function Minimisation. Comput. J., 7:308–313, (1964).
16. R. C. Swanson and E. Turkel, Multistage Scheme with Multigrid for Euler and Navier-Stokes Equations (components and analysis), Technical Report 3631, NASA, (1997).

Evolutionary Algorithm for Generalized Nash Equilibrium Problems

Mend-Amar Majig, Rentsen Enkhbat, and Masao Fukushima

Abstract This paper considers a method for finding multiple, hopefully all, solutions of the generalized Nash equilibrium problem (GNEP). Based on a merit function of the quasi-variational inequality (QVI) problem to GNEP, we reformulated GNEP as an unconstrained global optimization problem. To deal with the latter problem, we employ the evolutionary algorithm with adaptive fitness functions which help to search multiple global solutions. Numerical experiments for some test problems show the practical effectiveness of the method.

Key words Heuristics • Evolutionary algorithm • Generalized Nash equilibrium problem

1 Introduction

The generalized Nash equilibrium problem (GNEP) is an extension of the classical Nash equilibrium problem, in which each players's strategy set depends on the other players' strategies. Up to date, there are only a handful practical and effective algorithms for solving GNEP.

In this chapter, we adopt the idea of reformulating GNEP as a quasi-variational inequality (QVI) [4]. We are particularly interested in finding solutions of the problem as many as possible. To achieve this task we will design a special evolutionary algorithm.

M.-A. Majig (✉) • R. Enkhbat
School of Mathematics and Computer Science, National University of Mongolia, Ulaanbaatar, Mongolia
e-mail: mendamarm@num.edu.mn; renkhbat46@yahoo.com

M. Fukushima
Graduate School of Informatics, Kyoto University, Kyoto, Japan
e-mail: fuku@i.kyoto-u.ac.jp

A. Chinchuluun et al. (eds.), *Optimization, Simulation, and Control*, Springer Optimization and Its Applications 76, DOI 10.1007/978-1-4614-5131-0_7,

The organization of the chapter is as follows. In Sect. 2, we give a brief description of GNEP. Section 3 considers reformulations of GNEP as a QVI and a global optimization problem. An evolutionary algorithm for solving the resulting unconstrained global optimization problem is given in Sect. 4. In Sect. 5, we put some numerical results of the proposed method for some test problems, and Sect. 6 concludes this chapter.

2 Generalized Nash Equilibrium Problem

Let N be the number of players. Each player $\nu \in \{1,2,\ldots,N\}$ controls the variables $x^\nu \in \Re^{n_\nu}$, and let $x = (x^1,\ldots,x^N) \in R^n$ be the vector formed by all these decision variables, where $n = \sum_{\nu=1}^{N} n_\nu$. To separate the νth player's variables within the vector x, we sometimes write $x = (x^\nu, x^{-\nu})$ and $n_{-\nu} = n - n_\nu$, where $x^{-\nu}$ represents all other players' variables.

Let $\theta_\nu : \Re^n \to \Re$ be the νth player's cost function. We assume that for each ν, θ_ν is continuous, and the functions $\theta_\nu(x) = \theta_\nu(x^\nu, x^{-\nu})$ are convex in x^ν. In the GNEP, each player's strategy x^ν belongs to a nonempty, closed and convex set $X_\nu(x^{-\nu})$ which depends on other players' strategies. We assume that X_ν are given by

$$X_\nu(x^{-\nu}) = \{x^\nu \in \Re^{n_\nu} | (x^\nu, x^{-\nu}) \in X\},$$

where $X \subset \Re^n$ is a nonempty, closed, and convex set which represents the joint constraints of all players $\nu = 1,\ldots,N$.

The generalized Nash game is to find a vector $x^* = (x^{*,\nu})_{\nu=1}^N \in R^n$, called a generalized Nash equilibrium (GNE), such that for each $\nu = 1,\ldots,N$, $x^{*,-\nu}$ is an optimal solution of the convex optimization problem in the variable x^ν with $x^{-\nu}$ fixed at $x^{*,\nu}$:

$$\begin{aligned} &\text{minimize } \theta_\nu(x^{*,-\nu}, x^\nu), \\ &\text{subject to } x^\nu \in X_\nu(x^{*,-\nu}). \end{aligned} \tag{1}$$

The existence of generalized Nash equilibria under some mild conditions has been guaranteed by the following theorem.

Theorem 2.1 ([5]). *Let a GNEP be given and suppose that*

(a) There exist N nonempty, convex, and compact sets $K_\nu \subset \Re^{n_\nu}$ such that for every $x \in \Re^n$ with $x^\nu \in K_\nu$ for every ν, $X_\nu(x^{-\nu})$ is nonempty, closed, and convex, $X_\nu(x^{-\nu}) \in K_\nu$, and X_ν , as a point-to-set map, is both upper and lower semicontinuous.

(b) For every player ν, the function $\theta_\nu(\cdot, x^{-\nu})$ is quasi-convex on $X_\nu(x^{-\nu})$. Then a GNE exists.

3 Equivalent Reformulations

There are several ways to reformulate GNEP as a global optimization problem [1,3,4,6], and we use the one proposed in [4]. If we define the set-valued function $\Omega : \Re^n \to 2^{\Re^n}$ by

$$\Omega(x) = \prod_{\nu=1}^{N} X_\nu(x^{-\nu}) \subseteq \Re^n$$

and the function $F : \Re^n \to \Re^n$ by

$$F(x) = (\nabla_{x^\nu} \theta_\nu(x))_{\nu=1}^N \in R^n,$$

then we can see [9] that x^* is a GNE if and only if $x^* \in \Omega(x^*)$ and

$$(y - x^*)^T F(x^*) \geq 0, \ \forall y \in \Omega(x^*). \tag{2}$$

The latter problem (3) is known as a QVI, and through its merit function the problem can be reformulated as a global optimization problem with zero global minimum value.

Theorem 3.2 ([5]). *Let a GNEP be given, and for $\forall \nu$, $X_\nu(x^{-\nu})$ is closed convex and $\theta_\nu(\cdot, x^{-\nu})$ is convex and continuously differentiable. Then, a point $\bar{x}$ is a GNE if and only if it is a solution of the QVI $(\Omega(x), F(x))$, i.e., to find $x^* \in \Omega(x^*)$ and*

$$\langle F(x^*), x - x^* \rangle \geq 0, \ \forall x \in \Omega(x^*), \tag{3}$$

where $F(x) := (\nabla_{x^\nu} \theta_\nu(x))_{\nu=1}^N$. From now on we assume that the set Ω is given as follows:

$$\Omega(x) = \{y \in R^n | \ g_i(x,y) \leq 0, \ i = 1, \ldots, m\},$$

where $g_i : R^n \times R^n \to R, \ i = 1, \ldots, m$, are functions such that $g_i(x, \cdot)$ are convex and differentiable for each fixed x. Define a set

$$S := \{x \in R^n | \ x \in \Omega(x)\}.$$

This set is called a feasible set of QVIP (3). Then by definition, we have

$$S = \{x \in R^n | g_i(x,x) \leq 0, \ i = 1, \ldots, m\}.$$

To reformulate QVIP as a global optimization problem, we use a so-called gap function. Below we enlist some merit functions for QVI.

- Gap function

$$\theta_c(x) := \sup_{y \in \Omega(x)} \langle F(x), x - y \rangle - \frac{c}{2} \langle x - y, G(x - y) \rangle,$$

  where G is a positive definite matrix, $c > 0$ a scalar.

- Linearized gap function

$$\theta_c^{\text{lin}}(x) = \sup_{y \in \Gamma(x)} \langle F(x), x-y \rangle - \frac{c}{2}\langle x-y, G(x-y) \rangle,$$

where $\Gamma(x)$ is a polyhedral of $\Omega(x)$ at x

$$\Gamma(x) := \{y \in R^n | g_i(x,x) + \langle \nabla_y g_i(x,x), y-x \rangle \leq 0,\ i = 1,\ldots,m\}.$$

The following theorem shows that the linearized gap function is indeed a merit function for QVIP.

Theorem 3.3 ([4]). *For each $x \in S$, we have $f_c^{\text{lin}}(x) \geq 0$. Moreover, x solves QVI (3) if and only if $f_c^{\text{lin}}(x) = 0$ and $x \in S$.*

Calculation of gap function requires solving convex minimization problem while that of D-gap function requires solving convex quadratic programming problem. With these merit functions we can reformulate QVI as the following constrained global optimization problem:

$$\begin{aligned} &\min \theta(x), \\ &\text{subject to } x \in S. \end{aligned} \tag{4}$$

These approaches enables us to consider a constrained global optimization problem instead of the original quasi variational inequality problem. We also can consider some unconstrained optimization reformulations.

- D-gap function

$$\theta_{ab} := \theta_a(x) - \theta_b(x),\ a < b,\ a,b > 0$$

- Linearized D-gap function

$$\theta_{ab}^{\text{lin}} := \theta_a^{\text{lin}}(x) - \theta_b^{\text{lin}}(x),\ a < b,\ a,b > 0$$

Theorem 3.4 ([4]). *For each $x \in \Re^n$, we have $f_{ab}^{\text{lin}}(x) \geq 0$. Moreover, x solves QVI (3) if and only if $f_{ab}^{\text{lin}}(x) = 0$.*

Global minimum value of the problem is known to be zero if GNE exists. The latter two gap functions we have reformulated the QVIP as the following unconstrained global optimization problem:

$$\min\ \theta_{ab}(x),\ x \in R^n. \tag{5}$$

In our approach, we use the linearized D-gap function for the reformulation and an evolutionary algorithm to find global optimal solutions. Since the global minimum value of the equivalent optimization problem is known, we will directly have the stopping condition for our evolutionary algorithm which is not available for general evolutionary algorithm.

4 Evolutionary Algorithm

Since we are searching for multiple solutions, our evolutionary algorithm is different from ordinary ones in some aspects. First of all, we use an adaptive fitness function procedure which helps searching process to avoid returning back to already detected solutions and gives it opportunity to explore other regions. Secondly, since we know the global minimum value of the problem, termination conditions used in the algorithm are different from those used in ordinary evolutionary algorithms.

Adaptive Fitness Function. The purpose of this procedure is to help the searching process avoid lingering around already detected solutions. So once a global solution, or local solution, or some unpromising trial solution is detected during the search, the fitness function will be modified around this point. Depending on the type of the point, we use different modifications.

Suppose we have a point $\bar{x}$ on which adaptation is to be made. Let f_c be the current fitness function.

1. If $\bar{x}$ is a non-global local optimal solution, then the fitness function will be modified by means of the so-called tunneling function and the new fitness function will be given by:

$$\bar{f}_t(x,\bar{x}) := f_c(x) \cdot \exp\left(\frac{1}{\varepsilon_t + \frac{1}{\rho_t^2}\|x-\bar{x}\|^2}\right), \tag{6}$$

   where ε_t, ρ_t are parameters that control the degree and the range of modification.
2. If $\bar{x}$ is a global optimal solution, then the fitness function will be modified by means of the so-called hump-tunneling function, and the new fitness function will be given by:

$$\begin{aligned}\bar{f}_{ht}(x,\bar{x}) &:= f_h(x,\bar{x}) \cdot \exp\left(\frac{1}{\varepsilon_t + \frac{1}{\rho_t^2}\|x-\bar{x}\|^2}\right) \\ &= \left(f_c(x) + \alpha_h \max\left\{0, 1 - \frac{1}{\bar{\rho}_h^2}\|x-\bar{x}\|^2\right\}\right) \cdot \exp\left(\frac{1}{\varepsilon_t + \frac{1}{\rho_t^2}\|x-\bar{x}\|^2}\right),\end{aligned} \tag{7}$$

where ε_t, ρ_t, $\varepsilon_h, and \rho_h$ are parameters that control the degree and the range of modification. The idea underlying the tunneling and the hump-tunneling function modifications is explained in detail in the recent work [7]. Denote the adapting procedure on $\bar{x}$ with f_c as **AFF**$(f_c,\bar{x})$.

Termination. To terminate our EA, we use the following three different criteria:

- The number of function evaluations exceeds the pre-defined limit.
- The number of detected global solutions exceeds the pre-defined number.

– Let N_s be a pre-specified positive integer. If among the most recent N_s points of modification, there were not new global solutions, then we terminate the main algorithm.

The main loop of the proposed algorithm is stated as follows:

Algorithm

1. Initialization. Choose parameters $M, m, l_s, \bar{N}$ and $\beta \in (0,1), \varepsilon > 0$. Generate the population set $P := x^1, x^2, \ldots, x^M$ by using some Diversity Generation Method. Let the set of modification points $S := \emptyset$. Define the current fitness function as

$$f_c(x) := f(x).$$

Sort the elements in P in ascending order of their current fitness function values, i.e.,

$$f_c(x^1) \leq f_c(x^2) \leq \cdots \leq f_c(x^M).$$

Set the generation counters $t := 1$ and $s := 1$.

2. Parents Selection. Generate a parents pool

$$P' := \{(x^i, x^j) | x^i, x^j \in P,\ x^i \neq x^j\}.$$

3. Crossover and Mutation. Select a pair $(p^1, p^2) \in P'$ and generate a new pair by

$$(c^1, c^2) \longleftarrow \text{Crossover}[(p^1, p^2)] + \text{Mutation}.$$

4. Survival Selection. If the child c^1 or c^2 has a lower fitness function value than some element in the population set P, then let it survive in the population set and discard the worst member of P. If $P' = \emptyset$, then let $N := \min\{s, \bar{N}\}$ and

$$B := \{b^1, b^2, \ldots, b^N\} \leftarrow \{x^1, b^1, \ldots, b^{(N-1)}\},\ s := s+1$$

and go to step 5; otherwise go to step 3.

5. Intensification. If, during the last $\bar{N}$ generations of evolution, the fitness function has not been modified and the best point in the population set has not been improved enough, i.e.,

$$s \geq \bar{N} \text{ and } \left| f_c(b^{\bar{N}}) - f_c(b^1) \right| \leq \beta(1 + |f_c(b^1)|),$$

then choose $x^1, x^2, \ldots, x^m \in P$ and for each $x^i,\ i = 1, 2, \ldots, m$, perform the following procedure:

$$\bar{x}^i \longleftarrow \text{Local Search } (f(x), x^i, l_s).$$

If x^i is an unpromising trial point, then construct a new fitness function by

$$f_c(x) := \mathbf{AFF}(f_c, x^i).$$

Otherwise, $P := \{P \cup \bar{x}^i\} \backslash \{x^i\}$. If the fitness function is modified at least once during the above procedure, then set $s := 1$. Go to step 6.

6. Solutions and Adaptations. If $x^1 \in P$ is a global solution, i.e.,

$$f_c(x^1) < \varepsilon$$

or, it is regarded as a local solution, i.e.,

$$s \geq \bar{N} \text{ and } \left| f_c(b^{\bar{N}}) - f_c(x^1) \right| \leq \beta(1 + |f_c(x^1)|),$$

then construct a new fitness function by

$$f_c(x) := \mathbf{AFF}(f_c, x^1)$$

and set $s := 1$. Otherwise, let $B := \{b^1, b^2, \ldots, b^{\bar{N}}\} \leftarrow \{x^1, b^1, \ldots, b^{(\bar{N}-1)}\}$. Proceed to step 7 with $(f_c(x), P)$.

7. Stopping Condition. If one of the stopping conditions holds, then terminate the algorithm and refine the global solutions in S by some local search method. Otherwise, set $t := t + 1$ and go to step 2.

5 Numerical Experiments

To show the practical effectiveness of our approach, we have chosen the five test problems from literatures for the numerical experiments. Programming code was developed in MATLAB, and MATLAB command "fmincon" is used for local search in the evolutionary algorithm. Parameters' choice used in the algorithm is shown in Table 1.

For one of the stopping condition, the maximum number of global solutions to be detected, we have chosen ten. In other words, when the number of detected global solutions reaches ten, we stopped the algorithm assuming that the problem has infinitely many solutions. The numerical results for the evolutionary algorithm for solving GNEP are presented in Table 2.

In this table, the columns represent: n, dimension of the problem; $K_{\min}, K_{\max}, K_{\text{av}}$, the minimum, maximum, average numbers of detected global solutions; N_{gen}, the number of generation in the evolutionary search; N_{loc}, the number of the local search steps used; and NF, the number of function evolutions. As we can see in Table 2, our approach finds multiple generalized Nash equilibria in acceptable numbers of function evaluations, and all five test problems have at least ten solutions.

Table 1 Parameter settings

Parameters	Definition	Value
M	Number of elements in the population	10
a, b	Parameters defining the linearized D gap function	0.5, 1.0
l_N	Number of best points for which local search is used	1
l_s	Maximum number of steps per local search	8
N_k, η	Parameters controlling local search in EA	4, 0.995
ε_t, ρ_t	Tunneling parameters	0.1, 2
ε_{EA}	Tolerance for the objective	10^{-8}
N_g	The maximum number of global solutions to be detected	10
$\text{NF}_{\max}$	Maximum number of function evaluations	10,000

Table 2 Numerical experiments for the evolutionary algorithm with adaptive fitness functions

Problem	n	$K_{\min}$	K_{av}	$K_{\max}$	N_{gen}	N_{loc}	NF
Facchinei	2	10	10	10	53	6	3,517
Harker	2	10	10	10	53	58	4,249
Nabetani1	2	10	10	10	48	8	3,269
Nabetani2	2	10	10	10	46	6	3,032
RBP	3	10	10	10	54	18	3,782

6 Conclusion

In this chapter, we have considered the possible heuristic global optimization approach for finding generalized Nash equilibria. With help of adaptive fitness function, our evolutionary algorithm searches multiple solutions of the problem at the same time. Numerical experiments show the practical effectiveness of our method. In future study, it will be interesting to consider parallel computing for multiple solutions of the problem and adaptive fitness function techniques connected solutions of the problem.

Test Problems

- Test problem 1 (Facchinei). This test problem is taken from [2] and has infinitely many solutions given by $(\alpha, 1-\alpha), \quad \forall \alpha \in [\frac{1}{2}; 1]$. In this problem, the players solve the following optimization problems for a GNE:

$$P_1(x_2) : \text{minimize } (x_1 - 1)^2$$
$$\text{subject to } x_1 + x_2 \leq 1.$$
$$P_2(x_1) : \text{minimize } \left(x_2 - \frac{1}{2}\right)^2$$
$$\text{subject to } x_1 + x_2 \leq 1.$$

- Test problem 2 (Harker). This problem is taken from [5]. There are two players and they solve the following problems:

$$\begin{aligned} P_1(x_2) &: \text{minimize } x_1^2 + \frac{8}{3}x_1x_2 - 34x_1 \\ &\quad \text{subject to } x_1 + x_2 \leq 15,\ \ 0 \leq x_1 \leq 10. \\ P_2(x_1) &: \text{minimize } x_2^2 + \frac{5}{4}x_1x_2 - \frac{97}{4}x_2 \\ &\quad \text{subject to } x_1 + x_2 \leq 15,\ \ 0 \leq x_2 \leq 10. \end{aligned}$$

  It has infinitely many solutions given by $(\alpha, 15-\alpha),\ \ \forall\alpha \in [9;10]$.
- Test problem 3 (Nabetani1). This problem is taken from [8] and has infinitely many solutions given by $(\alpha, 1-\alpha),\ \ \forall\alpha \in [0;\frac{2}{3}]$.

$$\begin{aligned} P_1(x_2) &: \text{minimize } x_1^2 - x_1x_2 - x_1 \\ &\quad \text{subject to } x_1 + x_2 \leq 1,\ \ x_1 \geq 0. \\ P_2(x_1) &: \text{minimize } x_2^2 - \frac{1}{2}x_1x_2 - 2x_2 \\ &\quad \text{subject to } x_1 + x_2 \leq 1,\ \ x_2 \geq 0. \end{aligned}$$

- Test problem 4 (Nabetani2). This problem is taken from [8] and has infinitely many solutions given by $(\alpha, \sqrt{1-\alpha^2}),\ \ \forall\alpha \in [0;\frac{4}{5}]$.

$$\begin{aligned} P_1(x_2) &: \text{minimize } x_1^2 - x_1x_2 - x_1 \\ &\quad \text{subject to } x_1^2 + x_2^2 \leq 1,\ \ x_1 \geq 0. \\ P_2(x_1) &: \text{minimize } x_2^2 - \frac{1}{2}x_1x_2 - 2x_2 \\ &\quad \text{subject to } x_1^2 + x_2^2 \leq 1,\ \ x_2 \geq 0. \end{aligned}$$

- Test problem 5 (River Basin Pollution—RBP). In this problem [5], we consider the 3-person river basin pollution game, where the problem of player $\nu \in \{1,2,3\}$ is defined by

$$\begin{aligned} P_\nu(x_\nu) &: \text{minimize } \alpha_\nu x_\nu + \beta(x_1 + x_2 + x_3) - \gamma_\nu)x_\nu \\ &\quad \text{subject to } x_\nu \geq 0 \\ &\qquad 3.25x1 + 1.25x2 + 4.125x3 \leq 100, \\ &\qquad 2.2915x1 + 1.5625x2 + 2.8125x3 \leq 100 \end{aligned}$$

  with the parameters $\alpha_1 = 0.01,\ \alpha_2 = 0.05,\ \alpha_3 = 0.01,\ \beta = 0.01,\ \gamma_1 = 2.9,\ \gamma_2 = 2.88,$ *and* $\gamma_3 = 2.85$. It is also known that the problem has infinitely many solutions.

Acknowledgements This research was supported in part by a Grant-in-Aid for Scientific Research from Japan Society for the Promotion of Science.

References

1. Facchinei, F., Fischer, A. and Piccialli, V. (2007), On generalized Nash games and VIs, *Operation Research Letters*, 35, 159–164.
2. Facchinei, F., Fischer, A. and Piccialli, V. (2008), Generalized Nash equilibrium problems and Newton methods, *Mathematical Programming*, Volume 117, Numbers 1-2, 163–194.
3. Facchinei, F. and Kanzow, C. (2007), Generalized Nash equilibrium problems, *A Quarterly Journal of Operations Research*, 5, 173–210.
4. Fukushima, M. (2007), A class of gap functions for quasi-variational inequality problems, *Journal of Industrial and Management Optimization*, 3, 165–171.
5. Harker, P.T. (1991), Generalized Nash games and quasi-variational inequalities, *European Journal of Operational Research*, 54, 81–94.
6. Heusinger, A. and Kanzow, C. (2009), Optimization reformulations of the generalized Nash equilibrium problem using Nikaido-Isoda-type functions, *Computational Optimization and Applications*, 43, 353–377.
7. Majig, M., Hedar, A.R. and Fukushima, M. (2007), Hybrid evolutionary algorithm for solving general variational inequalities, *Journal of Global Optimization*, 38, 637–651.
8. Nabetani, K., Tseng, P. and Fukushima, M. (2011), Parametrized variational inequality approaches to generalized Nash equilibrium problems with shared constraints, *Computational Optimization and Applications*, 48, 423–452.
9. Pang, J.-S. and Fukushima, M. (2005), Quasi-variational inequalities, generalized Nash equilibria, and multi-leader-follower games, *Computational Management Science*, 2, 21–56.

Scalar and Vector Optimization with Composed Objective Functions and Constraints

Nicole Lorenz and Gert Wanka

Abstract In this chapter we consider scalar and vector optimization problems with objective functions being the composition of a convex function and a linear mapping and cone and geometric constraints. By means of duality theory we derive dual problems and formulate weak, strong, and converse duality theorems for the scalar and vector optimization problems with the help of some generalized interior point regularity conditions and consider optimality conditions for a certain scalar problem.

Key words Duality • Interior point regularity condition • Optimality conditions

1 Introduction

To a certain multiobjective optimization problem one can attach a scalar one whose optimal solution leads to solutions of the original problem. Different scalarization methods, especially *linear scalarization*, can be used to this purpose. Weak and strong duality results and required regularity conditions of the scalar and vector problem are associated with them. In the book of Boţ, Grad, and Wanka (cf. [1]), a broad variety of scalar and vector optimization problems is considered. Related to the investigations within that book we consider here some different scalar and vector optimization problems associated with each other and show how the duals, weak and strong duality, and some regularity conditions can be derived.

N. Lorenz
Faculty of Mathematics, Chemnitz University of Technology, D-09107 Chemnitz, Germany
e-mail: nicole.lorenz@mathematik.tu-chemnitz.de.

G. Wanka (✉)
Faculty of Mathematics, Chemnitz University of Technology, D-09107 Chemnitz, Germany
e-mail: gert.wanka@mathematik.tu-chemnitz.de

A. Chinchuluun et al. (eds.), *Optimization, Simulation, and Control*, Springer Optimization and Its Applications 76, DOI 10.1007/978-1-4614-5131-0_8,

We assume $\mathcal{X}, \mathcal{Y}, \mathcal{V}$, and $\mathcal{Z}$ to be Hausdorff locally convex spaces, whereas in order to guarantee strong duality some of the regularity conditions contain the assumption that we have Fréchet spaces.

We consider the scalar optimization problem

$$(PS^{\Sigma}) \qquad \inf_{x \in \mathcal{A}} \left\{ \sum_{i=1}^{m} \lambda_i f_i(Ax) \right\}, \quad \mathcal{A} = \{x \in S : g_i(x) \leq 0, i = 1, \ldots, k\},$$

taking proper and convex functions $f_i : \mathcal{Y} \to \overline{\mathbb{R}} := \mathbb{R} \cup \{\pm\infty\}, i = 1, \ldots, m$, weighted by positive constants $\lambda_i, i = 1, \ldots, m$, further $g = (g_1, \ldots, g_k)^T : \mathcal{X} \to \mathbb{R}^k$, where $g_i, i = 1, \ldots, k$, is assumed to be convex, $S \subseteq \mathcal{X}$ is a non-empty convex set and $A \in \mathcal{L}(\mathcal{X}, \mathcal{Y})$, i.e., a linear continuous operator mapping from $\mathcal{X}$ to $\mathcal{Y}$. Another problem is the scalar one

$$(PS) \qquad \inf_{x \in \mathcal{A}} f(Ax), \quad \mathcal{A} = \{x \in S : g(x) \in -C\},$$

which is related to the first one. Here we use the proper and convex function $f : \mathcal{Y} \to \overline{\mathbb{R}}$ and the C-convex function $g : \mathcal{X} \to \mathcal{Z}$ and a nontrivial convex cone $C \subseteq \mathcal{Z}$.

Further we consider two vector optimization problems to which scalar ones may be attached, whose dual problems are used to formulate duals to the vector optimization problems. This can be seen in the following sections.

For the space $\mathcal{X}$ partially ordered by the convex cone K we denote by $\mathcal{X}^{\bullet}$ the space to which a greatest element $+\infty_K$ is attached (cf. [1]).

We consider the following vector optimization problem:

$$(PV^m) \qquad \operatorname{Min}_{x \in \mathcal{A}} (f_1(Ax), \ldots, f_m(Ax))^T, \quad \mathcal{A} = \{x \in S : g_i(x) \leq 0, i = 1, \ldots, k\}.$$

Here we assume $f = (f_1, \ldots, f_m)^T : \mathcal{Y} \to \mathbb{R}^{m\bullet}$ to be a proper function with convex functions $f_i, i = 1, \ldots, m$, and $g_i : \mathcal{X} \to \mathbb{R}, i = 1, \ldots, k$, to be convex. Further we have $S \subseteq \mathcal{X}$. The problem (PS^{Σ}) arises by linear scalarization of (PV^m). Further we consider the following vector optimization problem related to the above one:

$$(PV) \qquad \operatorname{Min}_{x \in \mathcal{A}} f(Ax), \qquad \mathcal{A} = \{x \in S : g(x) \in -C\}.$$

Here $f : \mathcal{Y} \to \mathcal{V}^{\bullet}$ is a proper and K-convex function and $g : \mathcal{X} \to \mathcal{Z}$ is a C-convex function, using the nontrivial pointed convex cone $K \subseteq \mathcal{V}$ and the nontrivial convex cone $C \subseteq \mathcal{Z}$.

The conjugate dual problems to the scalar and vector optimization problem arise as a combination of the classical Fenchel and Lagrange duality. It is the so-called Fenchel-Lagrange duality introduced by Boţ and Wanka (cf. [2, 3, 10]).

For the primal-dual pair one has *weak duality*, where the values of the dual objective function at its feasible set do not surpass the values of the primal objective function at its feasible set. Further, for scalar optimization problems, we have *strong duality* if there exists a solution of the dual problem such that the objective values coincide, whereas for vectorial ones in case of *strong duality*,

we assume the existence of solutions of the primal and dual problem such that the objective values coincide, and for *converse duality* we start with a solution of the dual and prove the existence of a primal solution such that the objective values coincide.

In order to have strong and converse duality we have to formulate regularity conditions. Since the classical Slater constraint qualifications (cf. [5,9]) are often not fulfilled, we will present generalized interior point regularity conditions. Conditions for some dual problems were given by Boţ, Grad, and Wanka (cf. [1]). Thus we modify these conditions and resulting theorems to adopt them to the problems we study in this chapter. Further, in [11] also some vector optimization problems and their duals having a composition in the objective function and the constraints were considered.

The central aim of this chapter is to give an overview of special scalar and vector optimization problems. In addition, we point out the connections between them as well as the arising interior point regularity conditions.

This chapter is organized as follows. In the following section we introduce some definitions and notations from the convex analysis we use within this chapter. In Sect. 3 we consider two general scalar optimization problems, calculate the dual ones, give regularity conditions, further formulate weak and strong duality theorems, and give optimality conditions for one of them. Moreover, we consider two vector optimization problems and also calculate the dual ones and formulate weak, strong, and converse duality theorems, respectively.

2 Notations and Preliminaries

Let $\mathcal{X}$ be a Hausdorff locally convex space and $\mathcal{X}^*$ its topological dual space which we endow with the weak* topology $w(\mathcal{X}^*,\mathcal{X})$. We denote by $\langle x^*,x\rangle := x^*(x)$ the value of the linear continuous functional $x^* \in \mathcal{X}^*$ at $x \in \mathcal{X}$. For $\mathcal{X} = \mathbb{R}^n$ we have $\mathcal{X} = \mathcal{X}^*$ and for $x = (x_1,\ldots,x_n)^T \in \mathbb{R}^n, x^* = (x_1^*,\ldots,x_n^*)^T \in \mathbb{R}^n$ it holds $\langle x^*,x\rangle = (x^*)^T x = \sum_{i=1}^n x_i^* x_i$.

For $f : \mathcal{X} \to \mathcal{V}$ and $v^* \in \mathcal{V}^*$ we define the function $v^* f : \mathcal{X} \to \mathbb{R}$ by $v^* f(x) := \langle v^*, f(x)\rangle$ for $x \in \mathcal{X}$, where $\mathcal{V}$ is another Hausdorff locally convex space and $\mathcal{V}^*$ its topological dual space.

The zero vector will be denoted by $\mathbf{0}$, whereas the space we talk about will be clear from the context. By e we denote the vector $(1,\ldots,1)^T$.

For a set $D \subseteq \mathcal{X}$ the *indicator function* $\delta_D : \mathcal{X} \to \overline{\mathbb{R}}$ is defined by

$$\delta_D(x) := \begin{cases} 0, & x \in D, \\ +\infty, & \text{otherwise.} \end{cases}$$

When $D \subseteq \mathcal{X}$ is non-empty and $f : \mathcal{X} \to \overline{\mathbb{R}}$ we denote by $f_D^* : \mathcal{X}^* \to \overline{\mathbb{R}}$ the function defined by

$$f_D^*(x^*) = (f + \delta_D)^*(x^*) = \sup_{x \in D}\{\langle x^*,x\rangle - f(x)\}.$$

One can see that for $D = \mathcal{X}$, f_D^* becomes the *(Fenchel-Moreau) conjugate function of* f which we denote by f^*. We have the so-called *Young* or *Young-Fenchel inequality*:

$$f(x) + f^*(x^*) \geq \langle x^*, x \rangle, \qquad \forall x \in \mathcal{X}, \forall x^* \in \mathcal{X}^*. \tag{1}$$

The *support function* $\sigma_D : \mathcal{X}^* \to \overline{\mathbb{R}}$ is defined by $\sigma_D(x^*) = \sup_{x \in D} \langle x^*, x \rangle$ and it holds $\sigma_D = \delta_D^*$.

Let $K \subseteq \mathcal{X}$ be a nontrivial convex cone. The cone K induces on $\mathcal{X}$ a partial ordering $\leqq_K$ defined for $x, y \in \mathcal{X}$ by $x \leqq_K y \Leftrightarrow y - x \in K$. Moreover, let us define $x \leq_K y$ if and only if $x \leqq_K y$ and $x \neq y$. The *dual cone* $K^* \subseteq \mathcal{X}^*$ and the *quasi interior of the dual cone* of K^*, respectively, are defined by

$$K^* := \{x^* \in \mathcal{X}^* : \ \langle x^*, x \rangle \geq 0, \ \forall x \in K\},$$
$$K^{*0} := \{x^* \in K^* : \ \langle x^*, x \rangle > 0, \ \forall x \in K \setminus \{\mathbf{0}\}\}.$$

A convex cone K is said to be *pointed* if its *linearity space* $l(K) = K \cap (-K)$ is the set $\{\mathbf{0}\}$. For a set $U \subseteq \mathcal{X}$ the *conic hull* is

$$\operatorname{cone}(U) = \bigcup_{\lambda \geq 0} \lambda U = \{\lambda u : u \in U, \lambda \geq 0\}.$$

If we assume that $\mathcal{X}$ is partially ordered by the convex cone K, we denote by $+\infty_K$ the *greatest element with respect to* $\leqq_K$ and by $\mathcal{X}^\bullet$ the set $\mathcal{X} \cup \{+\infty_K\}$. For any $x \in \mathcal{X}^\bullet$ it holds $x \leqq_K +\infty_K$ and $x \leq_K +\infty_K$ for any $x \in \mathcal{X}$. On $\mathcal{X}^\bullet$ we consider the following operations and conventions (cf. [1]): $x + (+\infty_K) = (+\infty_K) + x := +\infty_K, \forall x \in \mathcal{X} \cup \{+\infty_K\}, \lambda \cdot (+\infty_K) := +\infty_K, \forall \lambda \in (0, +\infty], 0 \cdot (+\infty_K) := +\infty_K$. Note that we define $+\infty_{\mathbb{R}_+} =: +\infty$ and further $\leqq_{\mathbb{R}_+} =: \leq$ and $\leq_{\mathbb{R}_+} =: <$.

By $B_{\mathcal{X}}(x, r)$ we denote the *open ball with radius* $r > 0$ *and center* x in $\mathcal{X}$ defined by $B_{\mathcal{X}}(x, r) = \{y \in \mathcal{X} : d(x, y) < r\}$, where $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is the metric induced by the topology in $\mathcal{X}$ if $\mathcal{X}$ is metrizable.

The prefixes int, ri, icr, sqri, and core are used for the *interior*, the *relative interior*, the *relative algebraic interior (or intrinsic core)*, the *strong quasi relative interior* and the *algebraic interior* or *core* of a set $U \subseteq \mathcal{X}$, respectively, where

$$\operatorname{core}(U) = \{x \in \mathcal{X} : \forall y \in \mathcal{X}, \exists \delta > 0 \text{ such that } \forall \lambda \in [0, \delta] : x + \lambda y \in U\},$$
$$\operatorname{ri}(U) = \{x \in \operatorname{aff}(U) : \exists \varepsilon > 0 : B_{\mathcal{X}}(x, \varepsilon) \cap \operatorname{aff}(U) \subseteq U\},$$
$$\operatorname{icr}(U) = \{x \in \mathcal{X} : \forall y \in \operatorname{aff}(U - U), \exists \delta > 0 \text{ s.t. } \forall \lambda \in [0, \delta] : x + \lambda y \in U\},$$
$$\operatorname{sqri}(U) = \begin{cases} \operatorname{icr}(U), & \text{if } \operatorname{aff}(U) \text{ is a closed set,} \\ \emptyset, & \text{otherwise,} \end{cases}$$

and in case of having a convex set $U \subseteq \mathcal{X}$ we have

$$\operatorname{core}(U) = \{x \in U : \operatorname{cone}(U - x) = \mathcal{X}\},$$
$$\operatorname{sqri}(U) = \{x \in U : \operatorname{cone}(U - x) \text{ is a closed linear subspace}\}.$$

It holds $\operatorname{core}(U) \subseteq \operatorname{sqri}(U)$ and $\operatorname{aff}(U)$ is the *affine hull* of the set U,

$$\operatorname{aff}(U) = \left\{ \sum_{i=1}^{n} \lambda_i x_i : n \in \mathbb{N}, x_i \in U, \lambda_i \in \mathbb{R}, \sum_{i=1}^{n} \lambda_i = 1, i = 1, \ldots, n \right\}.$$

We assume $\mathcal{V}$ to be a Hausdorff locally convex space partially ordered by the nontrivial convex cone $C \subseteq \mathcal{V}$.

The *effective domain* of a function $f : \mathcal{X} \to \overline{\mathbb{R}}$ is $\operatorname{dom}(f) = \{x \in \mathcal{X} : f(x) < +\infty\}$, and we will say that f is *proper* if $\operatorname{dom}(f) \neq \emptyset$ and $f(x) > -\infty$, $\forall x \in \mathcal{X}$. The *domain* of a vector function $f : \mathcal{X} \to \mathcal{V}^\bullet$ is $\operatorname{dom}(f) = \{x \in \mathcal{X} : f(x) \neq +\infty_C\}$. When $\operatorname{dom}(f) \neq \emptyset$, the vector function f is called *proper*.

While a proper function $f : \mathcal{X} \to \overline{\mathbb{R}}$ is called *convex* if for all $x, y \in \mathcal{X}$ and all $\lambda \in [0,1]$ it holds $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$, a vector function $f : \mathcal{X} \to \mathcal{V}^\bullet$ is said to be *C-convex* if for all $x, y \in \mathcal{X}$ and all $\lambda \in [0,1]$ it holds $f(\lambda x + (1-\lambda)y) \leqq_C \lambda f(x) + (1-\lambda) f(y)$ (cf. [1]).

A function $f : \mathcal{X} \to \overline{\mathbb{R}}$ is called *lower semicontinuous at* $\bar{x} \in \mathcal{X}$ if $\liminf_{x \to \bar{x}} f(x) \geq f(\bar{x})$, while a function $f : \mathcal{X} \to \mathcal{V}^\bullet$ is *star C-lower semicontinuous at* $\bar{x} \in \mathcal{X}$ if $(v^* f)$ is lower semicontinuous at $\bar{x}$ for all $v^* \in C^*$. The latter notion was first given in [6].

For $f : \mathcal{X} \to \overline{\mathbb{R}}$ and $a \in \mathbb{R}$ we call $\operatorname{lev}_a(f) := \{x \in \mathcal{X} : f(x) \leq a\}$ the *level set* of f at a.

By $\mathcal{L}(\mathcal{X}, \mathcal{Y})$ we denote the set of linear continuous operators mapping from $\mathcal{X}$ into $\mathcal{Y}$. For $A \in \mathcal{L}(\mathcal{X}, \mathcal{Y})$ one can define the *adjoint operator*, $A^* : \mathcal{Y}^* \to \mathcal{X}^*$ by

$$\langle A^* y^*, x \rangle = \langle y^*, Ax \rangle, \quad \forall y^* \in \mathcal{Y}^*, x \in \mathcal{X}.$$

In the following we write min and max instead of inf and sup if we want to express that the infimum/supremum of a scalar optimization problem is attained.

Definition 1 (Infimal convolution). For the proper functions $f_1, \ldots, f_k : \mathcal{X} \to \overline{\mathbb{R}}$, the function $f_1 \square \cdots \square f_k : \mathcal{X} \to \overline{\mathbb{R}}$ defined by

$$(f_1 \square \cdots \square f_k)(p) = \inf \left\{ \sum_{i=1}^{k} f_i(p_i) : \sum_{i=1}^{k} p_i = p \right\}$$

is called the *infimal convolution* of $f_i, i = 1, \ldots, k$.

In order to state a theorem for the infimal convolution of conjugate functions, we introduce additionally to a classical condition (RC_1^Σ) the following generalized interior point regularity conditions $(RC_i^\Sigma), i \in \{2,3,4\}$:

$$(RC_1^\Sigma) \quad \left| \begin{array}{l} \exists x' \in \cap_{i=1}^k \mathrm{dom}(f_i) \text{ such that a number of } k-1 \text{ functions} \\ \text{of the functions } f_i, i = 1,\ldots,k, \text{ are continuous at } x', \end{array} \right. \tag{2}$$

$$(RC_2^\Sigma) \quad \left| \begin{array}{l} \mathcal{X} \text{ is Fréchet space, } f_i \text{ is lower semicontinuous, } i = 1,\ldots,k, \\ \text{and } \mathbf{0} \in \mathrm{sqri}\left(\Pi_{i=1}^k \mathrm{dom}(f_i) - \Delta_{\mathcal{X}^k}\right), \end{array} \right.$$

$$(RC_3^\Sigma) \quad \left| \begin{array}{l} \mathcal{X} \text{ is Fréchet space, } f_i \text{ is lower semicontinuous, } i = 1,\ldots,k, \text{ and} \\ \mathbf{0} \in \mathrm{core}\left(\Pi_{i=1}^k \mathrm{dom}(f_i) - \Delta_{\mathcal{X}^k}\right), \end{array} \right.$$

$$(RC_4^\Sigma) \quad \left| \begin{array}{l} \mathcal{X} \text{ is Fréchet space, } f_i \text{ is lower semicontinuous, } i = 1,\ldots,k, \text{ and} \\ \mathbf{0} \in \mathrm{int}\left(\Pi_{i=1}^k \mathrm{dom}(f_i) - \Delta_{\mathcal{X}^k}\right), \end{array} \right. \tag{3}$$

where for a set $M \subseteq \mathcal{X}$ we define $\Delta_{M^k} := \{(x,\ldots,x) \in \mathcal{X}^k : x \in M\}$. The following theorem holds (cf. [1, Theorem 3.5.8]):

Theorem 1. *Let $f_1,\ldots,f_k : \mathcal{X} \to \overline{\mathbb{R}}$ be proper and convex functions. If one of the regularity conditions $(RC_i^\Sigma), i \in \{1,2,3,4\}$, is fulfilled, then it holds for all $p \in \mathcal{X}^*$*

$$\left(\sum_{i=1}^k f_i\right)^*(p) = (f_1^* \square \cdots \square f_k^*)(p) = \min\left\{\sum_{i=1}^k f_i^*(p_i) : \sum_{i=1}^k p_i = p\right\}. \tag{4}$$

Remark 1. For $\mathcal{X} = \mathbb{R}^n$ formula (4) holds if $f_i, i = 1,\ldots,k$, is proper and convex and $\cap_{i=1}^k \mathrm{ri}(\mathrm{dom}(f_i)) \neq \emptyset$, i.e., we do not need one of the conditions $(RC_i^\Sigma), i \in \{1,2,3,4\}$ (cf. [8, Theorem 20.1]).

The function $f : \mathcal{X} \to \mathcal{V}^\bullet$ is called *C-epi closed* if its *C-epigraph*, namely $\mathrm{epi}_C f = \{(x,y) \in \mathcal{X} \times \mathcal{V} : f(x) \leqq_C y\}$, is a closed set (cf. [7]). For a real valued function $f : \mathcal{X} \to \overline{\mathbb{R}}$ and $C = \mathbb{R}_+$ we have $\mathrm{epi} f = \mathrm{epi}_C f$ and the following theorem holds (cf. [1, Theorem 2.2.9]):

Theorem 2. *Let the function $f : \mathcal{X} \to \overline{\mathbb{R}}$ be given. Then the following statements are equivalent:*

(i) *f is lower semicontinuous.*
(ii) $\mathrm{epi} f$ *is closed.*
(iii) *The level set* $\mathrm{lev}_a(f) = \{x \in \mathcal{X} : f(x) \leq a\}$ *is closed for all* $a \in \mathbb{R}$.

3 Some Dual Optimization Problems

In this section we consider the optimization problems (PS) and (PV). These are related problems, the first one a scalar, the latter one a vectorial, having as objective function a composition of a convex (vector) function and a linear continuous operator and cone and geometric constraints. For these we formulate dual problems and state weak, strong, and converse duality theorems under some classical and generalized interior point regularity conditions. Further, we consider two problems (PS^{Σ}) and (PV^{m}) related to the above ones and derive the same things.

For the whole section we assume that $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$, and $\mathcal{V}$ are Hausdorff locally convex spaces; $\mathcal{Z}$ and $\mathcal{V}$ are assumed to be partially ordered by the nontrivial convex cone $C \subseteq \mathcal{Z}$ and the nontrivial pointed convex cone $K \subseteq \mathcal{V}$, respectively. Further, let $S \subseteq \mathcal{X}$ be a nonempty convex set and $A \in \mathcal{L}(\mathcal{X}, \mathcal{Y})$.

3.1 The Scalar Optimization Problem (PS)

In this first section we consider a general scalar optimization problem. Therefore we assume the function $f : \mathcal{Y} \to \overline{\mathbb{R}}$ to be proper and convex and the vector function $g : \mathcal{X} \to \mathcal{Z}$ to be C-convex, fulfilling $A^{-1}(\mathrm{dom}(f)) \cap g^{-1}(-C) \cap S \neq \emptyset$. Consider the following primal scalar optimization problem:

$$(PS) \quad \inf_{x \in \mathcal{A}} f(Ax), \quad \mathcal{A} = \{x \in S : g(x) \in -C\}.$$

We derive here a dual problem which is called the Fenchel-Lagrange dual problem to (PS). For this purpose we consider the perturbation function $\Phi_{FL} : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \overline{\mathbb{R}}$, given by

$$\Phi_{FL}(x, y, z) = \begin{cases} f(Ax + y), & x \in S, g(x) \in z - C, \\ +\infty, & \text{otherwise}, \end{cases} \tag{5}$$

where $\mathcal{X}$ is the space of feasible variables and $\mathcal{Y}$ and $\mathcal{Z}$ are the spaces of perturbation variables. First we calculate to Φ_{FL} the conjugate function $(\Phi_{FL})^* : \mathcal{X}^* \times \mathcal{Y}^* \times \mathcal{Z}^* \to \overline{\mathbb{R}}$:

$$\begin{aligned} &(\Phi_{FL})^*(x^*, y^*, z^*) \\ &\quad = \sup_{\substack{(x,y,z) \in S \times \mathcal{Y} \times \mathcal{Z} \\ g(x) - z \in -C}} \{\langle x^*, x \rangle + \langle y^*, y \rangle + \langle z^*, z \rangle - f(Ax + y)\} \qquad (6) \\ &\quad = \sup_{(x,r,s) \in S \times \mathcal{Y} \times -C} \{\langle x^*, x \rangle + \langle y^*, r - Ax \rangle + \langle z^*, g(x) - s \rangle - f(r)\} \\ &\quad = \delta_{-C^*}(z^*) + \sup_{(x,r) \in S \times \mathcal{Y}} \{\langle x^* - A^* y^*, x \rangle + \langle y^*, r \rangle + (z^* g)(x) - f(r)\} \\ &\quad = \delta_{-C^*}(z^*) + \sup_{x \in S} \{\langle x^* - A^* y^*, x \rangle + (z^* g)(x)\} + \sup_{r \in \mathcal{Y}} \{\langle y^*, r \rangle - f(r)\}. \qquad (7) \end{aligned}$$

It follows

$$-(\Phi_{FL})^*(0,y^*,z^*) = -\delta_{-C^*}(z^*) - (-z^*g)^*_S(-A^*y^*) - f^*(y^*).$$

The dual problem becomes (cf. [1] and take $z^* := -z^*$):

$$\begin{aligned}(DS_{FL}) \quad & \sup_{(y^*,z^*)\in\mathcal{Y}^*\times\mathcal{Z}^*} (-(\Phi_{FL})^*(0,y^*,z^*)) \\ &= \sup_{(y^*,z^*)\in\mathcal{Y}^*\times\mathcal{Z}^*} (-\delta_{-C^*}(z^*) - (-z^*g)^*_S(-A^*y^*) - f^*(y^*)) \\ &= \sup_{(y^*,z^*)\in\mathcal{Y}^*\times C^*} (-(z^*g)^*_S(-A^*y^*) - f^*(y^*)). \end{aligned} \tag{8}$$

We denote by $v(PS)$ and $v(DS_{FL})$ the *optimal objective value* of (PS) and (DS_{FL}), respectively. Then weak duality holds by construction (cf. [1]), i.e., $v(PS) \geq v(DS_{FL})$. In order to have strong duality we introduce some regularity conditions.

For a general optimization problem given by

$$(P) \quad \inf_{x\in\mathcal{X}} \Phi(x,0),$$

depending on the perturbation function $\Phi : \mathcal{X}\times\mathcal{Y} \to \overline{\mathbb{R}}$, we introduce the following so-called *generalized interior point regularity conditions*, where we assume that Φ is a proper and convex function fulfilling $0 \in \mathrm{Pr}_{\mathcal{Y}}(\mathrm{dom}(\Phi))$ and $\mathrm{Pr}_{\mathcal{Y}} : \mathcal{X}\times\mathcal{Y} \to \mathcal{Y}$, defined for $(x,y) \in \mathcal{X}\times\mathcal{Y}$ by $\mathrm{Pr}_{\mathcal{Y}}(x,y) = y$, is the *projection operator* on $\mathcal{Y}$. Further, $\mathcal{X}$ is the space of feasible variables, and $\mathcal{Y}$ is the space of perturbation variables (cf. [1]). The conditions have the following form:

$$\begin{aligned}
&(RC_1^{\Phi}) && \exists x' \in \mathcal{X} \text{ such that } (x',0) \in \mathrm{dom}(\Phi) \text{ and } \Phi(x',\cdot) \text{ is continuous at } 0, \\
&(RC_2^{\Phi}) && \mathcal{X} \text{ and } \mathcal{Y} \text{ are Fréchet spaces, } \Phi \text{ is lower semicontinuous,} \\
& && \text{and } 0 \in \mathrm{sqri}(\mathrm{Pr}_{\mathcal{Y}}(\mathrm{dom}(\Phi))), \\
&(RC_3^{\Phi}) && \mathcal{X} \text{ and } \mathcal{Y} \text{ are Fréchet spaces, } \Phi \text{ is lower semicontinuous, and} \\
& && 0 \in \mathrm{core}(\mathrm{Pr}_{\mathcal{Y}}(\mathrm{dom}(\Phi))), \\
&(RC_4^{\Phi}) && \mathcal{X} \text{ and } \mathcal{Y} \text{ are Fréchet spaces, } \Phi \text{ is lower semicontinuous, and} \\
& && 0 \in \mathrm{int}(\mathrm{Pr}_{\mathcal{Y}}(\mathrm{dom}(\Phi))).
\end{aligned} \tag{9}$$

If $\mathcal{X}$ and $\mathcal{Y}$ are Fréchet spaces and Φ is lower semicontinuous, it holds

$$(RC_1^{\Phi}) \Rightarrow (RC_4^{\Phi}) \Leftrightarrow (RC_3^{\Phi}) \Rightarrow (RC_2^{\Phi}), \tag{10}$$

i.e., the second is the weakest one (see also [1]).

If (RC_1^{Φ}) is fulfilled, the condition $0 \in \mathrm{Pr}_{\mathcal{Y}}(\mathrm{dom}(\Phi))$ holds since it is equivalent with $\exists x' \in \mathcal{X} : (x', 0) \in \mathrm{dom}(\Phi)$. If $(RC_i^{\Phi}), i \in \{2,3,4\}$, is fulfilled, this obviously also holds since the sqri, core and int of $\mathrm{Pr}_{\mathcal{Y}}(\mathrm{dom}(\Phi))$ are subsets of the set $\mathrm{Pr}_{\mathcal{Y}}(\mathrm{dom}(\Phi))$.

We have to ensure that the perturbation function Φ_{FL} is proper and convex. The convexity follows by the convexity of f, g, and S. Further, Φ_{FL} is proper since f is proper and $A^{-1}(\mathrm{dom}(f)) \cap S \cap g^{-1}(-C) \neq \emptyset$. These properties will be maintained in the following (sub)sections.

For the given perturbation function Φ_{FL} it holds

$$\begin{aligned}
(y,z) \in \mathrm{Pr}_{\mathcal{Y}\times\mathcal{Z}}(\mathrm{dom}(\Phi_{FL})) & \\
\Leftrightarrow \quad & \exists x \in \mathcal{X} : \Phi_{FL}(x,y,z) < +\infty \\
\Leftrightarrow \quad & \exists x \in S : Ax + y \in \mathrm{dom}(f), g(x) \in z - C \\
\Leftrightarrow \quad & \exists x \in S : (y,z) \in (\mathrm{dom}(f) - Ax) \times (C + g(x)) \\
\Leftrightarrow \quad & (y,z) \in (\mathrm{dom}(f) \times C) - \bigcup_{x\in S} (Ax, -g(x)) \\
\Leftrightarrow \quad & (y,z) \in (\mathrm{dom}(f) \times C) - (A \times -g)(\Delta_{S^2}).
\end{aligned}$$

The lower semicontinuity of Φ_{FL} is equivalent with the closeness of $\mathrm{epi}\Phi_{FL}$ (see Theorem 2), and it holds

$$\begin{aligned}
\mathrm{epi}\Phi_{FL} = \{(x,y,z,r) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \times \mathbb{R} : (Ax + y, r) \in \mathrm{epi}f\} \\
\cap \{S \times \mathcal{Y} \times \mathcal{Z} \times \mathbb{R}\} \cap \{(x,y,z,r) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \times \mathbb{R} : (x,z) \in \mathrm{epi}_C g\}.
\end{aligned}$$

The closeness of this set is guaranteed if $\mathcal{X}, \mathcal{Y}$, and $\mathcal{Z}$ are Fréchet spaces, f is lower semicontinuous, S is closed, and g is C-epi closed. The regularity condition (RC_2^{Φ}) becomes

$$(RC_{2,FL}) \quad \left| \begin{array}{l} \mathcal{X}, \mathcal{Y}, \text{ and } \mathcal{Z} \text{ are Fréchet spaces, } f \text{ is lower semi-} \\ \text{continuous, } S \text{ is closed, } g \text{ is } C\text{-epi closed, and} \\ \mathbf{0} \in \mathrm{sqri}((\mathrm{dom}(f) \times C) - (A \times -g)(\Delta_{S^2}). \end{array} \right. \tag{11}$$

Analogously, one can rewrite the stronger conditions (RC_3^{Φ}) and (RC_4^{Φ}) using core and int, respectively, instead of sqri and get $(RC_{3,FL})$ and $(RC_{4,FL})$.

The regularity condition (RC_1^{Φ}) becomes under usage of the perturbation function Φ_{FL} in formula (5):

$$(RC_{1,FL}) \quad \left| \begin{array}{l} \exists x' \in A^{-1}(\mathrm{dom}(f)) \cap S \text{ such that } f \text{ is continuous at} \\ Ax' \text{ and } g(x') \in -\mathrm{int}(C). \end{array} \right. \tag{12}$$

We state now the following strong duality theorem:

Theorem 3 (Strong Duality). *Let the spaces $\mathcal{X}, \mathcal{Y}$, and $\mathcal{Z}$, the cone C, the functions f and g, the set S, and the linear mapping A be assumed as at the beginning of the (sub)section and further $A^{-1}(\mathrm{dom}(f)) \cap g^{-1}(-C) \cap S \neq \emptyset$.*

If one of the regularity conditions $(RC_{i,FL}), i \in \{1,2,3,4\}$, is fulfilled, then $v(PS) = v(DS_{FL})$ and the dual has an optimal solution.

Remark 2. If the function f is continuous and the primal problem (PS) has a compact feasible set $\mathcal{A}$, then there exists an optimal solution $\overline{x}$ to (PS).

3.2 The Scalar Optimization Problem (PS^{Σ})

A multiobjective optimization problem with objective functions $f_i, i = 1, \ldots, m$, can be handled by weighting the functions and considering the sum of it, which is a linear scalarization. The arising problem is the subject of this section. Similar perturbations of the primal problem can be found in [4], where the authors consider an optimization problem having also cone constraints but still a weighted sum of convex functions without the composition with a linear continuous mapping.

Assume the functions $f_i : \mathcal{Y} \to \overline{\mathbb{R}}, i = 1, \ldots, m$, to be proper and convex and $g = (g_1, \ldots, g_k)^T : \mathcal{X} \to \mathbb{R}^k$ to be C-convex, $C = \mathbb{R}^k_+$. Further, let λ be the fixed vector $\lambda = (\lambda_1, \ldots, \lambda_m)^T \in \mathrm{int}(\mathbb{R}^m_+)$. Let $A \in \mathcal{L}(\mathcal{X}, \mathcal{Y})$ and $A^{-1}(\bigcap_{i=1}^m \mathrm{dom}(f_i)) \cap g^{-1}(-\mathbb{R}^k_+) \cap S \neq \emptyset$. We consider the scalar optimization problem

$$(PS^{\Sigma}) \quad \inf_{x \in \mathcal{A}} \left\{ \sum_{i=1}^m \lambda_i f_i(Ax) \right\}, \quad \mathcal{A} = \{x \in S : g_i(x) \leq 0, i = 1, \ldots, k\},$$

and the following perturbation function $\Phi^{\Sigma}_{FL} : \mathcal{X} \times \mathcal{Y} \times \cdots \times \mathcal{Y} \times \mathcal{X} \times \cdots \times \mathcal{X} \to \overline{\mathbb{R}}$ in order to separate the conjugate functions of $f_i, i = 1, \ldots, m$, and the conjugate functions of $g_i, i = 1, \ldots, k$, in the dual:

$$\Phi^{\Sigma}_{FL}(x, y^1, \ldots, y^m, z^1, \ldots, z^k) = \begin{cases} \sum_{i=1}^m \lambda_i f_i(Ax + y^i), & x \in S, g_i(x + z^i) \leq 0, i = 1, \ldots, k, \\ +\infty, & \text{otherwise.} \end{cases}$$

The conjugate function $(\Phi^{\Sigma}_{FL})^* : \mathcal{X}^* \times \mathcal{Y}^* \times \cdots \times \mathcal{Y}^* \times \mathcal{X}^* \times \cdots \times \mathcal{X}^* \to \overline{\mathbb{R}}$ is given by

$$(\Phi_{FL}^{\Sigma})^*(x^*, y^{1*}, \ldots, y^{m*}, z^{1*}, \ldots, z^{k*})$$
$$= \sup_{\substack{x \in S, \\ y^i \in \mathcal{Y}, i=1,\ldots,m, \\ z^i \in \mathcal{X}, i=1,\ldots,k, \\ g_i(x+z^i) \leq 0, \\ i=1,\ldots,k}} \left\{ \langle x^*, x \rangle + \sum_{i=1}^{m} \langle y^{i*}, y^i \rangle + \sum_{i=1}^{k} \langle z^{i*}, z^i \rangle - \sum_{i=1}^{m} \lambda_i f_i(Ax + y^i) \right\}.$$

By setting $Ax + y^i =: r^i \in \mathcal{Y}, i = 1, \ldots, m$, and $x + z^i =: s^i \in \mathcal{X}, i = 1, \ldots, k$, we get:

$$-(\Phi_{FL}^{\Sigma})^*(0, y^{1*}, \ldots, y^{m*}, z^{1*}, \ldots, z^{k*})$$
$$= - \sup_{\substack{x \in S, \\ r^i \in \mathcal{Y}, i=1,\ldots,m, \\ s^i \in \mathcal{X}, i=1,\ldots,k, \\ g_i(s^i) \leq 0, i=1,\ldots,k}} \left\{ \sum_{i=1}^{m} \langle y^{i*}, r^i - Ax \rangle + \sum_{i=1}^{k} \langle z^{i*}, s^i - x \rangle - \sum_{i=1}^{m} \lambda_i f_i(r^i) \right\}$$
$$= - \sup_{x \in S} \left\{ - \sum_{i=1}^{m} \langle y^{i*}, Ax \rangle - \sum_{i=1}^{k} \langle z^{i*}, x \rangle \right\} - \sum_{i=1}^{m} \sup_{r^i \in \mathcal{Y}} \left\{ \langle y^{i*}, r^i \rangle - \lambda_i f_i(r^i) \right\}$$
$$- \sum_{i=1}^{k} \sup_{\substack{s^i \in \mathcal{X}, \\ g_i(s^i) \leq 0}} \langle z^{i*}, s^i \rangle$$
$$= -\delta_S^* \left(-A^* \sum_{i=1}^{m} y^{i*} - \sum_{i=1}^{k} z^{i*} \right) - \sum_{i=1}^{m} (\lambda_i f_i)^* (y^{i*}) - \sum_{i=1}^{k} \sup_{\substack{s^i \in \mathcal{X}, \\ g_i(s^i) \leq 0}} \langle z^{i*}, s^i \rangle.$$

We have $(\lambda_i f_i)^*(y^{i*}) = \lambda_i f_i^* \left(\frac{y^{i*}}{\lambda_i} \right)$ since $\lambda_i > 0$ for all $i = 1, \ldots, k$, and by setting $y^{i*} := \frac{y^{i*}}{\lambda_i}, i = 1, \ldots, k$, we get the following dual problem to (PS^{Σ}):

$$(DS_{FL}^{\Sigma}) \quad \sup_{\substack{(y^{1*}, \ldots, y^{m*}, z^{1*}, \ldots, z^{k*}) \\ \in \mathcal{Y}^* \times \ldots \times \mathcal{Y}^* \times \\ \mathcal{X}^* \times \ldots \times \mathcal{X}^*}} \left\{ -(\Phi_{FL}^{\Sigma})^*(0, y^{1*}, \ldots, y^{m*}, z^{1*}, \ldots, z^{k*}) \right\}$$
$$= \sup_{\substack{(y^{1*}, \ldots, y^{m*}, z^{1*}, \ldots, z^{k*}) \\ \in \mathcal{Y}^* \times \ldots \times \mathcal{Y}^* \times \\ \mathcal{X}^* \times \ldots \times \mathcal{X}^*}} \left\{ -\delta_S^* \left(-A^* \sum_{i=1}^{m} y^{i*} - \sum_{i=1}^{k} z^{i*} \right) \right.$$
$$\left. - \sum_{i=1}^{m} (\lambda_i f_i)^* (y^{i*}) - \sum_{i=1}^{k} \sup_{\substack{s^i \in \mathcal{X}, \\ g_i(s^i) \leq 0}} \langle z^{i*}, s^i \rangle \right\}$$

$$= \sup_{\substack{(y^{1*},\dots,y^{m*},\\ z^{1*},\dots,z^{k*})\\ \in\mathcal{Y}^*\times\dots\times\mathcal{Y}^*\times\\ \mathcal{X}^*\times\dots\times\mathcal{X}^*}} \left\{ -\delta_S^*\left(-A^*\sum_{i=1}^m \lambda_i y^{i*} - \sum_{i=1}^k z^{i*}\right) \right.$$

$$\left. -\sum_{i=1}^m \lambda_i f_i^*(y^{i*}) - \sum_{i=1}^k \sup_{\substack{s^i\in\mathcal{X},\\ g_i(s^i)\leq 0}} \left\langle z^{i*}, s^i\right\rangle \right\}. \tag{13}$$

The following theorem holds according to the general approach described in Sect. 3.1 and because of the previous calculations.

Theorem 4 (Weak Duality). *Between* (PS^{Σ}) *and* (DS^{Σ}_{FL})*, weak duality holds, i.e.,* $v(PS^{\Sigma}) \geq v(DS^{\Sigma}_{FL})$.

In order to formulate a strong duality theorem we consider the regularity conditions given in Sect. 3.1. The continuity of $\Phi^{\Sigma}_{FL}(x',\cdot,\dots,\cdot)$ at $\mathbf{0}$ is equivalent with the continuity of f_i at $Ax', i=1,\dots,k$, further $g(x') \in -\text{int}(\mathbb{R}^k_+)$ and the continuity of g at x' (which is equivalent with the continuity of $g_i, i=1,\dots,k$, at x'). So the first regularity condition becomes:

$$(RC^{\Sigma}_{1,FL}) \quad \left| \begin{array}{l} \exists x' \in A^{-1}\left(\bigcap_{i=1}^m \text{dom}(f_i)\right) \cap S \text{ such that } f_i \text{ is} \\ \text{continuous at } Ax', i=1,\dots,m,\ g_i \text{ is continuous at } x', \\ i=1,\dots,k, \text{ and } g(x') \in -\text{int}(\mathbb{R}^k_+). \end{array} \right. \tag{14}$$

We further have, using the definition of the level set:

$$\begin{aligned}
&(y^1,\dots,y^m,z^1,\dots,z^k) \in \text{Pr}_{\mathcal{Y}\times\dots\times\mathcal{Y}\times\mathcal{X}\times\dots\times\mathcal{X}}(\text{dom}(\Phi^{\Sigma}_{FL})) \\
&\Leftrightarrow\ \exists x\in\mathcal{X}:\ \Phi^{\Sigma}_{FL}(x,y^1,\dots,y^m,z^1,\dots,z^k) < +\infty, \\
&\Leftrightarrow\ \exists x\in S: Ax+y^i \in \text{dom}(f_i), i=1,\dots,m,\ g_i(x+z^i)\leq 0, i=1,\dots,k, \\
&\Leftrightarrow\ \exists x\in S: y^i \in \text{dom}(f_i)-Ax, i=1,\dots,m,\ x+z^i \in \text{lev}_0(g_i), i=1,\dots,k, \\
&\Leftrightarrow\ \exists x\in S:\ (y^1,\dots,y^m,z^1,\dots,z^k) \\
&\qquad \in \prod_{i=1}^m (\text{dom}(f_i)-Ax) \times \prod_{i=1}^k (\text{lev}_0(g_i)-x), \\
&\Leftrightarrow\ \exists x\in S:\ (y^1,\dots,y^m,z^1,\dots,z^k) \\
&\qquad \in \prod_{i=1}^m \text{dom}(f_i) \times \prod_{i=1}^k \text{lev}_0(g_i) - (Ax,\dots,Ax,x,\dots,x), \\
&\Leftrightarrow\ (y^1,\dots,y^m,z^1,\dots,z^k)
\end{aligned}$$

$$\in \prod_{i=1}^{m} \operatorname{dom}(f_i) \times \prod_{i=1}^{k} \operatorname{lev}_0(g_i) - \left(\prod_{i=1}^{m} A \times \prod_{i=1}^{k} \operatorname{id}_{\mathcal{X}} \right) (\Delta_{S^{m+k}}). \tag{15}$$

The lower semicontinuity of Φ_{FL}^{Σ}, we need for the further regularity conditions, is equivalent with the closeness of $\operatorname{epi}\Phi_{FL}^{\Sigma}$ (see Theorem 2) and it holds:

Lemma 1. *The set*

$$\operatorname{epi}\Phi_{FL}^{\Sigma} = \left\{ (x, y^1, \ldots, y^m, z^1, \ldots, z^k, r) \in \mathcal{X} \times \mathcal{Y} \times \cdots \times \mathcal{Y} \times \mathcal{X} \times \cdots \times \mathcal{X} \times \mathbb{R} : \right.$$

$$\left. \sum_{i=1}^{m} \lambda_i f_i(Ax + y^i) \leq r \right\} \cap \{S \times \mathcal{Y} \times \cdots \times \mathcal{Y} \times \mathcal{X} \times \cdots \times \mathcal{X} \times \mathbb{R}\}$$

$$\bigcap_{i=1}^{k} \{(x, y^1, \ldots, y^m, z^1, \ldots, z^k, r) \in \mathcal{X} \times \mathcal{Y} \times \cdots \times \mathcal{Y} \times \mathcal{X} \times \cdots \times \mathcal{X} \times \mathbb{R} :$$

$$x + z^i \in \operatorname{lev}_0(g_i)\}$$

is closed if $\mathcal{X}$ and $\mathcal{Y}$ are Fréchet spaces, f_i is lower semicontinuous, $i = 1, \ldots, m$, S is closed, and $\operatorname{lev}_0(g_i)$ is closed, $i = 1, \ldots, k$.

Proof. Let the sequence $(x_n, y_n^1, \ldots, y_n^m, z_n^1, \ldots, z_n^k, r_n) \in \operatorname{epi}(\Phi_{FL}^{\Sigma})$ converge to $(x, y^1, \ldots, y^m, z^1, \ldots, z^k, r) \in \mathcal{X} \times \mathcal{Y} \times \cdots \times \mathcal{Y} \times \mathcal{X} \times \cdots \times \mathcal{X} \times \mathbb{R}$. We show that it holds $(x, y^1, \ldots, y^m, z^1, \ldots, z^k, r) \in \operatorname{epi}(\Phi_{FL}^{\Sigma})$ in order to get the closeness of $\operatorname{epi}(\Phi_{FL}^{\Sigma})$.

We have $\sum_{i=1}^{m} \lambda_i f_i(Ax_n + y_n^i) \leq r_n$. Further it holds $x_n \in S$ and $x_n + z_n^i \in \operatorname{lev}_0(g_i)$ and we get by the lower semicontinuity of $f_i, i = 1, \ldots, m$,

$$\sum_{i=1}^{m} \lambda_i f_i(Ax + y^i) \leq \liminf_{n \to \infty} \sum_{i=1}^{m} \lambda_i f_i(Ax_n + y_n^i) \leq \liminf_{n \to \infty} r_n = r.$$

Since $x \in S$, which follows by the closeness of S, and $\lim_{n \to \infty} (x_n + z_n^i) = x + z^i \in \operatorname{lev}_0(g_i)$, which follows by the closeness of $\operatorname{lev}_0(g_i)$, the assertion follows. □

Remark 3. The fact that $\operatorname{lev}_0(g_i), i = 1, \ldots, k$, is closed is implied by the lower semicontinuity of $g_i, i = 1, \ldots, k$.

With this lemma we get [cf. formula (9)]

$$(RC_{2,FL}^{\Sigma}) \left| \begin{array}{l} \mathcal{X} \text{ and } \mathcal{Y} \text{ are Fréchet spaces, } f_i \text{ is lower semicontinuous,} \\ i = 1, \ldots, m, S \text{ is closed, } \operatorname{lev}_0(g_i) \text{ is closed, } i = 1, \ldots, k, \text{ and} \\ \mathbf{0} \in \operatorname{sqri}\left(\prod_{i=1}^{m} \operatorname{dom}(f_i) \times \prod_{i=1}^{k} \operatorname{lev}_0(g_i) - \left(\prod_{i=1}^{m} A \times \prod_{i=1}^{k} \operatorname{id}_{\mathcal{X}} \right) (\Delta_{S^{m+k}}) \right). \end{array} \right. \tag{16}$$

The conditions $(RC^{\Sigma}_{3,FL})$ and $(RC^{\Sigma}_{4,FL})$ can be formulated analogously using core and int instead of sqri. Then the following theorem holds:

Theorem 5 (Strong Duality). *Let the spaces $\mathcal{X}, \mathcal{Y}$, and $\mathcal{Z} = \mathbb{R}^k$, the cone $C = \mathbb{R}^k_+$, the functions $f_i, i = 1, \ldots, m$, and $g_i, i = 1, \ldots, k$, and the linear mapping A be assumed as at the beginning of the (sub)section and further $A^{-1}(\bigcap_{i=1}^m \operatorname{dom}(f_i)) \cap g^{-1}(-C) \cap S \neq \emptyset$.*

If one of the regularity conditions $(RC^{\Sigma}_{i,FL}), i \in \{1,2,3,4\}$, is fulfilled, then $v(PS^{\Sigma}) = v(DS^{\Sigma}_{FL})$ and the dual has an optimal solution.

Here Remark 2 also holds.

Remark 4. The dual problem (DS^{Σ}_{FL}) given in formula (13) contains terms of the form

$$- \sup_{s^i \in \mathcal{X}, g_i(s^i) \leq 0} \left\langle z^{i*}, s^i \right\rangle = \inf_{s^i \in \mathcal{X}, g_i(s^i) \leq 0} \left\langle -z^{i*}, s^i \right\rangle.$$

We use now Lagrange duality. In case of having strong duality it holds

$$\begin{aligned} \inf_{s^i \in \mathcal{X}, g_i(s^i) \leq 0} \left\langle -z^{i*}, s^i \right\rangle &= \sup_{\mu^{i*} \geq 0} \inf_{s^i \in \mathcal{X}} \left\{ - \left\langle z^{i*}, s^i \right\rangle + \mu^{i*} g_i(s^i) \right\} \\ &= \sup_{\mu^{i*} \geq 0} \left(-(\mu^{i*} g_i)^*(z^{i*}) \right). \end{aligned} \tag{17}$$

In order to have strong duality the following regularity condition has to be fulfilled for $i = 1, \ldots, k$ (cf. [1, Section 3.2.3]):

$$(RC^i_L) \quad \left| \exists x' \in \mathcal{X} : \quad g_i(x') < 0. \right.$$

Assuming that $(RC^{\Sigma}_{i,FL}), i \in \{1,2,3,4\}$, is fulfilled, we additionally only have to ask $(RC^i_L), i = 1, \ldots, k$, to be fulfilled in order to get the following dual problem [cf. formula (13)] and strong duality between (PS^{Σ}) and $(DS^{\Sigma'}_{FL})$:

$$\begin{aligned} &(DS^{\Sigma'}_{FL}) \\ &\sup_{\substack{(y^{1*}, \ldots, y^{m*}, \\ z^{1*}, \ldots, z^{k*}) \\ \in \mathcal{Y}^* \times \cdots \times \mathcal{Y}^* \times \\ \mathcal{X}^* \times \cdots \times \mathcal{X}^*}} \left\{ - \delta^*_S \left(-A^* \sum_{i=1}^m \lambda_i y^{i*} - \sum_{i=1}^k z^{i*} \right) - \sum_{i=1}^m \lambda_i f_i^*(y^{i*}) \right. \\ &\qquad \left. + \sum_{i=1}^k \sup_{\mu^{i*} \geq 0} \left(-(\mu^{i*} g_i)^*(z^{i*}) \right) \right\} \end{aligned}$$

$$= \sup_{\substack{(y^{1*},\dots,y^{m*},\\ z^{1*},\dots,z^{k*},\mu^{1*},\dots,\mu^{k*})\\ \in\mathcal{Y}^*\times\cdots\times\mathcal{Y}^*\times\\ \mathcal{X}^*\times\cdots\times\mathcal{X}^*\times\mathbb{R}_+\times\cdots\times\mathbb{R}_+}} \left\{ -\delta_S^*\left(-A^*\sum_{i=1}^m \lambda_i y^{i*} - \sum_{i=1}^k \mu^{i*} z^{i*}\right) - \sum_{i=1}^m \lambda_i f_i^*(y^{i*}) \right.$$

$$\left. - \sum_{i=1}^k \mu^{i*} g_i^*(z^{i*}) \right\}. \tag{18}$$

The last equality holds by the following consideration. In case of $\mu^{i*} > 0$ we have $(\mu^{i*} g_i)^*(z^{i*}) = \mu^{i*} g_i^*\left(\frac{z^{i*}}{\mu^{i*}}\right)$ and take $z^{i*} := \frac{z^{i*}}{\mu^{i*}}$ such that the term becomes $\mu^{i*} g_i^*(z^{i*})$ for $i = 1,\dots,k$. For $\mu^{i*} = 0$ it holds

$$(0 \cdot g_i)^*(z^{i*}) = \begin{cases} 0, & z^{i*} = 0, \\ +\infty, & \text{otherwise.} \end{cases}$$

Consequently we can always use $\mu^{i*} g_i^*(z^{i*})$ (notice the conventions $0 \cdot (+\infty) := +\infty$ and $0 \cdot (-\infty) := -\infty$ (cf. [1])).

In analogy with Theorem 4 between (PS^{Σ}) and $(DS_{FL}^{\Sigma'})$ weak duality holds, i.e., $v(PS^{\Sigma}) \geq v(DS_{FL}^{\Sigma'})$. Further we have:

Theorem 6 (Strong Duality). *Let the spaces $\mathcal{X}, \mathcal{Y}$, and $\mathcal{Z} = \mathbb{R}^k$, the cone $C = \mathbb{R}_+^k$, the functions $f_i, i = 1,\dots,m$, and $g_i, i = 1,\dots,k$, and the linear mapping A be assumed as at the beginning of the (sub)section and further $A^{-1}(\bigcap_{i=1}^m \operatorname{dom}(f_i)) \cap g^{-1}(-C) \cap S \neq \emptyset$.*

If one of the regularity conditions $(RC_{i,FL}^{\Sigma}), i \in \{1,2,3,4\}$, is fulfilled and (RC_L^i) is fulfilled for $i = 1,\dots,k$, then $v(PS^{\Sigma}) = v(DS_{FL}^{\Sigma'})$ and the dual has an optimal solution.

With respect to the fact mentioned in the above remark, the following theorem providing optimality conditions holds.

Theorem 7. *(a) If one of the regularity conditions $(RC_{i,FL}^{\Sigma}), i \in \{1,2,3,4\}$, is fulfilled, (RC_L^i) is fulfilled for $i = 1,\dots,k$, and (PS^{Σ}) has an optimal solution $\bar{x}$, then $(DS_{FL}^{\Sigma'})$ has an optimal solution $(\bar{y}^{1*},\dots,\bar{y}^{m*},\bar{z}^{1*},\dots,\bar{z}^{k*},\overline{\mu}^{1*},\dots,\overline{\mu}^{k*}) \in \mathcal{Y}^* \times \cdots \times \mathcal{Y}^* \times \mathcal{X}^* \times \cdots \times \mathcal{X}^* \times \mathbb{R}_+ \times \cdots \times \mathbb{R}_+$ such that the following optimality conditions are fulfilled:*

(i) $f_i(A\bar{x}) + f_i^*(\bar{y}^{i*}) - \langle \bar{y}^{i*}, A\bar{x} \rangle = 0, \quad i = 1,\dots,m,$

(ii) $\overline{\mu}^{i*} g_i(\bar{x}) = 0, \quad i = 1,\dots,k,$

(iii) $\overline{\mu}^{i*}(g_i^*(\bar{z}^{i*}) - \langle \bar{z}^{i*}, \bar{x} \rangle) = 0, \quad i = 1,\dots,k,$

(iv) $$\sum_{i=1}^{m} \lambda_i \left\langle \overline{y}^{i*}, A\overline{x} \right\rangle + \sum_{i=1}^{k} \overline{\mu}^{i*} \left\langle \overline{z}^{i*}, \overline{x} \right\rangle = \inf_{x \in S} \left\{ \sum_{i=1}^{m} \lambda_i \left\langle \overline{y}^{i*}, Ax \right\rangle + \sum_{i=1}^{k} \overline{\mu}^{i*} \left\langle \overline{z}^{i*}, x \right\rangle \right\}.$$

(b) Let $\overline{x}$ be feasible to (PS^{Σ}) and $(\overline{y}^{1*}, \ldots, \overline{y}^{m*}, \overline{z}^{1*}, \ldots, \overline{z}^{k*}, \overline{\mu}^{1*}, \ldots, \overline{\mu}^{k*}) \in \mathcal{Y}^* \times \cdots \times \mathcal{Y}^* \times \mathcal{X}^* \times \cdots \times \mathcal{X}^* \times \mathbb{R}_+ \times \cdots \times \mathbb{R}_+$ be feasible to $(DS_{FL}^{\Sigma'})$ fulfilling the optimality conditions (i)–(iv). Then $\overline{x}$ is an optimal solution for (PS^{Σ}), $(\overline{y}^{1*}, \ldots, \overline{y}^{m*}, \overline{z}^{1*}, \ldots, \overline{z}^{k*}, \overline{\mu}^{1*}, \ldots, \overline{\mu}^{k*})$ is an optimal solution for $(DS_{FL}^{\Sigma'})$, and $v(PS^{\Sigma}) = v(DS_{FL}^{\Sigma'})$.

Proof. (a) Since (PS^{Σ}) has an optimal solution $\overline{x} \in S$, one of the conditions $(RC_{i,FL}^{\Sigma}), i \in \{1,2,3,4\}$, is fulfilled and (RC_L^i) is fulfilled for $i = 1, \ldots, k$, Theorem 6 guarantees the existence of an optimal solution for $(DS_{FL}^{\Sigma'})$, namely $(\overline{y}^{1*}, \ldots, \overline{y}^{m*}, \overline{z}^{1*}, \ldots, \overline{z}^{k*}, \overline{\mu}^{1*}, \ldots, \overline{\mu}^{k*})$, such that

$$\begin{aligned}
& v(PS^{\Sigma}) = v(DS_{FL}^{\Sigma'}) \\
\Leftrightarrow\ & \sum_{i=1}^{m} \lambda_i f_i(A\overline{x}) = -\sum_{i=1}^{m} \lambda_i f_i^*(\overline{y}^{i*}) - \sum_{i=1}^{k} \overline{\mu}^{i*} g_i^*(\overline{z}^{i*}) \\
& \qquad - \delta_S^* \left(-A^* \sum_{i=1}^{m} \lambda_i \overline{y}^{i*} - \sum_{i=1}^{k} \overline{\mu}^{i*} \overline{z}^{i*} \right) \\
\Leftrightarrow\ & \sum_{i=1}^{m} \lambda_i \left[f_i(A\overline{x}) + f_i^*(\overline{y}^{i*}) - \left\langle \overline{y}^{i*}, A\overline{x} \right\rangle \right] + \sum_{i=1}^{m} \lambda_i \left\langle \overline{y}^{i*}, A\overline{x} \right\rangle \\
& \qquad + \sum_{i=1}^{k} \overline{\mu}^{i*} \left[g_i^*(\overline{z}^{i*}) + g_i(\overline{x}) - \left\langle \overline{z}^{i*}, \overline{x} \right\rangle \right] - \sum_{i=1}^{k} \overline{\mu}^{i*} g_i(\overline{x}) \\
& \qquad + \sum_{i=1}^{k} \overline{\mu}^{i*} \left\langle \overline{z}^{i*}, \overline{x} \right\rangle + \delta_S^* \left(-A^* \sum_{i=1}^{m} \lambda_i \overline{y}^{i*} - \sum_{i=1}^{k} \overline{\mu}^{i*} \overline{z}^{i*} \right) = 0.
\end{aligned}$$

By applying Young's inequality [cf. formula (1)] and having $\overline{\mu}^{i*} \geq 0$ and $g_i(\overline{x}) \leq 0$, this sum which is equal to zero consists of $m + 2k + 1$ nonnegative terms. Thus the inequalities have to be fulfilled with equality and we get the following equivalent formulation:

$$\begin{cases} (i) & f_i(A\bar{x}) + f_i^*(\bar{y}^{i*}) - \langle \bar{y}^{i*}, A\bar{x} \rangle = 0, \quad i = 1, \ldots, m, \\ (ii) & \bar{\mu}^{i*} g_i(\bar{x}) = 0, \quad i = 1, \ldots, k, \\ (iii) & \bar{\mu}^{i*}(g_i^*(\bar{z}^{i*}) + g_i(\bar{x}) - \langle \bar{z}^{i*}, \bar{x} \rangle) = 0, \quad i = 1, \ldots, k, \\ (iv) & \sum\limits_{i=1}^{m} \lambda_i \langle \bar{y}^{i*}, A\bar{x} \rangle + \sum\limits_{i=1}^{k} \bar{\mu}^{i*} \langle \bar{z}^{i*}, \bar{x} \rangle \\ & + \delta_S^* \left(-A^* \sum\limits_{i=1}^{m} \lambda_i \bar{y}^{i*} - \sum\limits_{i=1}^{k} \bar{\mu}^{i*} \bar{z}^{i*} \right) = 0, \end{cases}$$

$$\Leftrightarrow \begin{cases} (i) & f_i(A\bar{x}) + f_i^*(\bar{y}^{i*}) - \langle \bar{y}^{i*}, A\bar{x} \rangle = 0, \quad i = 1, \ldots, m, \\ (ii) & \bar{\mu}^{i*} g_i(\bar{x}) = 0, \quad i = 1, \ldots, k, \\ (iii) & \bar{\mu}^{i*}(g_i^*(\bar{z}^{i*}) - \langle \bar{z}^{i*}, \bar{x} \rangle) = 0, \quad i = 1, \ldots, k, \\ (iv) & \sum\limits_{i=1}^{m} \lambda_i \langle \bar{y}^{i*}, A\bar{x} \rangle + \sum\limits_{i=1}^{k} \bar{\mu}^{i*} \langle \bar{z}^{i*}, \bar{x} \rangle \\ & = \inf\limits_{x \in S} \left\{ \sum\limits_{i=1}^{m} \lambda_i \langle \bar{y}^{i*}, Ax \rangle + \sum\limits_{i=1}^{k} \bar{\mu}^{i*} \langle \bar{z}^{i*}, x \rangle \right\}. \end{cases}$$

(b) All calculations in part (a) can be carried out in reverse direction. □

3.3 *The Vector Optimization Problem* (PV)

In this section we consider a vector optimization problem with an objective function being the composition of a convex function f and a linear continuous operator A and cone and geometric constraints in analogy with the scalar problem in Sect. 3.1.

The properties of the spaces and sets were defined at the beginning of the section. Assume the function $f : \mathcal{Y} \to \mathcal{V}^\bullet$ to be proper and K-convex and $g : \mathcal{X} \to \mathcal{Z}$ to be C-convex, fulfilling $A^{-1}(\mathrm{dom}(f)) \cap g^{-1}(-C) \cap S \neq \emptyset$.

By $\mathrm{Min}(V, K)$ we denote the *set of minimal points of* V, where $y \in V \subseteq \mathcal{V}$ is said to be a *minimal point* of the set V if $y \in V$ and there exists no $y' \in V$ such that $y' \leq_K y$. The set $\mathrm{Max}(V, K)$ of *maximal points* of V is defined analogously.

We consider the following vector optimization problem:

$$(PV) \quad \operatorname*{Min}_{x \in \mathcal{A}} f(Ax), \quad \mathcal{A} = \{x \in S : g(x) \in -C\}.$$

We investigate a duality approach with respect to *properly efficient solutions* in the sense of linear scalarization (cf. [1]), that are defined as follows:

Definition 2 (Properly Efficient Solution). An element $\bar{x} \in \mathcal{A}$ is said to be a properly efficient solution to (PV) if $\bar{x} \in A^{-1}(\mathrm{dom}(f))$ and $\exists v^* \in K^{*0}$ such that $\langle v^*, f(A\bar{x}) \rangle \leq \langle v^*, f(Ax) \rangle$, $\forall x \in \mathcal{A}$.

Further, we define *efficient solutions*:

Definition 3 (Efficient Solution). An element $\bar{x} \in \mathcal{A}$ is said to be an efficient solution to (PV) if $\bar{x} \in A^{-1}(\mathrm{dom}(f))$ and $f(A\bar{x}) \in \mathrm{Min}((f \circ A)(A^{-1}(\mathrm{dom}(f)) \cap \mathcal{A}), K)$. This means that if $\bar{x} \in A^{-1}(\mathrm{dom}(f)) \cap \mathcal{A}$ then for all $x \in A^{-1}(\mathrm{dom}(f)) \cap \mathcal{A}$ from $f(Ax) \leqq_K f(A\bar{x})$ follows $f(A\bar{x}) = f(Ax)$.

Depending on the perturbation function Φ_{FL}, the dual problem to (PV) can be given by (cf. [1, Section 4.3.1]):

$$(DV_{FL}) \qquad \operatorname*{Max}_{(v^*,y^*,z^*,v) \in \mathcal{B}_{FL}} v,$$

where

$$\mathcal{B}_{FL} = \{(v^*,y^*,z^*,v) \in K^{*0} \times \mathcal{Y}^* \times Z^* \times \mathcal{V} : \\ \langle v^*, v\rangle \leq -(v^*\Phi_{FL})^*(0,-y^*,-z^*)\}.$$

Here we consider the perturbation function $\Phi_{FL} : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \mathcal{V}^\bullet$, analogously as given in the scalar case in Sect. 3.1:

$$\Phi_{FL}(x,y,z) = \begin{cases} f(Ax+y), & x \in S, g(x) \in z - C, \\ +\infty_K, & \text{otherwise.} \end{cases} \tag{19}$$

The formula for the conjugate function of $v^*\Phi_{FL} : \mathcal{X}^* \times \mathcal{Y}^* \times \mathcal{Z}^* \to \overline{\mathbb{R}}$ follows from the calculations above (cf. formulas (6) and (7)):

$$-(v^*\Phi_{FL})^*(x^*,y^*,z^*) = -(-z^*g)^*_S(x^* - A^*y^*) - (v^*f)^*(y^*) - \delta_{-C^*}(z^*).$$

From this formula, the dual problem of (PV) can be deduced. It is given by

$$(DV_{FL}) \operatorname*{Max}_{(v^*,y^*,z^*,v) \in \mathcal{B}_{FL}} v, \tag{20}$$

where

$$\begin{aligned} \mathcal{B}_{FL} &= \{(v^*,y^*,z^*,v) \in K^{*0} \times \mathcal{Y}^* \times Z^* \times \mathcal{V} : \\ &\qquad \langle v^*, v\rangle \leq -(v^*\Phi_{FL})^*(0,-y^*,-z^*)\} \\ &= \{(v^*,y^*,z^*,v) \in K^{*0} \times \mathcal{Y}^* \times C^* \times \mathcal{V} : \\ &\qquad \langle v^*, v\rangle \leq -(v^*f)^*(-y^*) - (z^*g)^*_S(A^*y^*)\}. \end{aligned}$$

Weak duality follows from [1, Theorem 4.3.1]:

Theorem 8 (Weak Duality). *There is no $x \in \mathcal{A}$ and no $(v^*,y^*,z^*,v) \in \mathcal{B}_{FL}$ such that $f(Ax) \leq_K v$.*

To formulate a strong and converse duality theorem we have to state a regularity condition. The conditions $(RC_{1,FL})$ and $(RC_{2,FL})$ from above [cf. formulas (11) and

(12)] (as well as $(RC_{3,FL})$ and $(RC_{4,FL})$) can, under some small modifications, be applied for the vectorial case. It holds (see [1, Remark 4.3.1]):

Remark 5. For having strong duality we only have to assume that for all $v^* \in K^{*0}$ the scalar optimization problem $\inf_{x\in\mathcal{X}}(v^*\Phi_{FL})(x,0,0)$ is stable.

This can be guaranteed by assuming that $\mathcal{X}$ and the spaces of perturbation variables, $\mathcal{Y}$ and $\mathcal{Z}$, are Fréchet spaces, f is star K-lower semicontinuous, S is closed, g is C-epi closed, and $\mathbf{0} \in \operatorname{sqri}((\operatorname{dom}(f) \times C) - (A \times -g)(\Delta_{S^2}))$ since $\operatorname{dom}(f) = \operatorname{dom}(v^* f)$. This follows by Theorem 3.

Further, this fact can be seen in the proof of the strong and converse duality Theorem 9. We have:

$$(RCV_{1,FL}) \quad \left|\begin{array}{l}\exists x' \in A^{-1}(\operatorname{dom}(f)) \cap S \text{ such that } f \text{ is continuous at } Ax' \\ \text{and } g(x') \in -\operatorname{int}(C),\end{array}\right.$$

which is identical with $(RC_{1,FL})$ [cf. formula (12)] and

$$(RCV_{2,FL}) \quad \left|\begin{array}{l}\mathcal{X}, \mathcal{Y}, \text{ and } \mathcal{Z} \text{ are Fréchet spaces, } f \text{ is star } K\text{-lower} \\ \text{semicontinuous, } S \text{ is closed, } g \text{ is } C\text{-epi closed, and} \\ \mathbf{0} \in \operatorname{sqri}((\operatorname{dom}(f) \times C) - (A \times -g)(\Delta_{S^2})).\end{array}\right.$$

Analogously we formulate $(RCV_{3,FL})$ and $(RCV_{4,FL})$ by using core and int instead of sqri.

Before we prove a strong and converse duality theorem we want to formulate the following preliminary result (in analogy with [1, Theorem 4.3.3], to which we also refer for the proof):

Lemma 2. *Assume that $\mathcal{B}_{FL}$ is nonempty and that one of the regularity conditions $(RCV_{i,FL}), i \in \{1,2,3,4\}$, is fulfilled. Then*

$$\mathcal{V} \setminus \operatorname{cl}\left((f \circ A)(A^{-1}(\operatorname{dom}(f)) \cap \mathcal{A}) + K\right) \subseteq \operatorname{core}(h(\mathcal{B}_{FL})),$$

where $h : K^{*0} \times \mathcal{Y}^* \times C^* \times \mathcal{V} \to \mathcal{V}$ is defined by $h(v^*, y^*, z^*, v) = v$.

Now we get the following theorem (in analogy with [1, Theorem 4.3.7]):

Theorem 9 (Strong and Converse Duality). *(a) If one of the conditions $(RCV_{i,FL}), i \in \{1,2,3,4\}$, is fulfilled and $\bar{x} \in \mathcal{A}$ is a properly efficient solution to (PV), then there exists $(\bar{v}^*, \bar{y}^*, \bar{z}^*, \bar{v}) \in \mathcal{B}_{FL}$, an efficient solution to (DV_{FL}), such that $f(A\bar{x}) = \bar{v}$.*

(b) If one of the conditions $(RCV_{i,FL}), i \in \{1,2,3,4\}$, is fulfilled, $(f \circ A)(A^{-1}(\operatorname{dom}(f)) \cap \mathcal{A}) + K$ is closed and $(\bar{v}^*, \bar{y}^*, \bar{z}^*, \bar{v})$ is an efficient solution to (DV_{FL}), then there exists $\bar{x} \in \mathcal{A}$, a properly efficient solution to (PV), such that $f(A\bar{x}) = \bar{v}$.

The following proof of the theorem will be done in analogy with the one of [1, Theorem 4.3.2 and 4.3.4]:

Proof. (a) Since $\bar{x} \in \mathcal{A}$ is a properly efficient solution, there exists $\bar{v}^* \in K^{*0}$ such that $\bar{x}$ is an optimal solution to the scalarized problem

$$\inf_{x \in \mathcal{A}} \langle \bar{v}^*, f(Ax) \rangle .$$

Using that one of the regularity conditions $(RCV_{i,FL}), i \in \{1,2,3,4\}$, is fulfilled we can apply Theorem 3. Therefore we have to show that the problem $\inf_{x \in \mathcal{A}} \langle \bar{v}^*, f(Ax) \rangle$ with the assumptions given by $(RCV_{i,FL})$ fulfills the regularity condition $(RC_{i,FL})$ for fixed $i \in \{1,2,3,4\}$.

Let us consider $(RCV_{i,FL}), i \in \{2,3,4\}$. Since f is assumed to be star K-lower semicontinuous, $v^* f$ is lower semicontinuous by definition. The assumptions regarding $\mathcal{X}, \mathcal{Y}, \mathcal{Z}, S$ and g hold analogously. Further, we have $\operatorname{dom}(v^* f) = \operatorname{dom}(f)$ and therefore

$$\operatorname{sqri}((\operatorname{dom}(v^* f) \times C) - (A \times -g)(\Delta_{S^2})) = \operatorname{sqri}((\operatorname{dom}(f) \times C) - (A \times -g)(\Delta_{S^2}))$$

and analogously for core and int. Thus the conditions $(RC_{i,FL}), i \in \{2,3,4\}$, hold.

The continuity of $v^* f$ follows by the continuity of f and since $\operatorname{dom}(v^* f) = \operatorname{dom}(f)$ the fulfillment of $(RC_{1,FL})$ follows by assuming $(RCV_{1,FL})$.

From the mentioned theorem it follows that there exist $\bar{z}^* \in C^*$ and $\bar{y}^* \in \mathcal{Y}^*$ such that $\langle \bar{v}^*, f(A\bar{x}) \rangle = -(\bar{v}^* f)^*(-\bar{y}^*) - (\bar{z}^* g)_S^*(A^* \bar{y}^*)$. It follows that for $\bar{v} = f(A\bar{x})$ the element $(\bar{v}^*, \bar{y}^*, \bar{z}^*, \bar{v})$ is feasible to the dual problem (DV_{FL}). By weak duality, which was given in Theorem 8, it follows that $(\bar{v}^*, \bar{y}^*, \bar{z}^*, \bar{v})$ is an efficient solution.

(b) Assume that $\bar{v} \notin (f \circ A)\left(A^{-1}(\operatorname{dom}(f)) \cap \mathcal{A}\right) + K$. From Lemma 2 it follows that $\bar{v} \in \operatorname{core}(h(\mathcal{B}_{FL}))$. By definition of the core for $k \in K \setminus \{0\}$ there exists $\lambda > 0$ such that $v_\lambda := \bar{v} + \lambda k \geq_K \bar{v}$ and $v_\lambda \in h(\mathcal{B}_{FL})$. This contradicts the fact that $(\bar{v}^*, \bar{y}^*, \bar{z}^*, \bar{v})$ is an efficient solution for (DV_{FL}) since v_λ is in the image set of (DV_{FL}) and $v_\lambda \geq_K \bar{v}$.

Thus we have $\bar{v} \in (f \circ A)\left(A^{-1}(\operatorname{dom}(f)) \cap \mathcal{A}\right) + K$, which means that there exists $\bar{x} \in A^{-1}(\operatorname{dom}(f)) \cap \mathcal{A}$ and $\bar{k} \in K$ such that $\bar{v} = f(A\bar{x}) + \bar{k}$. By Theorem 8 there is no $x \in \mathcal{A}$ and no $(v^*, y^*, z^*, v) \in \mathcal{B}_{FL}$ such that $f(Ax) \leq_K v$ and hence it holds $\bar{k} = 0$. Consequently we have $f(A\bar{x}) = \bar{v}$ and $\bar{x}$ is a properly efficient solution to (PV) which follows by the following calculation. It holds

$$\begin{aligned} \langle \bar{v}^*, f(A\bar{x}) \rangle &= \langle \bar{v}^*, \bar{v} \rangle \leq -(\bar{v}^* f)^*(-\bar{y}^*) - (\bar{z}^* g)_S^*(A^* \bar{y}^*) \\ &= -(\bar{v}^* \Phi_{FL})^*(0, -\bar{y}^*, -\bar{z}^*) \leq \inf_{x \in \mathcal{A}} \langle \bar{v}^*, f(Ax) \rangle . \end{aligned}$$

Here the last inequality follows by weak duality for the scalarized problem (cf. Sect. 3.1) and thus $\bar{x}$ turns out to be a properly efficient solution to (PV) by Definition 2 fulfilling $\bar{v} = f(A\bar{x})$. □

3.4 *The Vector Optimization Problem* (PV^m)

We assume that the spaces $\mathcal{V}$ and $\mathcal{Z}$ are finite dimensional, especially $\mathcal{V}=\mathbb{R}^m$, $K=\mathbb{R}^m_+, \mathcal{Z}=\mathbb{R}^k$, and $C=\mathbb{R}^k_+$. Further, let the functions $f_i:\mathcal{Y}\to\overline{\mathbb{R}}$, $i=1,\ldots,m$, be proper and convex and $g=(g_1,\ldots,g_k)^T:\mathcal{X}\to\mathbb{R}^k$ be $\mathbb{R}^k_+$-convex, fulfilling $A^{-1}(\bigcap_{i=1}^m \operatorname{dom}(f_i))\cap g^{-1}(-\mathbb{R}^k_+)\cap S\neq\emptyset$. We consider the following vector optimization problem:

$$(PV^m)\quad \operatorname*{Min}_{x\in\mathcal{A}} \begin{pmatrix} f_1(Ax)\\ \vdots \\ f_m(Ax)\end{pmatrix}, \qquad \mathcal{A}=\{x\in S: g_i(x)\leq 0, i=1,\ldots,k\}.$$

The perturbation function $\Phi^m_{FL}:\mathcal{X}\times\mathcal{Y}\times\cdots\times\mathcal{Y}\times\mathcal{X}\times\cdots\times\mathcal{X}\to\mathbb{R}^{m\bullet}$ is similar to the one in Sect. 3.2 in order to separate the conjugate functions of $f_i, i=1,\ldots,m$, and the conjugate functions of $g_i, i=1,\ldots,k$, in the dual problem:

$$\begin{aligned}&\Phi^m_{FL}(x,y^1,\ldots,y^m,z^1,\ldots,z^k)\\ &\quad=\begin{cases}(f_1(Ax+y^1),\ldots,f_m(Ax+y^m))^T, & x\in S, g_i(x+z^i)\leq 0, i=1,\ldots,k,\\ +\infty, & \text{otherwise.}\end{cases}\end{aligned}$$

Thus the dual problem becomes by taking $v:=(v_1,\ldots,v_m)^T\in\mathbb{R}^m$ and $v^*=(v_1^*,\ldots,v_m^*)^T\in\operatorname{int}(\mathbb{R}^m_+)$ [cf. formula (20)]:

$$(DV^m_{FL})\quad \operatorname*{Max}_{(v^*,y^*,z^*,v)\in\mathcal{B}^m_{FL}} v,$$

where

$$\begin{aligned}\mathcal{B}^m_{FL}=\Big\{&(v^*,y^{1*},\ldots,y^{m*},z^{1*},\ldots,z^{k*},v)\\ &\in\operatorname{int}(\mathbb{R}^m_+)\times\mathcal{Y}^*\times\cdots\times\mathcal{Y}^*\times\mathcal{X}^*\times\cdots\times\mathcal{X}^*\times\mathbb{R}^m:\\ &v^Tv^*\leq-(v^*\Phi^m_{FL})^*(0,-y^{1*},\ldots,-y^{m*},-z^{1*},\ldots,-z^{k*})\Big\}.\end{aligned}$$

Especially it holds

$$\begin{aligned}&-(v^*\Phi^m_{FL})^*(0,-y^{1*},\ldots,-y^{m*},-z^{1*},\ldots,-z^{k*})\\ &\quad=-\sup_{\substack{x\in S,\\ y^i\in\mathcal{Y},i=1,\ldots,m,\\ z^i\in\mathcal{X},i=1,\ldots,k,\\ g_i(x+z^i)\leq 0,i=1,\ldots,k}}\left\{-\sum_{i=1}^m v_i^*f_i(Ax+y^i)-\sum_{i=1}^m\langle y^{i*},y^i\rangle-\sum_{i=1}^k\langle z^{i*},z^i\rangle\right\}\end{aligned}$$

$$= -\delta_S^* \left(A^* \sum_{i=1}^{m} v_i^* y^{i*} + \sum_{i=1}^{k} z^{i*} \right) - \sum_{i=1}^{m} v_i^* f_i^*(-y^{i*}) - \sum_{i=1}^{k} \sup_{\substack{s^i \in \mathcal{X}, \\ g_i(s^i) \leq 0}} \left\langle -z^{i*}, s^i \right\rangle,$$

which arises from formula (13). The dual becomes:

$$(DV_{FL}^m) \quad \operatorname*{Max}_{(v^*, y^{1*}, \ldots, y^{k*}, z^{1*}, \ldots, z^{k*}, v) \in \mathcal{B}_{FL}^m} v, \tag{21}$$

where

$$\begin{aligned} \mathcal{B}_{FL}^m = \Bigg\{ & (v^*, y^{1*}, \ldots, y^{m*}, z^{1*}, \ldots, z^{k*}, v) \\ & \in \operatorname{int}(\mathbb{R}_+^m) \times \mathcal{Y}^* \times \cdots \times \mathcal{Y}^* \times \mathcal{X}^* \times \cdots \times \mathcal{X}^* \times \mathbb{R}^m : \\ & v^T v^* \leq -\delta_S^* \left(A^* \sum_{i=1}^{m} v_i^* y^{i*} + \sum_{i=1}^{k} z^{i*} \right) - \sum_{i=1}^{m} v_i^* f_i^*(-y^{i*}) \\ & - \sum_{i=1}^{k} \sup_{\substack{s^i \in \mathcal{X}, \\ g_i(s^i) \leq 0}} \left\langle -z^{i*}, s^i \right\rangle \Bigg\}. \end{aligned}$$

The following weak duality theorem holds:

Theorem 10 (Weak Duality). *Between (PV^m) and (DV_{FL}^m) weak duality holds, i.e., there is no $x \in \mathcal{A}$ and no $(v^*, y^{1*}, \ldots, y^{m*}, z^{1*}, \ldots, z^{k*}, v) \in \mathcal{B}_{FL}^m$ such that $f(Ax) \leq_K v$.*

In order to formulate a strong and converse duality theorem, we have to state some regularity conditions. Therefore let us first consider the following lemma:

Lemma 3. *Let be $f = (f_1, \ldots, f_m)^T : \mathcal{Y} \to \mathbb{R}^{m\bullet}$. If $f_i, i = 1, \ldots, m$, is lower semicontinuous, then f is star K-lower semicontinuous, where $K = \mathbb{R}_+^m$.*

Proof. Let be $v^* = (v_1^*, \ldots, v_m^*)^T \in K = \mathbb{R}_+^m$. If we assume that $f_i, i = 1, \ldots, m$, is lower semicontinuous, then $\langle v^*, f \rangle = \sum_{i=1}^m v_i^* f_i$ is lower semicontinuous since it is a sum of lower semicontinuous functions and $v_i^* \geq 0, i = 1, \ldots, m$ (cf. [1, Prop. 2.2.11]). This means by definition that f is star K-lower semicontinuous. □

As mentioned in the last section it is possible to apply the regularity conditions given in the scalar case under some modifications. So formulas (14) and (16) become

$$(RCV_{1,FL}^m) \quad \left| \begin{array}{l} \exists x' \in A^{-1} \left(\bigcap_{i=1}^{m} \operatorname{dom}(f_i) \right) \cap S \text{ such that} \\ f_i \text{ is continuous at } Ax', i = 1, \ldots, m, \\ g_i \text{ is continuous at } x', i = 1, \ldots, k, \\ \text{and } g(x') \in -\operatorname{int}(\mathbb{R}_+^k), \end{array} \right. \tag{22}$$

$$(RCV^m_{2,FL}) \left| \begin{array}{l} \mathcal{X} \text{ and } \mathcal{Y} \text{ are Fréchet spaces, } f_i \text{ is lower semicontinuous,} \\ i = 1,\ldots,m, S \text{ is closed, } \operatorname{lev}_0(g_i) \text{ is closed, } i = 1,\ldots,k, \text{and } \mathbf{0} \in \\ \operatorname{sqri}\left(\Pi_{i=1}^m \operatorname{dom}(f_i) \times \Pi_{i=1}^k \operatorname{lev}_0(g_i) - \left(\Pi_{i=1}^m A \times \Pi_{i=1}^k \operatorname{id}_{\mathcal{X}}\right)(\Delta_{S^{m+k}})\right). \end{array} \right. \tag{23}$$

The conditions $(RCV^m_{3,FL})$ and $(RCV^m_{4,FL})$ can be formulated analogously using core and int instead of sqri. The following theorem holds:

Theorem 11. *(a) If one of the conditions $(RCV^m_{i,FL}), i \in \{1,2,3,4\}$, is fulfilled and $\bar{x} \in \mathcal{A}$ is a properly efficient solution to (PV^m), then there exists $(\bar{v}^*, \bar{y}^{1*}, \ldots, \bar{y}^{m*}, \bar{z}^{1*}, \ldots, \bar{z}^{k*}, \bar{v}) \in \mathcal{B}^m_{FL}$, an efficient solution to (DV^m_{FL}), such that $f(A\bar{x}) = \bar{v}$.*

(b) If one of the conditions $(RCV^m_{i,FL}), i \in \{1,2,3,4\}$, is fulfilled, $(f \circ A)(A^{-1}(\bigcap_{i=1}^m \operatorname{dom}(f_i)) \cap \mathcal{A}) + K$ is closed and $(\bar{v}^*, \bar{y}^{1*}, \ldots, \bar{y}^{m*}, \bar{z}^{1*}, \ldots, \bar{z}^{k*}, \bar{v})$ is an efficient solution to (DV^m_{FL}), then there exists $\bar{x} \in \mathcal{A}$, a properly efficient solution to (PV^m), such that $f(A\bar{x}) = \bar{v}$.

Remark 6. Remark 4 can be applied here which leads to the dual problem [cf. formula (18)]

$$(DV^{m'}_{FL}) \qquad \operatorname*{Max}_{(v^*, y^{1*}, \ldots, y^{k*}, z^{1*}, \ldots, z^{k*}, \mu^{1*}, \ldots, \mu^{k*}, v) \in \mathcal{B}^m_{FL}} v, \tag{24}$$

where

$$\begin{aligned} \mathcal{B}^m_{FL} = \Big\{ & (v^*, y^{1*}, \ldots, y^{m*}, z^{1*}, \ldots, z^{k*}, \mu^{1*}, \ldots, \mu^{k*}, v) \\ & \in \operatorname{int}(\mathbb{R}^m_+) \times \mathcal{Y}^* \times \cdots \times \mathcal{Y}^* \times \mathcal{X}^* \times \cdots \times \mathcal{X}^* \times \mathbb{R}_+ \times \cdots \times \mathbb{R}_+ \times \mathbb{R}^m : \\ & v^T v^* \leq -\delta^*_S \left(A^* \sum_{i=1}^m v_i y^{i*} - \sum_{i=1}^k \mu^{i*} z^{i*} \right) - \sum_{i=1}^m v_i f_i^*(-y^{i*}) \\ & + \sum_{i=1}^k \mu^{i*} g_i^*(-z^{i*}) \Big\}. \end{aligned}$$

Further, weak duality holds by construction and Theorem 11 holds analogously under the assumption that one of the regularity conditions $(RCV^m_{i,FL}), i \in \{1,2,3,4\}$, is fulfilled and (RC^i_L) is fulfilled for $i = 1,\ldots,k$.

References

1. R.I. Boţ, S.-M. Grad, and G. Wanka. *Duality in Vector Optimization*. Springer-Verlag, Berlin Heidelberg, 2009.
2. R.I. Boţ, S.-M. Grad, and G. Wanka. New regularity conditions for Lagrange and Fenchel-Lagrange duality in infinite dimensional spaces. *Mathematical Inequalities & Applications*, 12(1):171–189, 2009.
3. R.I. Boţ, G. Kassay, and G. Wanka. Strong duality for generalized convex optimization problems. *Journal of Optimization Theory and Applications*, 127(1):44–70, 2005.
4. R.I. Boţ and G. Wanka. A new duality approach for multiobjective convex optimization problems. *Journal of Nonlinear and Convex Analysis*, 3(1):41–57, 2002.
5. I. Ekeland and R. Temam. *Convex analysis and variational problems*. North-Holland Publishing Company, Amsterdam, 1976.
6. V. Jeyakumar, W. Song, N. Dinh, and G.M. Lee. Stable strong duality in convex optimization. Applied Mathematics Report AMR 05/22, University of New South Wales, Sydney, Australia, 2005.
7. D.T. Luc. *Theory of vector optimization*. Number 319 in Lecture notes in Economics and Mathematical Systems. Springer-Verlag, Berlin, 1989.
8. R.T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, 1970.
9. R.T. Rockafellar. Conjugate duality and optimization. Regional Conference Series in Applied Mathematics 16. Society for Industrial and Applied Mathematics, Philadelphia, 1974.
10. G. Wanka and R.I. Boţ. On the relations between different dual problems in convex mathematical programming. In P. Chamoni, R. Leisten, A. Martin, J. Minnermann, and H. Stadtler, editors, *Operations Research Proceedings 2001*, pages 255–262. Springer-Verlag, Berlin, 2002.
11. G. Wanka, R.I. Boţ, and E. Vargyas. Conjugate duality for multiobjective composed optimization problems. *Acta Mathematica Hungarica*, 116(3):117–196, 2007.

A PTAS for Weak Minimum Routing Cost Connected Dominating Set of Unit Disk Graph

Qinghai Liu, Zhao Zhang, Yanmei Hong, Weili Wu, and Ding-Zhu Du

Abstract Considering the virtual backbone problem of wireless sensor networks with the shortest path constraint, the problem can be modeled as finding a minimum routing cost connected dominating set (MOC-CDS) in the graph. In this chapter, we study a variation of the MOC-CDS problem. Let k be a fixed positive integer. For any two vertices u, v of G and a vertex subset $S \subseteq V(G)$, denote $\ell_S(u,v)$ the length of the shortest (u,v)-path in G all whose intermediate vertices are in S and define

$$g(u,v) = \begin{cases} d(u,v)+4, & \text{if } d(u,v) \le k+1; \\ (1+\frac{4}{k})d(u,v)+6, & \text{if } d(u,v) > k+1. \end{cases}$$

The g-MOC-CDS problem asks for a subset S with the minimum cardinality such that S is a connected dominating set of G and $\ell_S(u,v) \le g(u,v)$ for any pair of vertices (u,v) of G. Clearly, g-MOC-CDS can serve as a virtual backbone of the network such that the routing cost is not increased too much. In this chapter, we give a PTAS for the g-MOC-CDS problem on unit disk graphs.

Key words Connected dominating set • Unit disk graph • PTAS

Q. Liu • Z. Zhang (✉)
College of Mathematics and System Sciences, Xinjiang University, Urumqi, Xinjiang, 830046, China
e-mail: hxhzz@163.com

Y. Hong
Department of Mathematics, Shanghai Univiersity, Shanghai, 200444, China

W. Wu • D.-Z. Du
Department of Computer Science, University of Texas at Dallas, Richardson, TX 75080, USA

A. Chinchuluun et al. (eds.), *Optimization, Simulation, and Control*, Springer Optimization and Its Applications 76, DOI 10.1007/978-1-4614-5131-0_9,

1 Introduction

Dominating set and connected dominating set have a wide range of applications in wireless sensor networks to make the system hierarchial and efficient. A *dominating set* (DS) of a graph G is a vertex subset D such that every vertex in $V(G) \setminus D$ has at least a neighbor in D. A *connected dominating set* (CDS) is a dominating set D such that $G[D]$ is connected, where $G[D]$ is the subgraph induced by D. In general, we expect cardinality of a CDS to be as small as possible. Thus the minimum connected dominating set (MCDS) problem has been proposed and been studied extensively, especially on the unit disk graph, which is a widely adopted model for the homogeneous wireless sensor network. In a *unit disk graph* G, every vertex corresponds to a point in the plane; there is an edge between two vertices in G if and only if the Euclidean distance between the two points is not greater than one. For more details on the study of algorithms for CDS, see, e.g., [1–5, 10–17, 19–21].

Although CDS is an efficient virtual backbone for routing protocols, the routing path between some pairs of vertices might increase greatly than the shortest path. In order to surmount this, Ding et al. [6] and Willson et al. [18] proposed a special CDS problem—*minimum routing cost connected dominating set* (MOC-CDS). Besides the constraints of CDS, MOC-CDS has an additional constraint that between any two nodes, the routing cost does not increase if messages are relayed only through the MOC-CDS. The formal definition is:

Definition 1 (MOC-CDS [6]). Given a connected graph $G = (V, E)$, the minimum routing cost connected dominating set (MOC-CDS) problem is to find a minimum size node set $D \subseteq V$ such that for every pair of nodes $u, v \in V$, there exists a shortest path between u and v all of whose intermediate nodes belong to D.

Ding et al. [6] showed that MOC-CDS has no polynomial time approximation with performance ratio $\rho \ln \Delta$ for $0 < \rho < 1$ unless $NP \subseteq DTIME(n^{O(\log \log n)})$, where Δ is the maximum node degree of the input graph G. They also gave a polynomial time distributed approximation algorithm with performance ratio $H(\frac{\delta(\delta-1)}{2})$, where H is the harmonic function, i.e., $H(k) = \sum_{i=1}^{k} \frac{1}{i}$.

However, in some networks, the MOC-CDS may be very large. In some cases, it may even be as large as the whole node set. Motivated by this situation, Du et al. [7] relaxed the requirement of shortest path and proposed the following problem.

Definition 2 (αMOC-CDS [7]). Given a graph G, compute the minimum CDS D such that for any two nodes u and v in $V(G)$, $m_D(u, v) \leq \alpha \cdot m_G(u, v)$, where $m_D(u, v)$ is the number of intermediate nodes for a path to connect u and v through D.

Du et al. [7] showed that for any $\alpha \geq 1$, αMOC-CDS in general graphs is APX-hard and hence has no PTAS unless NP=P. When restricted to unit disk graphs, the αMOC-CDS problem remains to be NP-hard. In [8], Du et al. transformed a minimum non-submodular cover problem into a problem of minimum submodular cover with submodular cost, and as an application, they provided a constant-approximation algorithm for the αMOC-CDS problem in unit disk graphs for $\alpha \geq 5$.

In [9], Du et al. gave a PTAS for αMOC-CDS in unit disk graphs when $\alpha \geq 5$, i.e., for any $\varepsilon > 0$, there is a $(1+\varepsilon)$-approximation algorithm for this problem which runs in time $n^{O(1/\varepsilon^4)}$.

A natural question is: how about $\alpha < 5$? In this chapter, we consider a variation of the αMOC-CDS problem as described in the following.

For two vertices $u, v \in V(G)$, we use $d(u,v)$ to denote the length of the shortest (u,v)-path in G, i.e., the number of edges on the path. It should be noted that $d(u,v)$ differs from $m_G(u,v)$ in Definition 2 by exactly one.

Let S be a vertex subset of G. A path P of G is called to be *fully intersecting with* S if all the inner vertices of P lie in S. If the vertex set S in clear from the context, then P is also called a *fully intersecting path*. We use the notion $P_S(u,v)$ to represent a shortest fully intersecting (u,v)-path and $\ell_S(u,v)$ is the length of $P_S(u,v)$. In this chapter, we consider a relaxed version of MOC-CDS, called *g-minimum routing cost dominating set* (g-MOC-CDS) problem. Let k be a fixed positive integer. For any two vertices u, v, we define

$$g(u,v) = \begin{cases} d(u,v)+4, & \text{if } d(u,v) \leq k+1; \\ (1+\frac{4}{k})d(u,v)+6, & \text{if } d(u,v) > k+1. \end{cases}$$

Definition 3 (g-MOC-CDS). Let G be a graph. The g-MOC-CDS problem asks for a subset S with the minimum cardinality such that S is a CDS and $\ell_S(u,v) \leq g(u,v)$ for any pair of vertices (u,v) of G.

In this chapter, we give a PTAS for the g-MOC-CDS problem of unit disk graphs.

2 Problem Transformation

In order to solve g-MOC-CDS problem, we do some transformation on this problem and propose another similar problem as follows.

Definition 4 (k-MOC-CDS). Let G be a graph. The k-MOC-CDS problem asks for a subset S with the minimum cardinality such that S is a CDS and $\ell_S(u,v) \leq d(u,v)+4$ for any pair of vertices (u,v) of G with $d(u,v) \leq k+1$.

Clearly, k-MOC-CDS is a relaxation of g-MOC-CDS, which seems to be easier than g-MOC-CDS. However, we can prove that these two problems are in fact equivalent with each other. For the simplicity of statement, we call the set S in Definition 3 *g-MOC-CDS* and call the set S in Definition 4 *k-MOC-CDS*.

Lemma 1. *Let G be a graph and k be a positive integer. Then S is a k-MOC-CDS of G if and only if S is a g-MOC-CDS of G.*

Proof. The sufficiency is clear. Next we show the necessity. Let S be a k-MOC-CDS. For any pair of vertices (u,v), it suffices to show that $\ell_S(u,v) \leq (1+\frac{4}{k})d(u,v)+6$ if $d(u,v) > k+1$. Let $d(u,v) = d = qk+r$, where $0 \leq r \leq k-1$. Given a shortest (u,v)-path P, we can find out q vertices $u_1, \ldots, u_q$, which divide P into $q+1$ segments such

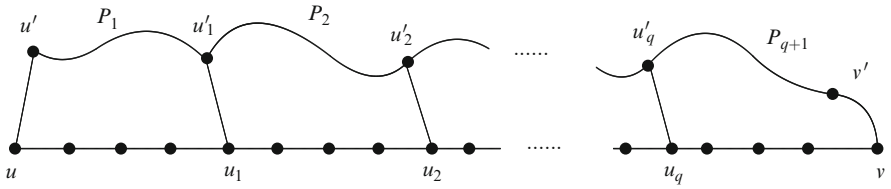


Fig. 1 An illustration of $P_S(u,v)$

that the segments $(u,u_1),(u_1,u_2),\ldots(u_{q-1},u_q)$ have length k and (u_q,v) has length r. By noting that S is a dominating set, there exists a neighbor of u in S, say u'. Then $d(u',u_1)\leq k+1$. According to our assumption that S is a k-MOC-CDS, there exists a fully intersecting (u',u_1)-path with length at most $d(u',u_1)+4\leq k+5$. Let u'_1 be the neighbor of u_1 on this path. Denote the (u',u'_1)-segment of the path as P_1. Then P_1 has length at most $k+4$ and all of the vertices are in S. Then consider the pair of vertices (u'_1,u_2). Since $d(u'_1,u_2)\leq k+1$, the above procedure can be continued (see Fig. 1). By this way, we find $q+1$ paths $P_1,\ldots,P_q,P_{q+1}$ such that P_i has length at most $k+4$ for $i=1,\ldots,q$, and P_{q+1} has length at most $r+4$. Furthermore each path has all its vertices in S. By noting that the initial vertex of P_1 is adjacent with u and the end vertex of P_{q+1} is adjacent with v, we see that the combination of $P_1,\ldots,P_{q+1}$ yields a fully intersecting (u,v)-path with length at most $q(k+4)+r+4+2=qk+r+4q+6\leq(1+\frac{4}{k})d(u,v)+6$. □

As a consequence of Lemma 1, in order to find a PTAS for g-MOC-CDS problem, we only need to find a PTAS for k-MOC-CDS problem. In the next section, we first give a constant approximation for k-MOC-CDS.

3 A Constant Approximation

In this section, we give an algorithm to compute a k-MOC-CDS for a given unit disk graph. The algorithm makes use of maximal independent set. A vertex set I is called an *independent set* if there is no edge between any two vertices in I. A *maximal independent set* (MIS) is an independent set which cannot be properly contained in any other independent set. It is easy to verify that an MIS is also a DS.

The next lemma shows the correctness of Algorithm 1.

Lemma 2. *The set S output by Algorithm 1 is a k-MOC-CDS of G.*

Proof. First we show that

$$\ell_S(u,v)\leq d(u,v)+4 \text{ for any } u,v\in V(G) \text{ with } d(u,v)\leq k+1. \tag{1}$$

In fact, for any two vertices $u,v\in V(G)$, by noting that an MIS is also a DS of G, we see that both u and v have neighbors in I, say u' and v', respectively. Then

Algorithm 1 A CONSTANT APPROXIMATION OF A k-MOC-CDS PROBLEM ON UDG

Input: The geometric representation of a connected unit disk graph G and an integer k.
Output: A k-MOC-CDS S of G.

```
Let I be an MIS of G and C = ∅.
for all u, v ∈ I do
    if d(u, v) ≤ k + 3 then
        Add all the inner vertices of a shortest (u, v)-path into C.
    end if
end for
Return S = I ∪ C.
```

$d(u',v') \le d(u,v)+2 \le k+3$. By the construction of C, we see that $\ell_S(u',v') = d(u',v') \le d(u,v)+2$. Thus $\ell_S(u,v) \le \ell_S(u',v')+2 \le d(u,v)+4$ and (1) holds.

Next, we show that S is a CDS of G. In fact, since I is a DS of G, so is $S = I \cup C$. Suppose that $G[S]$ is not connected. Let C_1 and C_2 be two components of $G[S]$. Let $u \in V(C_1)$ and $v \in V(C_2)$ such that $d_G(u,v) = \min\{d_G(x,y) \mid x \in V(C_1), y \in V(C_2)\}$. By noting that u, v lie in different components of $G[S]$, we see that the path $P_S(u,v)$ does not exist and neither does $\ell_S(u,v)$. From (1), we have $d_G(u,v) > k+1$. Let w be the vertex on a shortest (u,v)-path such that $d(u,w) = k+1$. Consider the path $P_S(u,w)$. Let w' be the neighbor of w on $P_S(u,w)$. Then $w' \in V(C_1)$ since $P_S(u,w)$ connects w' to vertex u in C_1. However, $d(w',v) \le d(u,v) - d(u,w) + 1 = d(u,v) - k < d(u,v)$, contradicting the selection of the pair (u,v). Thus we have shown that S is a CDS and the lemma is proved. □

In order to show the performance ratio of Algorithm 1, we prove the following result first.

Lemma 3. *Let* $\alpha_k = 2(k+2)(k+3.5)^2 + 1$. *Then* $|S| \le \alpha_k \cdot |I|$.

Proof. For each vertex v in I, we draw a disk D_v with center v and radius 0.5. Then any two of these disks are disjoint since I is an independent set. Furthermore, if v has hop-distance at most $k+3$ from u, then v has Euclidean distance at most $k+3$ from u. Thus D_v is contained in a disk with center u and radius $k+3+0.5$. Thus the number of vertices having hop-distance at most $k+3$ from u is no more than $\frac{(k+3.5)^2\pi}{0.5^2\pi} = 4(k+3.5)^2$.

Construct an auxiliary graph H with vertex set I, $uv \in E(H)$ if and only if u and v are at most $k+3$ hops away from each other in G. Then each vertex in H has degree at most $4(k+3.5)^2$ by the analysis in the previous paragraph. It follows that $|E(H)| \le \frac{1}{2} \cdot 4(k+3.5)^2|I| = 2(k+3.5)^2|I|$. Since for each edge uv in $E(H)$, we add at most $k+2$ vertices into C to connect u and v, we have $|C| \le 2(k+2)(k+3.5)^2|I|$. Thus $|S| = |C| + |I| \le [1 + 2(k+2)(k+3.5)^2]|I| = \alpha_k|I|$. □

It is well known that every vertex has at most five neighbors in any MIS. Thus the vertex in A has also at most five neighbors in I, where A is a minimum CDS of G and I is as the IS in Algorithm 1. Combining this with that every vertex in I either lies in A or has at least one neighbor in A, we see that $|I| \le 5|A|$. By noting that the k-MOC-CDS is a special CDS of G, we see that the minimum k-MOC-CDS

has cardinality at least $|A|$. Combining this with Lemma 3, we have the following theorem.

Theorem 1. *Algorithm 1 is a $5\alpha_k$-approximation for k-MOC-CDS of unit disk graph.*

4 A PTAS

In this section, we combine the constant approximation algorithm in Section 3 with the partition and shifting technique to build a PTAS for the k-MOC-CDS problem on unit disk graph.

Let W, U be two subsets of vertices of G with $W \subseteq U$. A vertex set $S \subseteq U$ is called a *k-MOC W-DS* of U if $S \subseteq U$ is a DS of W and for any pair of vertices (u,v) of W with $d_{G[W]}(u,v) \leq k+1$, there exists a fully intersecting (u,v)-path with length at most $d_{G[W]}(u,v)+4$, where $G[W]$ is the subgraph of G induced by W. Specially, if $W = U$, then k-MOC W-DS is k-MOC-CDS.

Let $Q = \{(x,y) \mid 0 \leq x \leq q, 0 \leq y \leq q\}$ be a minimal square containing all the nodes. For a given real number $\varepsilon > 0$, let m be an integer with $m = \lceil \frac{10\alpha_k(3k+14)}{\varepsilon} \rceil$. Set $p = \lfloor q/m \rfloor + 1$, and $\tilde{Q} = \{(x,y) \mid -m \leq x \leq mp, -m \leq y \leq mp\}$. Divide $\tilde{Q}$ into $(p+1) \times (p+1)$ grid such that each cell is an $m \times m$ square. Denote this partition as $P(0)$. For $i = 0, 1, \ldots, m-1$, $P(i)$ is the partition obtained by shifting $P(0)$ such that the left-bottom corner of $P(i)$ is at the coordinate $(i-m, i-m)$. For each cell e, the *central region* C_e of e is the region of e such that each point is at least distance $\lceil \frac{k}{2} \rceil + 3$ away from the boundary of e. The *inside region* I_e of C_e is the region contained in e such that each point in this region is at least $k+3$ away from the boundary of C_e. Let $B_e = e - I_e$. Then B_e and C_e have an overlap. For simplicity of statement, we write $V_e^I = V_e \cap I_e$, $V_e^C = V_e \cap C_e$, $V_e^B = V_e \cap B_e$ (see Fig. 2).

Denote $\beta(m) = \alpha_k \cdot \lceil \sqrt{2}m \rceil^2$. We have the following lemma.

Lemma 4. *Let e be a square with width m. There exists a k-MOC V_e^C-DS of V_e with order at most $\beta(m)$.*

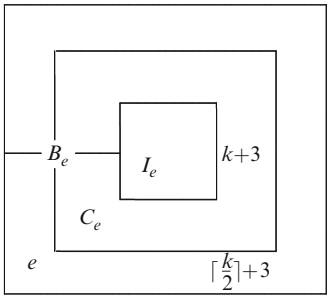


Fig. 2 An illustration of C_e, I_e, B_e

Proof. The proof can be done by constructing such a k-MOC V_e^C-DS of V_e by using Algorithm 1.

Note that the graph $G[V_e]$ may not be connected. Assume that there are t components of $G[V_e]$ and $C_1, C_2, \ldots, C_t$ are the t components. Applying Algorithm 1 to C_i, let S_{C_i} be the output. By Lemma 2 and Lemma 3, we see that S_{C_i} is a k-MOC-CDS of C_i, and $|S_{C_i}| \leq \alpha_k |I_{C_i}|$ where I_{C_i} is an MIS of C_i. Let $S = \bigcup_{i=1}^{t} S_{C_i}$ and $I = \bigcup_{i=1}^{t} I_{C_i}$. Then S is a DS of $G[V_e]$, I is an MIS of $G[V_e]$, and $|S| \leq \alpha_k |I|$. Furthermore, by noting that any pair (u,v) with $d_{G[V_e]}(u,v) \leq k+1$ lie in a same C_i and $d_{C_i}(u,v) = d_{G[V_e]}(u,v)$, we see that S is a k-MOC V_e-DS of V_e, and thus a k-MOC V_e^C-DS of V_e.

Next, we show that $|S| \leq \beta(m)$. We partition e into small squares, each of which has width at most $\sqrt{2}/2$. Then there are at most $\lceil \frac{m}{\sqrt{2}/2} \rceil^2 = \lceil \sqrt{2}m \rceil^2$ small squares. Since each such small square may contain at most one vertex from I, we see that $|I| \leq \lceil \sqrt{2}m \rceil^2$. Hence $|S| \leq \alpha_k |I| \leq \beta(m)$. □

Corollary 1. *Let e be a square with width m. The minimum k-MOC V_e^C-DS of V_e can be found in $|V_e|^{O(m^2)}$ time.*

Proof. Let S_e be a minimum k-MOC V_e^C-DS of V_e. By Lemma 4, $|S_e| \leq \beta(m)$. Thus to find a k-MOC V_e^C-DS of V_e by exhaust search, it suffices to check all the subsets of V_e with order no more than $\beta(m)$. Then the running time is bounded by $O\left(\binom{|V_e|}{1} + \binom{|V_e|}{2} + \cdots + \binom{|V_e|}{\beta(m)}\right) = |V_e|^{O(m^2)}$. □

The PTAS is described in Algorithm 2. It first finds a k-MOC-CDS S_0 by the constant approximation in Algorithm 1. For each $i = 0, 1, \ldots, m-1$, let $B(i)$ be the boundary region of the partition $P(i)$, i.e., $B(i) = \bigcup_{e \in P(i)} B_e$. The first loop of Algorithm 2 finds out a partition i^* such that $|B(i^*) \cap S_0| = \min_{0 \leq i \leq m-1} |B(i) \cap S_0|$. In the second loop, a minimum k-MOC V_e^C-DS of V_e is found for each cell e, using exhaust search in Corollary 1. The final output S of the algorithm is the union of these local optimal solutions and the vertices of S_0 which fall into $B(i^*)$.

The next lemma shows the correctness and the running time of Algorithm 2.

Lemma 5. *The output S of Algorithm 2 is a k-MOC-CDS of $V(G)$ and the running time is $n^{O(\varepsilon^{-2})}$.*

Proof. First, we show that S is a k-MOC-CDS of $V(G)$.

Claim 1. For any two vertices u, v of G with $d(u,v) \leq k+1$, we have $\ell_S(u,v) \leq d_G(u,v) + 4$.

Assume that $u \in V_e$ and $v \in V_{e'}$.

First we consider the case $e \neq e'$. By $d(u,v) \leq k+1$, we see that $u \in V_e^B$ and $v \in V_{e'}^B$. Consider the path $P_{S_0}(u,v)$. Since S_0 is a k-MOC-CDS, we see that $P_{S_0}(u,v)$ has length $\ell_{S_0}(u,v) \leq d_G(u,v) + 4 \leq k+5$. Furthermore, $P_{S_0}(u,v)$ cannot contain any vertices in $I_{e''}$ for any cell $e'' \in P(i^*)$, since otherwise there would exist a segment of $P_{S_0}(u,v)$ with length at least $k + 3 + \lceil \frac{k}{2} \rceil + 3 > k+5$, a contradiction.

Algorithm 2 A PTAS OF A k-MOC-CDS PROBLEM ON UDG

Input: The geometric representation of a connected unit disk graph G, an integer k and a positive real number $\varepsilon > 0$.
Output: A k-MOC-CDS S of V.

Let $m = \frac{10\alpha_k(3k+14)}{\varepsilon}$.
Use the $5\alpha_k$-approximation algorithm to compute a k-MOC-CDS S_0.
Let $S = V$.
for $i = 0$ to $m-1$ **do**
    Let $S_b = \emptyset$.
    for all $e \in P(i)$ **do**
        Let $S_b = S_b \cup (V_e^B \cap S_0)$.
    end for
    if $|S_b| < |S|$ **then**
        Let $S = S_b$ and $i^* = i$.
    end if
end for
for all $e \in P(i^*)$ and $V_e \neq \emptyset$ **do**
    Compute S_e which is a minimum k-MOC V_e^C-DS of V_e by exhaust search.
    Let $S = S \cup S_e$.
end for
Return S.

Thus all the inner vertices of $P_{S_0}(u,v)$ lie in the boundary region $B(i^*)$. By the construction of S in Algorithm 2, we see that all the inner vertices of $P_{S_0}(u,v)$ lie in S. Thus $\ell_S(u,v) \leq \ell_{S_0}(u,v) \leq d(u,v)+4$.
Next, we consider the case $e = e'$. If $u \in V_e^C$ and $v \in V_e^C$ simultaneously, then the path $P_{S_e}(u,v)$ has length $\ell_{S_e}(u,v) \leq d_{G[V_e]}(u,v)+4$ since S_e is a k-MOC V_e^C-DS of V_e. Furthermore, we claim that

$$d_{G[V_e]}(u,v) = d_G(u,v). \tag{2}$$

Suppose that this is not true, then there exists a shortest (u,v)-path in G containing at least one vertex outside of e, say w. Then $d(u,w) \geq \lceil \frac{k}{2} \rceil + 3$ and $d(w,v) \geq \lceil \frac{k}{2} \rceil + 3$. Thus $d(u,v) \geq k+6$, contradicting that $d(u,v) \leq k+1$. Thus (2) is proved. It follows that $\ell_{S_e}(u,v) \leq d_{G[V_e]}(u,v)+4 = d_G(u,v)+4$. Moreover, by noting that $S_e \subseteq S$, we see that $\ell_S(u,v) \leq \ell_{S_e}(u,v) \leq d_G(u,v)+4$. Next, assume, without loss of generality, that $u \in e - C_e$. In this case, v cannot lie in I_e since otherwise $d(u,v) \geq k+3 > k+1$. Thus $v \in B_e$. It can be proved that $P_{S_0}(u,v)$ cannot contain any vertex in I_e. Suppose this is not true. Then there exists a vertex $w \in V(P_{S_0}(u,v)) \cap I_e$. Since $\rho(u,w) > k+3$ and $\rho(u,v) \leq k+1$ we have $\rho(v,w) \geq \rho(u,w) - \rho(u,v) > 2$ and thus $d(v,w) \geq 3$. It follows that $\ell_{S_0} \geq d(u,w) + d(w,v) \geq k+4+3 > d_G(u,v)+4$, a contradiction. Similarly, a contradiction can be obtained if $P_{S_0}(u,v)$ contains some vertex in $I_{e''}$ for

any $e'' \in P(i^*)$. Thus we have proved that all the inner vertices of $P_{S_0}(u,v)$ lie in $B(i^*)$. By the construction of S in Algorithm 2, we see that $\ell_S(u,v) \le \ell_{S_0}(u,v) \le d(u,v)+4$. Claim 1 is proved.

Claim 2. S is a CDS of G.

By the construction of S, we see that S is a DS of G, since the vertices in C_e are dominated by S_e and the vertices in $e - C_e$ are dominated by $S_0 \cap B(i^*)$. Similar to the proof of Lemma 2, we can prove that S is a CDS of G.

Next, we consider the time complexity. It is clear that the most time-consuming step of Algorithm 2 is to compute S_e. By Corollary 1, we see that it takes $|V_e|^{O(m^2)}$-time to compute S_e for each cell e. Thus the complexity of Algorithm 2 is bounded by $\sum_{e \in P(i^*)} |V_e|^{O(m^2)} = |V|^{O(m^2)} = |V|^{O(\varepsilon^{-2})}$. □

Next, we analyze the performance ratio of Algorithm 2.

Lemma 6. *Algorithm 2 is a $(1+\varepsilon)$-approximation for k-MOC-CDS problem of a UDG.*

Proof. Let A be a minimum k-MOC-CDS of G and S be the output of Algorithm 2. It suffices to show that $|S| \le (1+\varepsilon)|A|$. For each cell $e \in P(i^*)$, let $A_e = A \cap e$. Then $A = \bigcup_{e \in P(i^*)} A_e$, where i^* is the integer as in Algorithm 2. First we have the following claim.

Claim 1. A_e is a k-MOC V_e^C-DS of V_e.

First we see that A_e is a DS of V_e^C, since each vertex in V_e^C has a neighbor in A and this neighbor must lie in V_e.

Suppose there exists a pair of vertices (u,v) in V_e^C with $d_{G[V_e]}(u,v) \le k+1$ such that $\ell_{A_e}(u,v) \ge d_{G[V_e]}(u,v)+5 \ge d_G(u,v)+5$. By noting that A is a k-MOC-CDS, we see that $\ell_A(u,v) \le d_G(u,v)+4 \le k+5$. Then $P_A(u,v)$ has to use some vertex not in A_e, say w. The vertex w divides $P_A(u,v)$ into two segments, (u,w)-segment and (w,v)-segment. By noting that $u,v \in C_e$ and $w \notin V_e$, we see that both the two segments have length at least $\lceil \frac{k}{2} \rceil + 3$. Thus $\ell_A(u,v) \ge 2(\lceil \frac{k}{2} \rceil + 3) \ge k+6 > k+5$, a contradiction. Claim 1 is proved.

Claim 2. Each vertex of S_0 lies in at most $6k+28$ V_e^b's among all cells of the m partitions.

From the definition, we see that each point in B_e has Euclidian distance at most $k+3+\lceil \frac{k}{2} \rceil + 3 \le \frac{3k}{2}+7$ from the boundary of e. Then this claim can be seen from Fig. 3.

By Claim 1, we see that $|A_e| \ge |S_e|$, where S_e is the local optimal solution for cell e in Algorithm 2. Let $S_i^B = B(i) \cap S_0$. Then $S_{i^*}^B = \arg\min\{|S_i^B| \mid 0 \le i \le m-1\}$ and $S = S_{i^*}^B \cup \left(\bigcup_{e \in P(i^*)} S_e\right)$. Thus

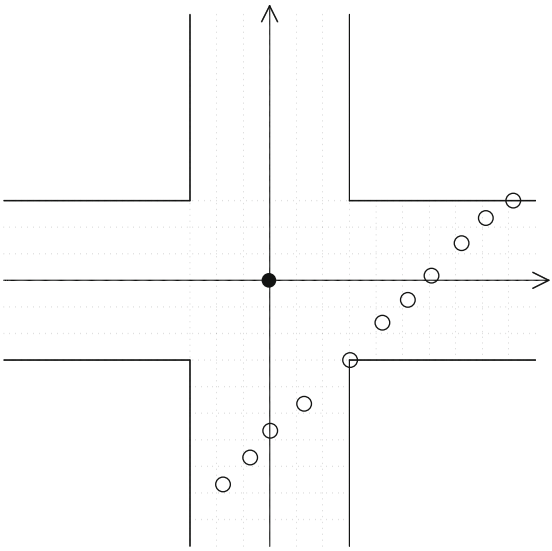


Fig. 3 When the partition shifts, each vertex falls into at most $6k+28$ boundary regions

$$
\begin{aligned}
m|S| &\le m\Big|\bigcup_{e\in P(i^*)} S_e\Big| + \sum_{i=0}^{m-1} |S_i^B| \\
&\le m \sum_{e\in P(i)} |S_e| + \sum_{i=0}^{m-1} |\bigcup_{e\in P(i)} V_e^B \cap S_0| \\
&\le m \sum_{e\in P(i)} |A_e| + (6k+28)|S_0| \\
&\le m|A| + (6k+28)5\alpha_k|A| \\
&= (m+10(3k+14)\alpha_k)|A|.
\end{aligned}
$$

The fourth inequality holds because S_0 is a $5\alpha_k$-approximation. It follows that $|S| \le (1+\frac{10(3k+14)\alpha_k}{m})|A| = (1+\varepsilon)|A|$ and the performance ratio follows. □

By Lemma 1, we see that k-MOC-CDS problem is equivalent to the g-MOC-CDS problem. Thus we have the following theorem.

Theorem 2. *Algorithm 2 is a* $(1+\varepsilon)$*-approximation for g-MOC-CDS of UDG.*

Acknowledgements This research was supported by NSFC (61222201), the Key Project of Chinese Ministry of Education (208161), the Program for New Century Excellent Talents in University, and the project Sponsored by SRF for ROCS, SEM. The research was jointly sponsored in part by MEST, Korea, under WCU (R33-2008-000-10044-0), an NRF Grant under (KRF-2008-314-D00354), and MKE, Korea, under ITRC NIPA-2010-(C1090-1021-0008), and also from NSF of USA under grants CCF0829993 and CCF0728851.

References

1. K.M. Alzoubi, P.J. Wan, O. Frieder. New distributed algorithm for connected dominating set in wireless ad hoc networks, in Porc. 35th Hawaii Int. Conf. System Science, Big Island, Hawaii (2002)
2. V. Bharghavan and B. Das, Routing in ad hoc networks using minimum connected dominating sets, International Conference on Communication, Montreal, Canada (1997)
3. J. Blum, D. Min, A. Thaeler and X.Z. Cheng, Connected dominating set in sensor networks and MANETs, Handbook of Combinatorial Optimization, D.-Z. Du and P. Pardalos (Eds.), Kluwer Academic Publishers. Dordrecht/Boston/London 329–369 (2004)
4. M. Cadei, X. Cheng, D.-Z. Du, Connected domination in ad hoc wireless networks, in Proc. 6th Int. Conf. Computer Science and Informatics. (2002)
5. X. Cheng, X. Huang, D. Li, W. Wu and D.-Z. Du, Polynomial-time approximation scheme for minimum connected dominating set in ad hoc wireless networks, Networks, **42**, 202–208 (2003).
6. L. Ding, X.F. Gao, W.L. Wu, W.J. Lee, X. Zhu and D.-Z. Du, Distributed contruction of connected dominating sets with minimum routing cost in wireless network, IEEE International Conference on Distributed Computing Systerms. 448–457 (2010)
7. L. Ding, W.L. Wu, J. Willson, H.J. Du, W.J. Lee, and D.-Z. Du, Efficient algorithms for topology control problem with routing cost constraint in wireless networks, IEEE Trans. Parallel Distrib. Syst. **22**(10): 1601–1609 (2011)
8. H.J. Du, W.L. Wu, W.J. Lee, Q.H. Liu, Z. Zhang and D.-Z. Du, On minimum submodular cover with submodular cost, J. Global Optimization, doi:10.1007/s10898-010-9563-3.
9. H.W. Du, Q. Ye, J.F. Zhong, Y.X. Wang, W.J. Lee and H. Park, PTAS for minimum connected dominating set with routing cost constraint in wireless sensor networks, COCOA (1) 2010: 252–259
10. S. Funke, M. Segal, A simple improved distributed algorithm for minimum CDS in unit disk graphs, ACM Transactions on Sensor Networks. **2**, 444–453 (2006).
11. B. Gao, Y.H. Yang, H.Y. Ma, An efficient approximation scheme for minimum connected dominating set in wireless ad hoc networks. IEEE Vehicular Technology conference No. 60, Los Angeles CA 2004.
12. S. Guha and S. Khuller, Approximation algorithms for connected dominating sets, Algorithmica. 20, 374–387 (1998)
13. B. Han, W.J. Jia, Design and analysis of connected dominating set formation for topology control in wireless ad hoc networks, Proc 14th International Conference on Computer Communications and Networks (ICCCN 2005) 7–12 (2005)
14. H.B. Hunt III, M.V. Marathe, V. Radhakrishnan, S.S. Ravi, D.J. Rosenkrantz and R.E. Stearns, NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs, J. Algorithms. **26**, 238–274 (1998)
15. M. Min, H.W. Du, X.H. Jia, C.X. Huang, S.C.H. Huang and W.L. Wu, Improving construction for connected dominating set with steiner tree in wireless sensor networks, J. Global Optim. **35**, 111–119 (2006)
16. L. Ruan, H.W. Du, X.H. Jia, W.L. Wu, Y.S. Li and K. Ko, A greedy approximation for minimum connected dominating sets, Theoretical Computer Science. **329** 325–330 (2004)
17. P.J. Wan, L.X. Wang and F. Yao, Two-phased approximation algorithms for minimum CDS in wireless ad hoc networks, The 28th International Conference on Distributed Computing Systems. 337–344 (2008)
18. J.K. Willson, X.F. Gao, Z.H. Qu, Y. Zhu, Y.S. Li and W.L. Wu, Efficient distributed algorithms for toplogy control problem with shortest path constraints, Discrete Mathematics, Algorithms and Applications. **4**, 437–461 (2009).
19. J. Wu, H. Li, On calculating connected dominating set for efficient routing in ad hoc wireless networks. Proc. 3rd ACM Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications. 7–14, (1999)

20. W.L. Wu, H.W. Du, X.H. Jia, Y.S. Li and S.C.-H. Huang, Minimum connected dominating sets and maximal independent sets in unit disk graphs, Theoretical Computer Science. **352**, 1–7 (2006)
21. Z. Zhang, X.F. Gao, W.L. Wu and D.-Z. Du, A PTAS for minimum connected dominating set in 3-dimensionalWireless sensor networks, J. Global Optimization. **45**, 451–458 (2009)

Power Control in Wireless Ad Hoc Networks: Stability and Convergence Under Uncertainties

Themistoklis Charalambous

Abstract A successful distributed power control algorithm requires only local measurements for updating the power level of a transmitting node, so that eventually all transmitters meet their QoS requirements, i.e. the solution converges to the global optimum. There are numerous algorithm which claim to work under ideal conditions in which there exist no uncertainties and the model is identical to the real-world implementation. Nevertheless, the problem arises when real-world phenomena are introduced into the problem, such as uncertainties (such as changing environment and time delays) or the QoS requirements cannot be achieved for all the users in the network. In this chapter, we study some distributed power control algorithms for wireless ad hoc networks and discuss their robustness to real-world phenomena. Simulations illustrate the validity of the existing results and suggest directions for future research.

1 Introduction

Wireless communication is used as a term for transmission of information from one place to another without using cables. This may be one-way communication as in broadcasting systems (such as radio and TV), or two-way communication (e.g. cellular phones). Wireless communication may be via radio frequency (RF) communication, microwave communication or infrared. In wireless networking, in general, radio waves carry the signal over the communication path.

Applications of the wireless technology encompass cellular phones, satellite television, personal digital assistants (PDAs), global positioning systems (GPS)

T. Charalambous (✉)
Automatic control lab, Electrical Engineering Royal Institute of Technology, Stokholm, Sweden
e-mail: themisc@Kth.se

A. Chinchuluun et al. (eds.), *Optimization, Simulation, and Control*, Springer Optimization and Its Applications 76, DOI 10.1007/978-1-4614-5131-0_10,

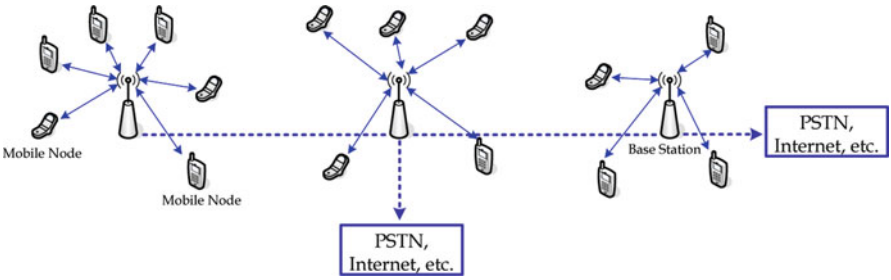


Fig. 1 An example of a cellular topology

units, garage doors, wireless computer equipments (mice, keyboards, printers) and wireless networking.

Even though wireless communication has some disadvantages over the wired communication (security issues and lower data rates), there are plenty of advantages that make them desirable. The advantages of wireless compared to wired communication are: (a) mobility, (b) faster speed of deployment, (c) accessibility of areas that are difficult to reach, and (d) lower cost and effort in adding or removing a subscriber compared to the cost required to install cables for a wired connection.

At the moment, two general types of wireless communication systems have been realised. The first type is cellular systems (e.g. GSM, GPRS, UMTS), based on the fixed infrastructure of base stations. They consist of a number of mobile nodes, and a number of strategically placed immobile base stations that do not have communication needs of their own and exist to serve the communication needs of the mobile nodes. They communicate with each other through high-speed wired or wireless connections to form a *cellular network* of base stations, which is in turn interconnected to the public switched telephone network (PSTN) (as shown in Fig. 1). A mobile node typically only communicates with the base station that lies in its cell, and will have to use that base station to send data both to other networks, such as the PSTN, and to other mobile users that belong to the same network, no matter how close these users are.

The second type is the peer-to-peer (P2P) radio communication systems (e.g. Bluetooth, UWB, ZigBee), where there is no fixed infrastructure. Networks based on P2P radio communication are called *mobile ad hoc networks*. The simplest ad hoc network is a P2P network formed by a set of two nodes within range of each other that dynamically configure themselves to set up a temporary single-hop network. Due to the ease of deployment and a foreseeable wide range of commercial applications, there has been an explosive growth of interest in MANETs. In such networks, nodes communicate with each other without the support of a fixed infrastructure, and each node can act as a source, a destination, or as a relay for the traffic of other nodes (see Fig. 2).

An advantage of MANETs over cellular networks is their flexibility. If a node runs out of battery, or malfunctions, or disappears for some reason, the nodes in its

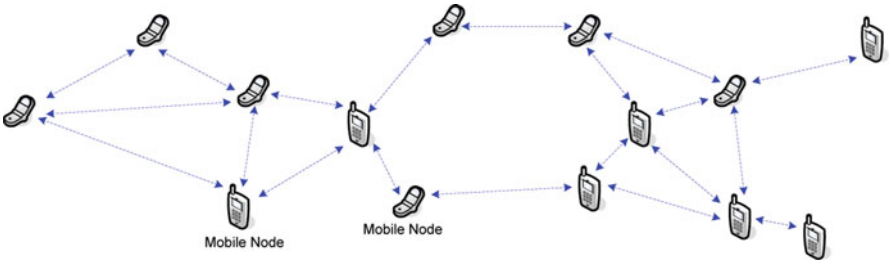


Fig. 2 An example of an ad hoc topology

vicinity will take over its routing responsibilities. On the contrary, if a base station becomes unavailable, then all the nodes in its cell will lose their connection to the network, unless a handoff occurs to another base station, if one happens to be nearby. Another advantage is that MANETs achieve higher resource utilisation; in cellular networks, users that do not have a good wireless link with any base station are either denied service, or the system consumes a lot of resources (bandwidth and energy) to support their operation. On the other hand, in MANETs there are many different paths with which a packet can reach its destination. If the channel link quality between two nodes is not good—provided there are other nodes around to handle the traffic—another route will be utilised. Of course, MANETs cannot be used for far away communication, since that would require a huge number of hops and probably a drop of the signal. A combination of the two types of networks could combine the benefits of each system.

The potential of deployment of wireless ad hoc networks, where infrastructure is either undesirable or infeasible, exists in many scenarios, which among others include battlefield communications [1], disaster recovery efforts [2], interactive information sharing, wireless traffic sensor networks [3], ecological habitat monitoring [4] and industrial process control. In addition, MANETs provide the ability to enhance new applications, alleviate inevitable accidents, anticipate destructive events, as well as observe and understand real-world phenomena. This opportunity for autonomous communication between wireless devices also provides the potential to burgeon new breakthrough scientific advances. Examples of networks deployed that rely on wireless ad hoc networks in order to extend the Internet and/or support well-defined application requirements are *mesh*, *sensor*, *vehicular* and *opportunistic* networks.

Traditionally, network protocols have a strictly layered structure and implement congestion control, routing and scheduling independently at different layers. However, the wireless channel is a shared medium and interference-limited. In order to use the wireless channel more efficiently the interference and contention among links should be exploited. Thus, in wireless networks, there exist issues that naturally span many layers.

Our focus is mainly on the power control algorithm that specifies the power with which signals are transmitted. The problem is more complicated than a simple

tuning problem, since the choice of the power affects many aspects of the operation of the network. In particular, the transmission power [5]:

- Determines the received signal quality, thus affecting the physical layer.
- Determines the range of the transmission, affecting the network layer since the transmission range affects routing.
- Determines the magnitude of the interference, which causes congestion; thus, it affects the transport layer.

Cross-layer design in communication networks, especially in wireless networks, has attracted great attention recently (for an overview, see, e.g. [6,7]). This characteristic of the wireless network should be viewed as an opportunity for a cross-layer design in order to optimise the performance of the network.

In this chapter, we study the stability of some power control algorithms and their robustness to time delays and channel variations, important aspects that are encountered in real-world situations and affect the stability and performance of algorithms. More specifically, we study the conditions for stability of the well-known Foschini–Miljanic algorithm [8] when the topology changes or there exist time-varying delays during the implementation of the algorithm. In addition, the general framework presented by Yates is also presented drawing the parallels to the Foschini–Miljanic algorithm. Furthermore, we show for the first time the similarity of the linear positive systems and the linear power control algorithm presented in this chapter. Thus, we can easily use new results on positive systems in order to study the properties of the FM algorithm.

The remainder of this chapter is organised as follows: In the next section, we present related work in the field, and next, the notation used throughout the chapter is introduced. Then, in Sect. 4, the system model, which comprises the network topology and the channel conditions, is presented. Next, centralised and distributed approaches to the power control problem are presented. In Sect. 7, we show the similarities between linear positive systems and the FM algorithm For this algorithm, conditions for stability with delays (Sect. 8) and changing topology (Sect. 10) are explained. Illustrative examples in Sect. 10 show the validity of the results, and in Sect. 11 useful conclusions about the results presented are drawn. Finally, in Sect. 12, future directions and open problems in the area of power control in wireless ad hoc networks are discussed.

2 Related Work

Since wireless channel is a shared medium, it is limited by interference. Distributed algorithms preferably require no or minimal explicit message passing; since each wireless node has no knowledge of the number of nodes in the network, it is not aware of the action of others a priori and can only get limited information about the channel (interference experienced by its intended receiver). The conventional power control has as objective to meet fixed pre-defined QoS requirements of individual

communication links. This is accomplished by increasing the transmitter power when the link condition is poor. Power is a valuable resource in wireless networks, since the batteries of the wireless nodes have limited lifetime. As a result, power control has been a prominent research area for all kinds of wireless communication networks (e.g. [8–13]). Increased power ensures longer transmission distance and higher data transfer rate. However, power minimisation not only increases battery lifetime, but also the effective interference mitigation that increases the overall network capacity by allowing higher frequency reuse. Adaptive power control in wireless networks allows devices to setup and maintain wireless links with minimum power while satisfying constraints on QoS. Such power control approach is very suitable for services with strict QoS requirements, such as voice and video telephony, with prescribed fixed transmission and bit error rates.

The initial work on power control schemes based on signal-to-interference-and-noise ratio (SINR) has been done by Zander, where a centralised [14] and distributed [9] power control algorithms are presented. It is assumed that the thermal noise is negligible and the power levels of the M transmitters consisting the network are updated to obtain the greatest signal-to-interference-ratio (SIR) that they are capable of jointly achieving. In [14] the algorithm is maximizing the minimum SIR, whereas in [9] Zander proposes the distributed balancing algorithm (DBA) in which the system approaches the target SIR with probability one in a distributed way, if a solution to the system exists. However, Zander's model assumes that there is no thermal noise in the network and finds the maximum SIR for all users. In [8], a power control algorithm is derived (Foschini–Miljanic algorithm) that accounts for the thermal noise and provides power control of wireless ad hoc networks with user-specific SINR requirements. This algorithm converges to the optimal power allocation, if there exists one, by use of local information only; namely, the power and interference measurements of each communication pair are utilised for the update of the power level on that link. If there does not exist a solution to the system, then the algorithm fails and the power levels diverge, i.e. the algorithm converges if there exists a feasible solution to the system and diverges otherwise because of the hard constraint on SIR requirements. Using this algorithm, every user tries to achieve its required SIR value, no matter how high the power consumption is, ignoring the basic fact that power is itself a limited and valuable commodity.

The seminal work of [8] triggered off for numerous publications (e.g. [10, 11, 13, 15–18]) by various authors that extended the original algorithm to account for additional issues, such as constrained power [10] and admission control [16].

An elegant axiomatic framework for studying more general power control iterations was proposed by [11]. The so-called *standard interference functions* include the linear iterations, and several important nonlinear power control laws. Various extensions of the basic framework have been proposed in the literature with the most prominent those by [19, 20].

Recently, Feyzmahdavian et al. [21] explored the connections between the standard interference function framework and the theory for fixed-point iterations. It is shown that interference functions do not define contraction mappings and introduced *contractive interference functions*, that guarantee existence and uniqueness

of fixed-point along with linear convergence of iterates. It is demonstrated that several important distributed power control algorithms proposed in the literature are contractive and derived the associated convergence rates. In some cases, such as linear iterations [8], the convergence rate coincides with known results from the literature that has been obtained using a detailed and tailored analysis. In other cases, such as the utility-based power control [22], the convergence rate is estimated for the first time in the literature. Feyzmahdavian et al. [21] also provided a link between standard interference functions and para-contractions. This result is related to the work by [47], who demonstrated that in logarithmic variables, two-sided scalability implies global Lipschitz continuity of the interference function, and an alternative restriction allows to establish linear convergence rates and uniqueness of fixed-points.

The literature presented in this chapter is by no means exhaustive on the subject. However, the papers cited are representative of the work done in the area of power control under uncertainties in wireless ad-hoc networks.

3 Notation

The sets of complex, real and natural numbers are denoted by $\mathbb{C}$, $\mathbb{R}$ and $\mathbb{N}$, respectively; their positive orthant is denoted by the subscript $+$ (e.g. $\mathbb{C}_+$). Vectors are denoted by bold letters whereas matrices are denoted by capital letters. A^T and A^{-1} denote the transpose and inverse of matrix A, respectively. For two symmetric matrices A and B, $A \succ (\succeq) B$ means that $A - B$ is (semi-)positive definite. By I we denote the identity of a squared matrix. $|A|$ is the elementwise absolute value of the matrix (i.e. $|A| \triangleq [|A_{ij}|]$), and $A(<) \leq B$ is the (strict) elementwise inequality between matrices A and B. A matrix whose elements are nonnegative, called nonnegative matrix, is denoted by $A \geq 0$, and a matrix whose elements are positive, called positive matrix, is denoted by $A > 0$. $\sigma(A)$ denotes the spectrum of matrix A, $\lambda(A)$ denotes an eigenvalue of matrix A, and $\rho(A)$ denotes its spectral radius. $\det(A)$ denotes the determinant of a squared matrix A and $\mathrm{diag}(x_i)$ the matrix with elements $x_1, x_2, \ldots$ on the leading diagonal and zeros elsewhere.

4 Model

The system model can be divided into two levels: the network as a whole and the channel. Thus, we have the network model and the channel model. The network model concerns the general topology of the nodes and their characteristics. The channel model describes the assessment of the link quality between communication pairs and the interaction between the nodes in the network.

4.1 Network Model

In this study, we consider a network where the links are assumed to be unidirectional and each node is supported by an omnidirectional antenna. For a planar network (easier to visualise without loss of generality), this can be represented by a graph $\mathscr{G} = (\mathscr{N}, \mathscr{L})$, where $\mathscr{N}$ is the set of all nodes and $\mathscr{L}$ is the set of the active links in the network. Each node can be a receiver or a transmitter only at each time instant due to the half-duplex nature of the wireless transceiver. Each transmitter aims to communicate with a single node (receiver) only, which cannot receive from more than one node simultaneously. We denote by $\mathscr{T}$ the set of transmitters and $\mathscr{R}$ the set of receivers in the network.

4.2 Channel Model

A transmitted radio signal is an electromagnetic (EM) wave. As with sound waves, electromagnetic waves can be reflected, diffracted and attenuated depending upon the medium and the size (and number) of the obstacles the wave encounters. There exist many phenomena that deteriorate the signal at the receiver, such as noise, interference, multi-path fading, shadowing and attenuation with distance.

The link quality is measured by the SINR. The channel gain on the link between transmitter i and receiver j is denoted by g_{ij} and incorporates the mean path-loss as a function of distance, shadowing and fading, as well as cross-correlations between signature sequences. All the g_{ij}'s are positive and can take values in the range $(0, 1]$. Without loss of generality, we assume that the intended receiver of transmitter i is also indexed by i. The power level chosen by transmitter i is denoted by p_i. ν_i denotes the variance of thermal noise at the receiver i, which is assumed to be additive Gaussian noise. The interference power at the ith node, I_i, includes the interference from all the transmitters in the network and the thermal noise, and is given by

$$I_i = \sum_{j \neq i, j \in \mathscr{T}} g_{ji} p_j + \nu_i. \tag{1}$$

Note that it is implicitly assumed that the interference is a linear combination of all transmitting powers with some given nonnegative coefficients. Thus, the power of all interfering wireless nodes at the receiver is equal to a weighted sum of all transmit power levels with nonnegative weights plus the noise power. The interference is called *an affine interference function* since it is affine in the power vector. Therefore, the SINR at the receiver i is given by

$$\Gamma_i = \frac{g_{ii} p_i}{\sum_{j \neq i, j \in \mathscr{T}} g_{ji} p_j + \nu_i}\ . \tag{2}$$

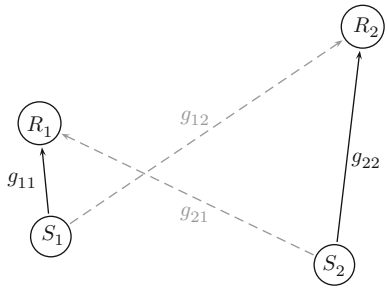


Fig. 3 An example of a network consisting of two communication pairs only. Each pair i consists of a transmitter S_i and a receiver R_i connected with a *solid line* while the *grey dotted arrows* indicate the interference that transmitters cause to the neighbouring receivers

Due to the unreliability of the wireless links, it is necessary to ensure quality of service (QoS) in terms of SINR in wireless networks. Hence, independently of nodal distribution and traffic pattern, a transmission from transmitter i to its corresponding receiver is successful (error-free) if the SINR of the receiver is greater or equal to the *capture ratio* γ_i ($\Gamma_i \geq \gamma_i$) (Fig. 3). The value of γ_i depends on the modulation and coding characteristics of the radio. Therefore,

$$\frac{g_{ii}p_i}{\sum_{j\neq i, j\in\mathscr{T}} g_{ji}p_j + \nu_i} \geq \gamma_i. \tag{3}$$

Inequality (3) depicts the QoS requirement of a communication pair i while transmission takes place. After manipulation it becomes equivalent to the following:

$$p_i \geq \gamma_i \left(\sum_{j\neq i, j\in\mathscr{T}} \frac{g_{ji}}{g_{ii}} p_j + \frac{\nu_i}{g_{ii}} \right). \tag{4}$$

In matrix form, for a network consisting of n communication pairs, this can be written as

$$\mathbf{p} \geq \Gamma G\mathbf{p} + \eta, \tag{5}$$

where $\Gamma = \mathrm{diag}(\gamma_i)$, $\mathbf{p} = \begin{pmatrix} p_1 & p_2 & \dots & p_n \end{pmatrix}^T$, $\eta_i = \frac{\gamma_i \nu_i}{g_{ii}}$ and

$$G_{ij} = \begin{cases} 0 & , \text{if } i = j, \\ \frac{g_{ji}}{g_{ii}} & , \text{if } i \neq j. \end{cases} \tag{6}$$

Let

$$C = \Gamma G, \tag{7}$$

so that (5) can be written as

$$(I - C)\mathbf{p} \geq \eta, \tag{8}$$

The matrix C has nonnegative elements and it is reasonable to assume that is irreducible, since we are not considering totally isolated groups of links that do not interact with each other. By the Perron–Frobenius Theorem [23], we have that the spectral radius of the matrix C is a simple eigenvalue, while the corresponding eigenvector is positive componentwise. The necessary and sufficient condition for the existence of a nonnegative solution to inequality (8) for every positive vector η is that $(I-C)^{-1}$ exists and is nonnegative. However, $(I-C)^{-1} \geq 0$ if and only if $\rho(C) < 1$ [24] (*Theorem 2.5.3*), [25], where $\rho(C)$ denotes the spectral radius of C.

Remark 1. A sufficient condition to establish stability to the system without requiring the knowledge of the whole matrix C, could be $\| C \|_\infty < 1$, i.e.

$$\frac{g_{ii}}{\sum_{j\neq i, j\in\mathcal{T}} g_{ji}} > \gamma_i \quad \forall\, i. \tag{9}$$

Since $\rho(C) \leq \| C \|_\infty$, this condition is more conservative. This condition is equivalent to H being a diagonally dominant matrix with all main diagonal entries being positive. Hence, this guarantees that all the eigenvalues of matrix H have positive real part, [23]. It, therefore, provides an upper bound on the achievable target SINR levels in a given network, and hence, leads to a soft capacity constraint for the underlying system. The return for this conservatism is that the only extra information required at each transmitter is a measure of the sum of the channel gains at its receiver by all other transmitters. Hence, we are able to use a distributed way of updating the desired SINR levels and keep the network functioning. In case a communication pair cannot reach its desired SINR and cannot be compromised by a lower SINR level, then the transmitter may wish to either back-off until condition (9) is satisfied for a reasonable SINR level, or go closer to the receiver, if possible (i.e. increase g_{ii}).

Definition 1 ((Feasibility)[26]). A set of target SINRs Γ_i is said to be feasible with respect to a network, if it is possible to assign transmitter powers $p_i \geq 0$ so that the requirement in inequality (3) is met for all nodes transmitting in the network. Analogously, the power control problem is said to be feasible under the same conditions. Otherwise, the target SINRs and the power control problem are said to be infeasible.

Remark 2. We have not specified any model for determining the positions of the nodes, since we investigate the general case of a network that any position could be possible. We have also not specified any model related to the propagation of signals. In our context, these two models will ultimately specify the channel gains, g_{ij}. Nevertheless, they are of secondary importance in this study since it is focused on how the QoS is improved given the channel gains and it only depends on the power of the received signals.

Remark 3. Note that the effect of nodes' mobility is not considered in this study. However, this could be relaxed to the case of low mobility, where the link structure is expected to change slowly with respect to the packet rates and network updates.

5 Centralised Power Control

When considering centralised power control, it is assumed that all channel gains are available to a central station and a power control optimisation is conducted centrally. One example is provided in this section. The aim is to minimise the overall transmitted power under the constraint that each wireless node maintains its SINR above the desired SINR. Hence, the optimisation problem is given by the following:

Model 1 Mathematical formulation: power minimisation

Minimise

$$\sum_{i\in\mathcal{T}} p_i \tag{10}$$

subject to:

$$\mathbf{p} \geq C\mathbf{p} + \eta \tag{11}$$

$$\mathbf{p} > 0. \tag{12}$$

If the spectral radius of matrix C is less than unity, then $(I - C)$ is invertible and positive [23]. Therefore, the network is feasible, and the optimal solution is given by

$$\mathbf{p}^* = (I - C)^{-1}\eta. \tag{13}$$

As aforementioned, in order to find the optimal solution in a centralised manner, all the channel gains are known to a central station. However, this is impossible in wireless ad hoc networks due to the nature of their deployment and operation. Furthermore, even if it was possible to gather this information, the network changes continuously due to mobility, and in large networks it would be impossible to collect all the information centrally continuously and calculate the optimal power in real time, since the computational complexity and channel estimation overheads increase rapidly with the number of nodes in the network.

In the next section, some algorithms are presented in which the defined centralised optimisation problem (5) is solved in a distributed fashion. Indeed, not only the power control algorithm is able to find a feasible solution, but it converges to the optimal solution.

6 Distributed Power Control

In this section, the most prominent power control algorithms are reviewed; namely, the Foschini–Miljanic (FM) algorithm and later a general framework in which the FM algorithm belongs.

6.1 Review of the Foschini–Miljanic Algorithm

The Foschini–Miljanic algorithm [8] is a distributed algorithm where a transmitter uses only information about the interference the intended receiver experiences. It succeeds in attaining the required SINRs for all nodes in the network if a solution exists and fails if there does not exist a solution.

6.1.1 The Continuous-Time Algorithm

The following differential equation is defined in [8] in order to model the continuous-time power dynamics:

$$\frac{\mathrm{d}p_i(t)}{\mathrm{d}t} = k_i \left(-p_i(t) + \gamma_i \left(\sum_{j \neq i, j \in \mathcal{T}} \frac{g_{ji}}{g_{ii}} p_j(t) + \frac{\nu}{g_{ii}} \right) \right), \tag{14}$$

where $k_i \in \mathbb{R}$, $k_i > 0$, denotes the proportionality constant, g_{ji} denotes the channel gain on the link between transmitter j and receiver i and γ_i denotes the desired SINR. It is assumed that each transmitter i has knowledge of the interference at its receiver only,

$$I_i(t) = \sum_{j \neq i, j \in \mathcal{T}} \frac{g_{ji}}{g_{ii}} p_j(t) + \frac{\nu}{g_{ii}}.$$

In matrix form this is written as

$$\dot{\mathbf{p}}(t) = -KH\mathbf{p}(t) + K\eta, \tag{15}$$

where $K = \mathrm{diag}(k_i)$ and

$$H_{ij} = \begin{cases} 1 & , \text{if } i = j, \\ -\gamma_i \frac{g_{ji}}{g_{ii}} & , \text{if } i \neq j. \end{cases} \tag{16}$$

For this differential equation, it is proved that the system will converge to the optimal set of solutions, $\mathbf{p}^* > 0$, for any initial power vector, $\mathbf{p}(0) > 0$. Therefore, the

distributed algorithm (14) for each communication pair leads to global stability of the distributed system. Note that, at $p_i(t) = 0$, from (14), $\mathrm{d}p_i(t)/\mathrm{d}t > 0$ restricting the power to be nonnegative, thus fulfilling the physical constraint that the power $p_i \geq 0$. Hence, we should not worry about saturation issues in the system.

Remark 4. Since H is an M-matrix, the system is D-stable, and therefore there exists diagonal matrix D with positive entries such that $DH + H^T D \succ 0$. Pre-multiplying by M^T and post-multiplying by M, where M is a diagonal matrix with positive entries, then

$$M^T(DH + H^T D)M \succ 0.$$

Therefore,

$$MDHM + (HM)^T DM \succ 0 \Rightarrow EHM + (HM)^T E \succ 0.$$

Thus, matrix H is scaled and its stability is not affected, as a consequence of the diagonal stability property of matrix H. That is why, in the power update formula (14), any positive gain guarantees stability of the system. More details on D-stability can be found in the Appendix.

6.1.2 The Discrete-Time Algorithm

As in [8], in the discrete time, we define the time coordinate so that unity is the time between consecutive power vector iterations. In correspondence with the differential equation (15), the discrete-time Foschini–Miljanic algorithm is written as in [8],

$$\mathbf{p}(n+1) - \mathbf{p}(n) = -KH\mathbf{p}(n) + K\eta. \tag{17}$$

The distributed power control algorithm is then given by

$$p_i(n+1) = (1-k_i)p_i(n) + k_i\gamma_i\left(\sum_{j\neq i, j\in\mathscr{T}} \frac{g_{ji}}{g_{ii}}p_j(n) + \frac{\nu}{g_{ii}}\right). \tag{18}$$

It has been shown that whenever a centralised "genie" [8, 27] can find a power vector, $\mathbf{p}^*$, meeting the desired criterion, then so long as the proportionality constant (k_i) is appropriately chosen $(k_i \in (0,1])$, then the iterative algorithm (18) converges from any initial values for the power levels of the individuals transmitters. Note that, since $k_i \leq 1$, from (17) it is obvious that $p_i(n+1)$ is always nonnegative. Thus, the physical constraint that the power $p_i \geq 0$ is fulfilled.

Theorem 1. *If the spectral radius of matrix C in (7) is less than* 1*, then the continuous-time Foschini–Miljanic power control algorithm*

$$\frac{\mathrm{d}p_i(t)}{\mathrm{d}t} = k_i\left(-p_i(t)+\gamma_i\left(\sum_{j\neq i, j\in\mathscr{T}}\frac{g_{ji}}{g_{ii}}p_j(t)+\frac{\nu}{g_{ii}}\right)\right),\ i\in\mathscr{T},$$

and the discrete-time Foschini–Miljanic algorithm

$$p_i(n+1) = (1-k_i)p_i(n)+k_i\gamma_i\left(\sum_{j\neq i, j\in\mathscr{T}}\frac{g_{ji}}{g_{ii}}p_j(n)+\frac{\nu}{g_{ii}}\right),$$

for γ_i, g_{ji}, $\nu > 0$*, are asymptotically stable for any initial state* $p_i(0) > 0$ *and for any proportionality constant,* $k_i > 0$ *in the continuous-time FM algorithm, and* $0 < k_i < 1$ *in the discrete-time FM algorithm.*

6.2 Review of Yates' Framework

Although affine interference functions are the most common ones, they are not the only ones that can be encountered in real-world networks. Hence, a more general framework of interference functions is introduced here.

Now the interference function for node $i \in \mathscr{R}$, $I_i : \mathbb{R}^n \to \mathbb{R}$ is any standard interference function that fulfills the following axioms:

Definition 2 (Standard Interference Function [11]). We say that $I_i : \mathbb{R}^n \to \mathbb{R}$ is a standard interference function if each of the following holds:

1. $I_i(\mathbf{p}) > 0$ for all $\mathbf{p} \geq 0$ (*positivity*).
2. $I_i(\alpha\mathbf{p}) < \alpha I(\mathbf{p})$ for any $\mathbf{p} \geq 0$ and $\alpha > 1$ (*scalability*).
3. $I_i(\mathbf{p}_{(1)}) \geq I(\mathbf{p}_{(2)})$ if $\mathbf{p}_{(1)} \geq \mathbf{p}_{(2)}$ (*monotonicity*).

The SINR is hence given by

$$\mathrm{SINR}_i(\mathbf{p}) = p_i/I_i(\mathbf{p}) \geq 0\,.$$

It may be verified that the affine interference function assumed in [8] and given by

$$R_i(\mathbf{p}) = \sum_{j\neq i, j\in\mathscr{T}}\frac{g_{ji}}{g_{ii}}p_j + \frac{\nu_i}{g_{ii}} \tag{19}$$

satisfies the axioms.

Now the power control problem is reduced to finding a power vector $\mathbf{p}$ satisfying $\mathbf{p} \geq I(\mathbf{p})$, which expresses the fact that the transmitted power must overcome the interference.

6.2.1 The Continuous-Time Algorithm

In the continuous-time algorithm, by replacing $\gamma_i R_i(\mathbf{p})$ in the FM algorithm with the general form of interference function $I_i(\mathbf{p})$, the differential equation becomes

$$\frac{\mathrm{d}p_i(t)}{\mathrm{d}t} = k_i\left(-p_i(t) + I_i(\mathbf{p}(t))\right). \tag{20}$$

6.2.2 The Discrete-Time Algorithm

The iteration function in the general form is called the *standard power-control algorithm*, and it is given by

$$P_i(n+1) = I_i(\mathbf{p}(n)). \tag{21}$$

The general nonlinear representation of the interference function allows for constraints on the power levels of the wireless devices, as it holds in real world. For example, if there exists a maximum power only then $I_M(\mathbf{p}) = \min\{p_{\max}, I(\mathbf{p})\}$, or a minimum power only, then $I_m(\mathbf{p}) = \max\{p_{\min}, I(\mathbf{p})\}$. If there exist constraints on both maximum and minimum power, then $I_c(\mathbf{p}) = \max\{p_{\min}, I_M(\mathbf{p})\}$ (equivalently $I_c(\mathbf{p}) = \min\{p_{\max}, I_m(\mathbf{p})\}$).

It should be noted that if the power vector is positive at the initial state,$\mathbf{p}(t_0) > 0$, then it remains positive for all times $t > 0$, $\mathbf{p}(t) > 0$, for both the continuous-and discrete-time algorithms even in the presence of delays.

Theorem 2. *If the algorithm (continuous or discrete) has a fixed convergent point, then that fixed point is unique. The continuous-time algorithm*

$$\frac{\mathrm{d}p_i(t)}{\mathrm{d}t} = k_i\left(-p_i(t) + I_i(\mathbf{p}(t))\right),$$

and discrete-time algorithm

(continued)

(continued)

$$P_i(n+1) = I_i(\mathbf{p}(n)),$$

converge to that fixed point $\mathbf{p}^*$ *for any initial power vector* $\mathbf{p}$ *when operating in both synchronous and asynchronous modes.*

7 Power Control in Wireless Networks and Positive Systems

Firstly, the concept of positive systems is introduced, and then we show the relation between positive systems and power control algorithms.

Definition 3. A system is called positive if, for a positive initial condition, all its states remain in the positive orthant throughout the time.

It means that the states and outputs of positive systems are nonnegative whenever the initial conditions and inputs are nonnegative. The states of positive systems are confined within a cone located in the positive orthant rather than on the whole space $\mathbb{R}^n$ [28]. In general, a continuous system,

$$\dot{x}(t) = f(x(t)), \quad x(0) = x_0,$$

is positive, if $\mathbf{R}^n_+$ is forward invariant ($x_0 \geq 0$ implies $x(t, x_0) \geq 0$ for all $t \geq 0$).

For example, the linear time-invariant (LTI) system $\dot{\mathbf{x}} = A\mathbf{x}$ is said to be positive if $\mathbf{x}(0) = \mathbf{x}_0 \geq 0$ implies that $\mathbf{x}(t) \geq 0$ for all $t \geq 0$. An LTI system is positive, if and only if, matrix A is an M-matrix [29]. By means of a linear co-positive Lyapunov function that captures the properties of positive systems, [30] proposed necessary and sufficient conditions for stability of such systems, triggering off for further research in positive systems, such as [31, 32]. The linear function

$$V(\mathbf{x}) = \mathbf{v}^T\mathbf{x}$$

defines a linear co-positive Lyapunov function for the positive LTI system $\dot{\mathbf{x}} = A\mathbf{x}$, if and only if the vector $\mathbf{v} \in \mathbb{R}^n$ satisfies $\mathbf{v} > 0$ and $A^T\mathbf{v} < 0$.

By the definition of positive systems, we can easily deduce that both the FM algorithm (both continuous and discrete) and Yates' framework constitute a positive system, since the power levels remain positive throughout the operation of the network. The FM algorithm constitutes a linear positive system. Hence, we are able to use existing results and properties to study the stability and performance of the FM algorithm. Similarly, the general framework introduced also fulfils the conditions for a positive system.

8 On the Stability of Power Control Algorithms with Delays

In [33] by using the multivariate Nyquist criterion [34] and by determining the set in which the spectrum of the multivariate system lies, we prove that both the continuous-and discrete-time FM algorithms are GAS for arbitrarily large constant time delays. Note that stability of the discrete-time algorithm in the presence of delays (constant or time-varying) is stated in [15], where he proves that the system converges under asynchronous operation. These results indicate that the FM algorithm, compared to other power control algorithms (e.g. example [35]), is suitable to be used in any network without requiring any bound on constant time delays. Making the connection with positive systems, stability of the continuous-time Foschini–Miljanic algorithm with constant time delays is guaranteed by a linear co-positive Lyapunov function, as proposed in [36].

The robustness of the algorithm in the case where there exist time-varying delays was also studied. Firstly in [37], a dependent of delays (DoD) approach was adopted and derived the stability conditions for which the system is stable by proposing a Lyapunov-Krasovskii functional in the form of a linear matrix inequality (LMI) [38]. It is an effective and practical methodology providing LMI conditions which can be solved efficiently with semi-definite optimisation solvers in a polynomial time ensuring the global stability of the wireless network. Numerical examples though showed that the Foschini–Miljanic power control algorithm is able to converge to the optimal vector of powers even in cases of time-varying delays, when the nodes adjust their proportionality constants (k_i) accordingly. Next, in [39], an independent of delays (IoD) stability condition for the FM algorithm under time-varying delays is presented. The functional proposed is the classical Lyapunov-Krasovskii which provides IoD condition. It is proven that the continuous-time FM algorithm is asymptotically stable whatever the delay introduced into the network, provided that the delay derivative is less than one. Hence, the nodes can arbitrarily choose the positive proportionality gains, k_i, they wish throughout the operation of the network, and no communication is required in the network, maintaining the fully distributed nature of the FM algorithm. Again, making the connection with positive systems, stability of the FM algorithm for Lebesgue measurable, time-varying but bounded delays can be deducted by [40] in which the authors find the stability conditions for continuous-time linear positive systems with time-varying delays and changing topologies. For the discrete-time FM algorithm, many approaches prove the algorithm's stability in the presence of time-varying delays; for example, [11,28] prove asymptotic stability, whereas [15, 21, 41] prove convergence with linear convergence rates.

The following theorem summarises the results on the stability of the FM algorithm with delays:

Theorem 3. *If the spectral radius of matrix C in (7) is less than* 1 *and the delay is Lebesgue measurable and bounded, then the continuous-time Foschini–Miljanic power control algorithm with time-varying delays*

$$\frac{\mathrm{d}p_i(t)}{\mathrm{d}t} = k_i\left(-p_i(t)+\gamma_i\left(\sum_{j\neq i, j\in\mathscr{T}}\frac{g_{ji}}{g_{ii}}p_j(t-T_i(t))+\frac{v}{g_{ii}}\right)\right),\ i\in\mathscr{T},$$

and the discrete-time Foschini–Miljanic algorithm with time-varying delays

$$p_i(n+1) = (1-k_i)p_i(n)+k_i\gamma_i\left(\sum_{j\neq i, j\in\mathscr{T}}\frac{g_{ji}}{g_{ii}}p_j(n-T_i(n))+\frac{v}{g_{ii}}\right),$$

for γ_i, g_{ji}, $v>0$, *are asymptotically stable for arbitrarily large time-varying delays,* $T_i(t), T_i(n)>0$, *for any initial state* $p_i(0)>0$ *and for any proportionality constant,* $k_i>0$ *in the continuous-time FM algorithm, and* $0<k_i<1$ *in the discrete-time FM algorithm.*

Yates [11] showed that if an iteration involving standard interference function converges synchronously, it also converges when executed totally asynchronously. A similar result holds for contracting interference functions [21].

9 On Topology Changes in Wireless Networks and Stability

Due to mobility and the dynamic environmental changes, there exist cases for which channel variability time and network updates scales are similar. In addition, the network links may change even for stationary users depending on the network demands. For these reasons, it is important to find the conditions for which the network is feasible, and hence the power control algorithm is stable throughout the changes.

Proposition 1 ([42]). *If the network as a system is feasible for the worst case where the link assignment causes maximum interference in all wireless receivers, then the FM algorithm is stable for all time-variations as well.*

For the proof of the proposition, we need the following result [23](*Theorem 8.1.18*):

Theorem 4. *Let $A \in \mathbb{C}^{N\times N}$ and $B \in \mathbb{R}^{N\times N}$, with $B \geq 0$. If $|A| \leq B$, then*

$$\rho(A) \leq \rho(|A|) \leq \rho(B).$$

Proof. For the worst-case scenario, matrix C_{worst} that characterises the network is a non-negative matrix that satisfies

$$C \leq C_{\text{worst}},$$

for all possible network configurations. That is, since the off-diagonal entries of the matrix depict the interference a node experiences from all other nodes in the network, then in the worst case scenario, matrix C_{worst} is bigger entry-wise than any other matrix C. Since the system is stable for the worst-case, then $\rho(C_{\text{worst}}) < 1$. Therefore, from Theorem 4,

$$\rho(C) \leq \rho(C_{\text{worst}}) < 1.$$

As proven in [43], as long as the spectral radius of matrix C remains less than 1 in changing topology, then the FM power control algorithm is stable. □

Secondly, we demonstrate the condition for which a network is stable under channel-varying conditions. The spectral radius of a real matrix A is defined by

$$\rho(A) := \max\{|\lambda| : \lambda \text{ is an eigenvalue of A}\}.$$

Using the identity (see, e.g. [23])

$$\rho(A) = \lim_{k\to\infty} \|A^k\|^{1/k},$$

the spectral radius to a set of matrices is generalised. Hence, for a set of matrices $\Sigma = \{A_1, \ldots, A_s\} \in \mathbb{R}^{n\times n}$, the joint spectral radius $\bar{\rho}(\Sigma)$ is defined by [44]

$$\bar{\rho}(\Sigma) = \limsup_{k\to\infty} \bar{\rho}_i(\Sigma),$$

where $\bar{\rho}_i(\Sigma) = \sup\{\|A_{i_1}A_{i_2}\ldots A_{i_k}\|^{1/k} \; \forall A_i \in \Sigma\}$ for $k \geq 1$.

In the case of changing topology the convergence condition is that the joint spectral radius $\bar{\rho}(\Sigma)$ is smaller than 1, i.e.,

$$\bar{\rho}(\Sigma) = \lim_{k\to\infty} \|A_1A_2\ldots A_{k-1}A_k\|^{1/k}.$$

Since the logarithm of the joint spectral radius coincides with the Lyapunov exponent, this is equivalent to requiring the Lyapunov exponent λ_F to be negative, where

$$\lambda_F = \lim_{k\to\infty} \frac{1}{k} \log \|A_1 A_2 \dots A_{k-1} A_k\|,$$

In [45] it is proven that the condition of stability is that the Lyapunov exponent is negative, if and only if Σ is a stationary ergodic sequence of random matrices. In [46], they lift this assumption, and they show that the stability condition is purely deterministic and is equivalent to $\bar{\rho}(\Sigma) < 1$.

Summarising the existing results on changing topology in wireless ad-hoc networks we can state the following theorem:

Theorem 5. *If the network as a system is feasible for the worst case where the link assignment causes maximum interference in all wireless receivers, then the FM power control algorithm is stable for all time variations of the network as well. Otherwise, for switching network topologies, if the joint spectral radius $\bar{\rho}(\Sigma)$ is smaller than 1, i.e.,*

$$\bar{\rho}(\Sigma) = \lim_{k\to\infty} \|A_1 A_2 \dots A_{k-1} A_k\|^{1/k},$$

then the FM power control algorithm is stable.

In [40], the authors consider both the effects of time-varying delays and changing network topologies. They provide a new theoretical result concerning the stability of such positive systems, which they use to show that the Foschini–Miljanic algorithm is globally asymptotically stable even under those harder, more realistic conditions. These results are of practical importance when designing wireless networks in changing environments with communication delays, as is typically the case for CDMA networks.

10 Illustrative Examples

In this section, the stability of the Foschini–Miljanic algorithm is studied via illustrative examples.

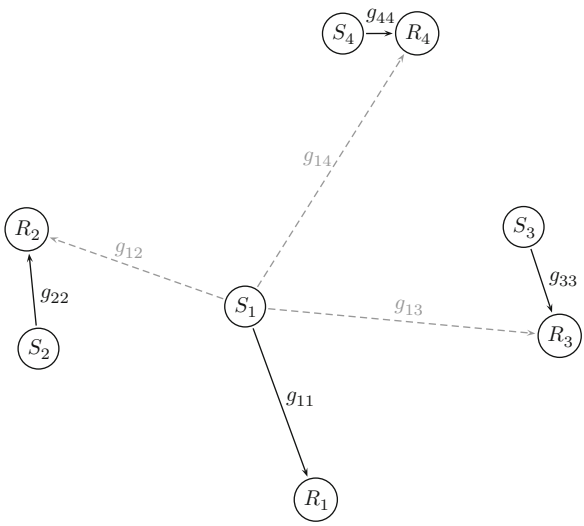


Fig. 4 Example of a wireless ad-hoc network of $n = 8$ nodes, consisting of four communication pairs $\{S_i \to R_i\}$. The *grey dotted arrows* are included to indicatively show the interference caused to the receivers by S_1

10.1 Power Control with Constant Delays

Consider an ad-hoc network consisting of four communicating pairs, i.e. eight mobile devices in total. For this example we have that $\gamma_i = 3$ and $\nu = 0.04$ W. The initial power $p_i(0)$ for each transmitter is 1 W. The network is described by matrix C and it is schematically shown in Fig. 4.

$$C = \begin{pmatrix} 0 & 0.5405 & 0.3880 & 0.1131 \\ 0.2143 & 0 & 0.0101 & 0.0323 \\ 0.0522 & 0.0070 & 0 & 0.0271 \\ 0.0084 & 0.0016 & 0.0385 & 0 \end{pmatrix}.$$

For this setup, the Perron–Frobenius eigenvalue of C is 0.3759, so the power control algorithm is stable, even though $\| C \|_\infty > 1$. This is illustrated in the top figure (Fig. 5) for the continuous-time FM algorithm. For the same network, utilising the discrete-time FM algorithm, the system is asymptotically stable, provided the proportionality constant is appropriately chosen such that $k_i \in (0, 1]$. This is demonstrated in the bottom figure (Fig. 5) for a proportionality constant $k_i = 1$ and different time delays for each communication pair.

In a distributed implementation of the algorithm where all transmitters satisfy $\rho(C) < 1$, assuming that the nodes acquire the information required for updating their desired SINRs, the first communicating pair has to reduce the data rate, and hence require smaller SINR, such that

$$\sum_j C(i, j) < 1, \text{i.e., } \gamma_1 < 2.8802.$$

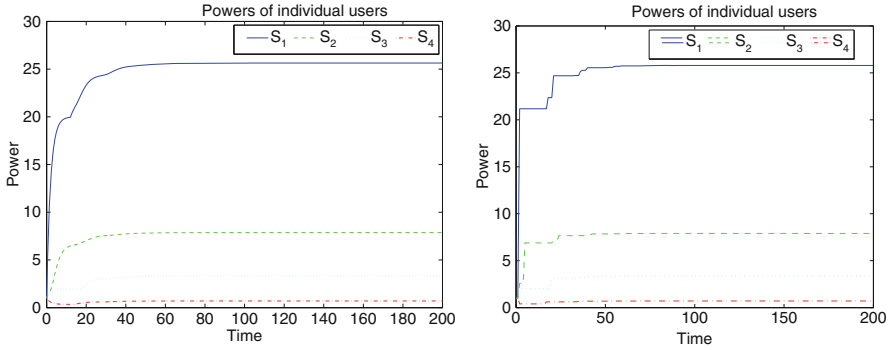


Fig. 5 Continuous and discrete time FM algorithm with delays ($T = \{15, 2, 17, 14\}$). The algorithm asymptotically converges to the desired SINR in a distributed manner

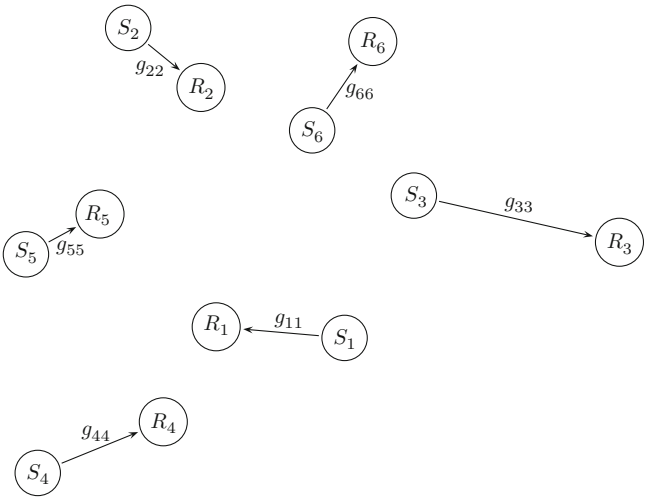


Fig. 6 Example of a wireless ad-hoc network of $n = 12$ nodes, consisting of six communication pairs $\{S_i \to R_i\}$. Interference caused is not depicted in the figure

10.2 Power Control with Time-Varying Delays

We consider a wireless network with six communicating pairs (shown in Fig. 6) characterised by matrix (22):

$$C_2 = \begin{bmatrix} 0 & 0.0414 & 0.2074 & 0.2925 & 0.3998 & 0.1345 \\ 0.0159 & 0 & 0.0506 & 0.0043 & 0.0422 & 1.164 \\ 0.7335 & 0.0626 & 0 & 0.0364 & 0.0477 & 0.4231 \\ 0.6359 & 0.0222 & 0.0644 & 0 & 0.3283 & 0.0447 \\ 0.0227 & 0.0536 & 0.0155 & 0.0215 & 0 & 0.0407 \\ 0.0228 & 0.1114 & 0.2458 & 0.0030 & 0.011 & 0 \end{bmatrix}. \tag{22}$$

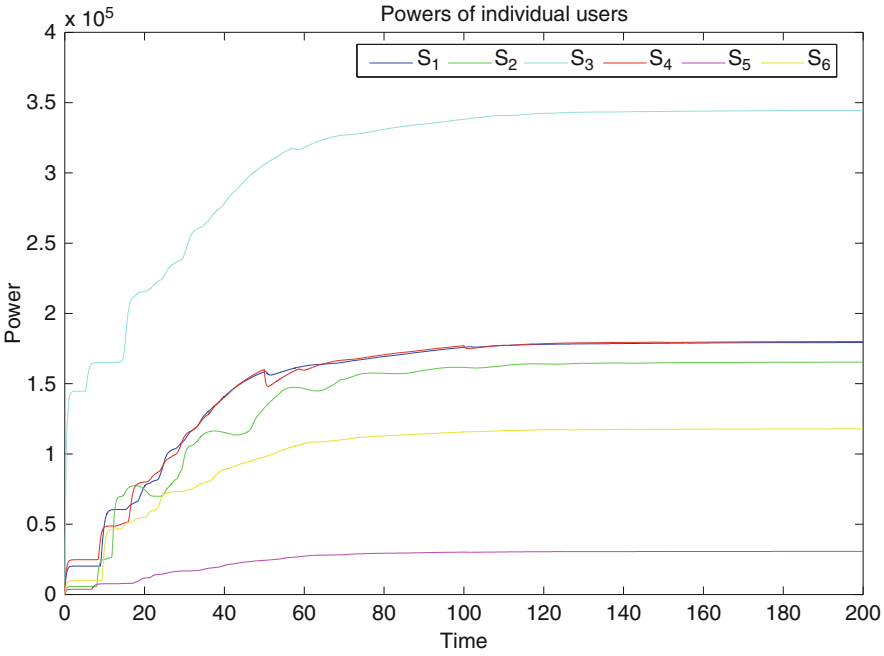


Fig. 7 Simulation of the network represented in Fig. 6. Power levels converge to the desired SINR and to the minimal power vector for different signal generators (sine, sawtooth and steps with different frequencies)

For this example, the SINR threshold and the thermal noise for each node are again set to $\gamma_i = 3$ and $v = 0.04\,\text{mW}$, respectively. The initial power $p_i(0)$ for all transmitters is set to 1 mW. In this case, the time-varying delays are bounded. For comparison again, we set the maximum delay to 10 s for all users (Fig. 7).

As proven earlier, the FM algorithm is asymptotically stable for arbitrarily large time-varying delays and delay derivatives. In this example, the maximum delay is 10 s and the proportionality gain is equal to 3, for all users in the network. The time-varying delays between the different pairs have been simulated with different signal generators (sine, sawtooth and steps with different frequencies).

10.3 Quantitative Analysis on the Relation Between the Convergence Rate and Delays

In this section, we provide a quantitative analysis on how the convergence rate of the system changes with delays. We divide the study into two subsections. In the first one, we consider constant delays only, whereas in the second one, we consider

time-varying delays. For convenience, we use the same maximum delay for all the users in the network, and we observe how the convergence rate changes with the maximum delay.

10.3.1 Constant Time-Delays

We again use the network shown in Fig. 6 and we vary the delay of all the users in the network. We use the same delay for all users in the network since we want to study the worst rate of convergence for the whole system, and hence, it is the same as all having the maximum delay in the network. We vary the delay from 1 to 20 s and we observe the convergence rate. The proportionality gain is kept constant and equal to 0.1. For larger gains the rate of convergence is higher, but more oscillatory. The results from the simulations are shown in Fig. 8, where the rate of convergence seems to approximately follow a linear relationship with time delays. Therefore, in practice the convergence time does not grow much faster than the communication delays, as expected, supporting the significance of the results derived.

10.3.2 Time-Varying Delays

For the same network we observe the convergence for time-varying delay. The maximum delay is the same for all users and the proportionality constant is again equal to 0.1. The results of our simulations are shown in Fig. 9, and it is evident that the convergence rate decreases approximately linearly with the time-varying delay. Further, an analytical approach could reveal the exact relationship or, at least, lower bounds on the convergence rate of the system.

10.4 Power Control with Switching Topologies

We consider three network configurations described by the following matrices:

$$C_1 = \begin{bmatrix} 0 & 0.35 & 0.45 \\ 0.12 & 0 & 0.05 \\ 0.04 & 0.23 & 0 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 0 & 0.15 & 0.15 \\ 0.40 & 0 & 0.20 \\ 0.70 & 0.12 & 0 \end{bmatrix}$$

$$C_3 = C_{\mathrm{w}} = \begin{bmatrix} 0 & 0.37 & 0.45 \\ 0.40 & 0 & 0.27 \\ 0.70 & 0.23 & 0 \end{bmatrix}, \quad \rho(C_{\mathrm{w}}) = 0.8136 < 1.$$

For this example, the SINR threshold and the thermal noise for each node are again set to $\gamma_i = 3$ and $\nu_i = 4 \times 10^{-5}$ W, respectively. The initial power $p_i(0)$ for all

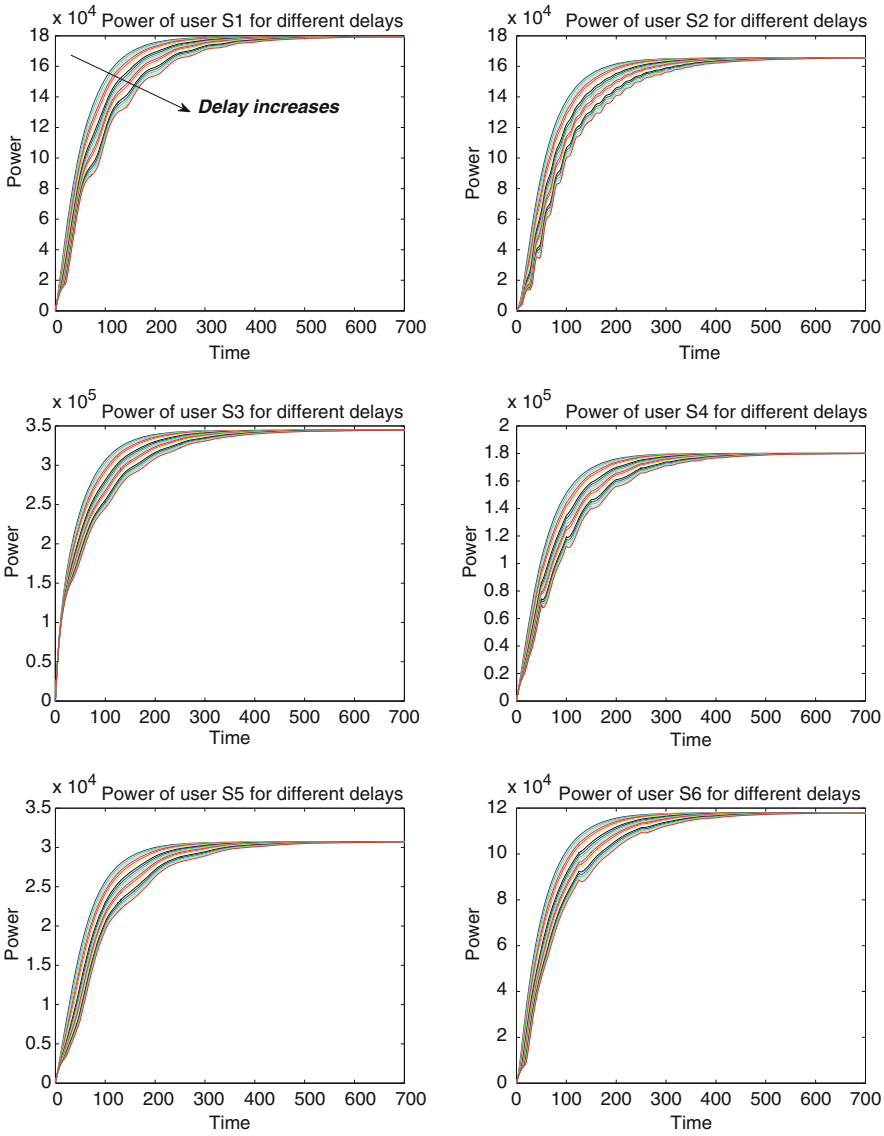


Fig. 8 In this figure, we observe the convergence rate of the FM algorithm in the presence of constant time delays for each of the users in the network shown in Fig. 6. The delay varies from 1 to 20

transmitters is set to 1 mW. The switching sequence between the different network configurations is arbitrary, and it is shown at the top in Fig. 10.

The fact that there exist oscillations is due to the switching between different equilibria, since each topology requires convergence to a different equilibrium point.

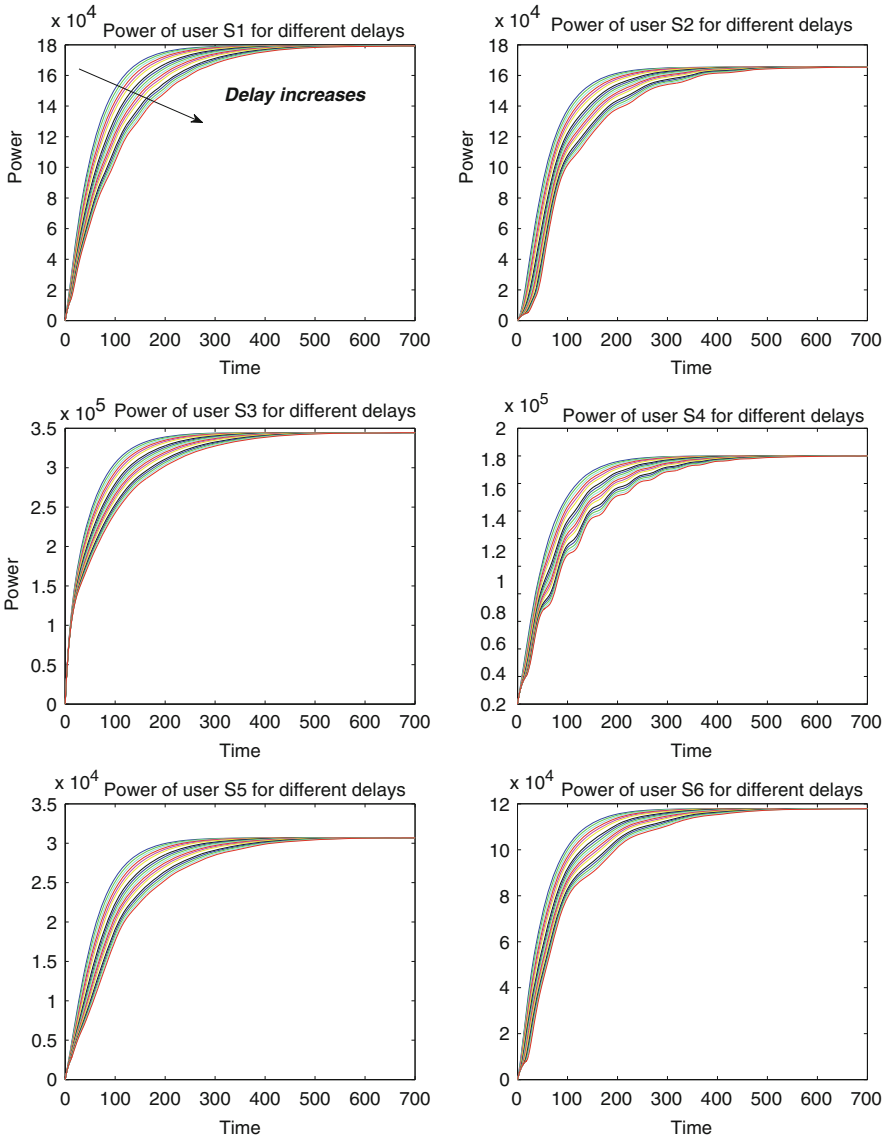


Fig. 9 In this figure we observe the convergence rate of the FM algorithm in the presence of time-varying delays for each of the users in the network shown in Fig. 6. The delay varies from 1 to 20

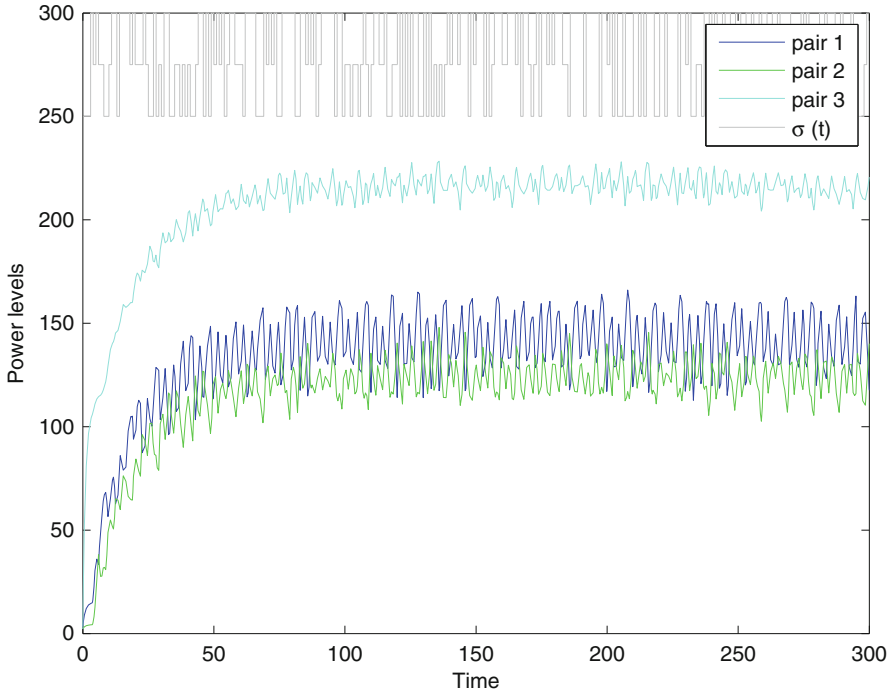


Fig. 10 Simulation of the switching networks represented by matrices C_1, C_2 and C_3, each consisting of three communication pairs. The switching between the matrices is arbitrary. $\sigma(t)$ shows the switching between the matrices. At the lowest level (250) is at C_1, at the middle (275) is at C_2 and at the top (300) is at C_3

11 Conclusions

In this chapter, we focused on power control in an environment where there exist time-varying delays within the communicating pairs or changing topologies. We find the stability condition for the FM algorithm under no, constant and time-varying delays. We illustrate that the FM algorithm is asymptotically stable whatever the delay introduced into the network, provided that the system is stable when no delays are present in the network. Hence, the nodes can arbitrarily choose the positive proportionality gains, k_i, they wish throughout the operation of the network, and no communication is required in the network, maintaining the fully distributed nature of the FM algorithm. From the simulations, the validity of the theoretical results are demonstrated. We also refer to the corresponding results regarding the power control algorithm under Yates' framework.

12 Current Problems and Research Directions

In this work, we have not considered delays on the current value of a node's state, which is not always the case since filtering and control signalling introduce delays to wireless nodes' state. However, stability and convergence properties of the power control algorithms are affected. Power control laws of higher order to include models with delays and delay compensation have been studied in [47, 48] for the discrete time only, in which more structure of the interference feedback is exploited in order to find conditions for stability and convergence. Towards this direction, other methods should be found to resolve the conservativeness of the current results.

Even though stability has been studied for networks with delays and for changing/switching topology nobody studied the effect of those to effects combined. That is, what are the stability conditions when there exist both time-varying delays and the network's topology is changing?

In view of the proliferation of wireless data though, it is essential to investigate further transmission schemes, i.e., techniques that facilitate elastic and/or opportunistic traffic should be considered, where time-varying rates are allowed and larger time delays can be tolerated. Some schemes tried to deal with the case where the desired SINR could be adaptive, depending on the channel conditions. The fluctuation of wireless channels can be exploited using power control in order to meet QoS requirements, i.e., a node can increase its transmit power whenever the interference at its receiver is low and decrease it when the interference is high. In that way, more information is transmitted when the channel conditions are favourable by adjusting the transmission rate accordingly. This approach enables the improvement of the system convergence and the satisfaction of heterogeneous service requirements. Xiao et al. and Abbas-Turki et al. [22, 49] designed some adaptive power control algorithms that change their QoS requirements depending on the channel conditions. However, they only consider a single channel and studied the effects on a single communication pair, ignoring its impact on the stability and convergence of the whole network as a system. In [50] an opportunistic power control algorithm is proposed that provides tunable parameters to have trade-off between throughput and power consumption. The algorithm is proven to converge to a unique fixed point. However, it does not consider a QoS requirement, but rather tries to maximise the throughput for individual users. Consequently, none of the approaches associate the QoS requirement with a cost or utility function that leads to a solvable power control problem. As a result there is no way to guarantee that on average their QoS targets are fulfilled.

Appendix: Mathematical Preliminaries

Some notions that are used in this chapter are more thoroughly described in the appendix, just for completeness.

Lyapunov Stability

Consider a differential equation

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)), \;\; \mathbf{x}(0) = \mathbf{x}_0 \tag{23}$$

that has a unique solution $\mathbf{x}_e$, i.e. $f(\mathbf{x}_e) = 0$ has a unique solution given by $\mathbf{x} = \mathbf{x}_e$. Let $\mathbf{z}(t) = \mathbf{x}(t) - \mathbf{x}_e$. Then, the differential equation (23) becomes

$$\dot{\mathbf{z}}(t) = g(\mathbf{z}(t)), \;\; \mathbf{z}(0) = \mathbf{z}_0 = \mathbf{x}_0 - \mathbf{x}_e \tag{24}$$

that has a unique solution $\mathbf{z}_e = \mathbf{0}$, i.e. $g(\mathbf{0}) = 0$ has a unique solution given by $\mathbf{z} = \mathbf{0}$.

Theorem 6 ([51]). *Consider a continuously differentiable function $V(\mathbf{z})$ such that*

$$V(\mathbf{z}) > 0, \;\; \forall \mathbf{z} \neq 0 \tag{25}$$

and $V(\mathbf{0}) = 0$. If $\dot{V}(\mathbf{z}) \leq 0 \; \forall \mathbf{z}$, then the equilibrium point is stable. If in addition, $\dot{V}(\mathbf{z}) < 0 \; \forall \mathbf{z} \neq \mathbf{0}$, then the equilibrium point is asymptotically stable. If in addition to these, V is radially unbounded, i.e., $V(\mathbf{z}) \to \infty$ when $\mathbf{z} \to \infty$, then the equilibrium point is globally asymptotically stable.

D-Stability

The notion of D-stability was initially introduced in the field of mathematical economics, but its properties are very useful for the study of dynamic equilibria and many important classes of matrices are linked with D-stability. The following summary is adopted from [52].

Definition 4 ([52]). Matrix $A \in C^{n \times n}$ is D-stable if DA is stable for all diagonal matrix D with positive entries.

Remark 5. If $A \in C^{n \times n}$ is D-stable, then:

1. AD is similar to DA ($DA = D(AD)D^{-1}$), so it is irrelevant in defining D-stability whether D is multiplied by the left or the right side of A.
2. A is nonsingular.
3. A^{-1} and A^* are D-stable.
4. DAE is D-stable, D, E positive diagonal matrices.
5. $P^T AP$ is D-stable, where P is any permutation matrix.

The following are some of the sufficient conditions for D-stability:

1. There exists a diagonal matrix D with positive entries, such that $DA + A^*D$ is positive definite.
2. $A \in \mathbb{R}^{n\times n}$ is an M-matrix.
3. There exists a positive diagonal matrix D, such that $DA = B = \{b_{ij}\}$ satisfies

$$\Re(b_{ii}) > \sum_{j=1,j\neq}^{n} |b_{ij}|,\ \forall i = 1,2,\ldots,n.$$

4. $A = \{a_{ij}\}$ is triangular and $\Re(a_{ii}) > 0$, $i = 1,2,\ldots,n$.
5. For each $0 \neq x \in C^{n\times n}$, there is a diagonal matrix D with positive entries such that $\Re(x^*DAx) > 0$.
6. $A \in \mathbb{R}^{n\times n}$ is oscillatory.

Fixed-Points, Contractions and Para-Contractions

We consider iterative algorithms on the form

$$\mathbf{x}(n+1) = T\big(\mathbf{x}(n)\big),\quad n = 0,1,2,\ldots, \tag{26}$$

where T is a mapping from a subset X of $\mathbb{R}^K$ into itself. A vector $\mathbf{x}^\star$ is called a fixed point of T if $T(\mathbf{x}^\star) = \mathbf{x}^\star$. If T is continuous at $\mathbf{x}^\star$ and the sequence $\{\mathbf{x}(n)\}$ converges to $\mathbf{x}^\star$, then $\mathbf{x}^\star$ is a fixed point of T [53, Chapter 3]. Therefore, the iteration (26) can be viewed as an algorithm for finding such a fixed point. T is called a *contraction mapping*, if it has the following property

$$\|T(\mathbf{x}) - T(\mathbf{y})\| \leq c\,\|\mathbf{x} - \mathbf{y}\|\,, \forall x, y \in X,$$

where $\|\cdot\|$ is some norm on X, and $c \in [0,1)$. The following proposition shows that contraction mappings have unique fixed points and linear convergence rates.

Proposition 2 (Convergence of Contracting Iterations [53, Chapter 3]). *If $T : X \to X$ is a contraction mapping and that X is a closed subset of $\mathbb{R}^K$, then:*

- *(Existence and uniqueness of fixed points) The mapping T has a unique fixed point $\mathbf{x}^\star \in X$.*
- *(Linear convergence) For every initial vector $\mathbf{x}(0) \in X$, the sequence $\{\mathbf{x}(n)\}$ generated by $\mathbf{x}(n+1) = T\big(\mathbf{x}(n)\big)$ converges to $\mathbf{x}^\star$ linearly. In particular,*

$$\|\mathbf{x}(n) - \mathbf{x}^\star\| \leq c^n \|\mathbf{x}(0) - \mathbf{x}^\star\|\,.$$

An operator T on X is called *para-contraction* if

$$\|T(\mathbf{x}) - T(\mathbf{y})\| < \|\mathbf{x} - \mathbf{y}\|\,,\ \text{for all}\ \mathbf{x} \neq \mathbf{y}\,.$$

Para-contractions have at most one fixed point and, in contrast to contractions, may not have a fixed point. As an example, consider the para-contracting function $T(x) = x + e^{-x}$ in $[0, \infty)$. It is easily seen that T has no fixed point. The following theorem summarises properties of para-contractions.

Proposition 3 ([54]). *If $T : X \to X$ is a para-contraction, then:*

- *If T has a fixed point $\mathbf{x}^\star$, then that fixed point is unique; moreover*
- *If X is a finite-dimensional space, for every initial vector $\mathbf{x}(0) \in X$, the sequence $\{\mathbf{x}(n)\}$ generated by $\mathbf{x}(n+1) = T\big(\mathbf{x}(n)\big)$ converges to $\mathbf{x}^\star$.*

As can be seen from Proposition 3, para-contractivity does not yield any estimate of the rate of convergence to the fixed point.

References

1. R. Sanchez, J. Evans, and G. Minden. Networking on the battlefield: challenges in highly dynamic multi-hop wireless networks. In *Military Communications Conference Proceedings, 1999. MILCOM 1999. IEEE*, volume 2, pages 751–755, 1999.
2. Gil Zussman and Adrian Segall. Energy efficient routing in ad hoc disaster recovery networks. *Ad Hoc Networks*, 1(4):405 – 421, 2003.
3. Chee-Yee Chong and S.P. Kumar. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, August 2003.
4. Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97, New York, NY, USA, 2002. ACM.
5. V. Kawadia and P.R. Kumar. Principles and Protocols for Power Control in Wireless Ad Hoc Networks. *Journal on Selected Areas in Communications (JSAC)*, 23(1): 76–88, January 2005.
6. S. Shakkottai, T. S. Rappaport, and P. C. Karlsson. Cross-Layer Design for Wireless Networks. *IEEE Communications Magazine*, 41(10):74–80, October 2003.
7. V. Srivastava and M. Motani. Cross-layer design: a survey and the road ahead. *Communications Magazine, IEEE*, 43(12):112–119, Dec. 2005.
8. G. Foschini and Z. Miljanic. A Simple Distributed Autonomous Power Control Algorithm and its Convergence. *IEEE Transactions on Vehicular Technology*, 42(4):641–646, November 1993.
9. Jens Zander. Distributed co-channel interference control in cellular radio systems. *IEEE Transaction on Vehicular Technology*, 41(3):305–311, August 1992.
10. S. Grandhi, J. Zander, and R. Yates. Constrained power control. *Wireless Personal Communications*, 2(3): 257–270, August 1995.
11. R. D. Yates. A framework for uplink power control in cellular radio systems. *IEEE Journal on Selected Areas in Communications*, 13:1341–1347, September 1995.
12. T. ElBatt and A. Ephremides. Joint Scheduling and Power Control for Wireless Ad-hoc Networks. In *Proceedings of IEEE INFOCOM*, 2002.
13. Zoran Gajic, Dobrila Skataric, and Sarah Koskie. Optimal SIR-based Power Updates in Wireless CDMA Communication Systems. In *IEEE Conference on Decision and Control*, volume 5, pages 5146–5151, December 2004.
14. Jens Zander. Performance of Optimum Transmitter Power Control in Cellular Radio Systems. *IEEE Transaction on Vehicular Technology*, 41(1):57–62, February 1992.

15. D. Mitra. An asynchronous distributed algorithm for power control in cellular radio systems. In *4th WINLAB Workshop*, Rutgers University, New Brunswick, NJ, 1993.
16. Nicholas Bambos, Shou C. Chen, and Gregory J. Pottie. Channel Access Algorithms with Active Link Protection for Wireless Communication Networks with Power Control. *IEEE/ ACM Transactions on Networking*, 8(5):583–597, 2000.
17. Fredrik Gunnarsson. Power control in cellular radio systems: Analysis, design and estimation, PhD Thesis, Linköping universitet 2000.
18. K.K. Leung, C.W. Sung, W.S. Wong, and T.M. Lok. Convergence theorem for a general class of power control algorithms. *IEEE Transactions of Communications*, 52(9):1566–1574, September 2004.
19. C.W. Sung and K.K. Leung. A generalized framework for distributed power control in wireless networks. *IEEE Transactions on Information Theory*, 51(7):2625–2635, 2005.
20. H. Boche and M. Schubert. A unifying approach to interference modeling for wireless networks. *IEEE Transactions on Signal Processing*, 58(6):3282–3297, June 2010.
21. H. R. Feyzmahdavian, M. Johansson, and Themistoklis Charalambous. Contractive Interference Functions and Rates of Convergence of Distributed Power Control Laws. In *International Conference on Communications (ICC)*, 2012.
22. M. Xiao, N.B.Shroff, and E.K.P. Chong. A utility-based power-control scheme in wireless cellular systems. *IEEE/ACM Transaction on Networking*, 11(2):210–221, April 2003.
23. Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge CB2 2RU, UK, 1985.
24. Roger A. Horn and Charles R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, Cambridge CB2 2RU, UK, 1994.
25. S.U. Pillai, T. Suel, and Seunghun Cha. The Perron-Frobenius theorem: some of its applications. *Signal Processing Magazine, IEEE*, 22:62–75, March 2005.
26. Fredrik Gunnarsson and Fredrik Gustafsson. Control theory aspects of power control in UMTS. *Control Engineering Practice*, 11(10):1113 – 1125, 2003.
27. D. ONeill, D. Julian, and S. Boyd. Seeking Foschini's genie: Optimal rates and powers in wireless networks. *IEEE Transaction on Vehicular Technology*, 2003 (accepted but not presented).
28. Xingwen Liu, Wensheng Yu, and Long Wang. Stability analysis of positive systems with bounded time-varying delays. *Trans. Cir. Sys.*, 56(7):600–604, 2009.
29. Lorenzo Farina and Sergio Rinaldi. *Positive Linear Systems: Theory and Applications*. New York: Wiley, July 2000.
30. M.A. Rami and F. Tadeo. Controller synthesis for positive linear systems with bounded controls. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 54(2):151–155, 2007.
31. O Mason and R Shorten. On linear copositive lyapunov functions and the stability of switched positive linear systems. *IEEE Transactions on Automatic Control*, 52(7), 2007.
32. Florian Knorn, Oliver Mason, and Robert Shorten. On linear co-positive lyapunov functions for sets of linear positive systems. *Automatica*, 45(8):1943 – 1947, 2009.
33. Themistoklis Charalambous, Ioannis Lestas, and Glenn Vinnicombe. On the stability of the foschini-miljanic algorithm with time-delays. In *CDC*, pages 2991–2996, 2008.
34. C. A. Desoer and Y. Yang. On the generalized Nyquist stability criterion. *IEEE Transaction on Automatic Control*, 25(1):187–196, 1980.
35. Xingzhe Fan, M. Arcak, and J.T. Wen. Robustness of CDMA power control against disturbances and time-delays. In *American Control Conference*, volume 3, pages 3622–3627, 2004.
36. Wassim M. Haddad and VijaySekhar Chellaboina. Stability theory for nonnegative and compartmental dynamical systems with time delay. *Systems and Control Letters*, 51(5):355 – 361, 2004.
37. Themistoklis Charalambous and Yassine Ariba. On the stability of a power control algorithm for wireless networks in the presence of time-varying delays. In *The 10th European Control Conference (ECC)*, August 2009.

38. S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM, Philadelphia, USA, 1994. in Studies in Applied Mathematics, vol.15.
39. Themistoklis Charalambous. A lyapunov krasovskii method for the stability of the foschini-miljanic algorithm under time-varying delays: An independent of delays approach. In *CUED/F-INFENG/TR.646*, January 2010.
40. Annalisa Zappavigna, Themistoklis Charalambous, and Florian Knorn. Unconditional stability of the Foschini-Miljanic algorithm. *Automatica*, 48(1):219 – 224, 2012.
41. F. Berggren. *Power control and adaptive resource allocation in DS-CDMA*. PhD thesis, KTH, Stockholm, Sweden, 2003.
42. Themistoklis Charalambous. *Power Control for Wireless Ad-Hoc Networks*. PhD thesis, University of Cambridge, July 2010.
43. Mohammed M. Olama, Seddik M. Djouadi, and Charalambos D. Charalambous. A general framework for continuous time power control in time varying long term fading wireless networks. In *Proceedings of the Ninth IASTED International Conference on Control and Applications*, CA '07, pages 69–74, Anaheim, CA, USA, 2007. ACTA Press.
44. G. C. Rota and G. Strang. A note on the joint spectral radius. *Proceedings of the Netherlands Academy*, 22:379–381, 1960.
45. T. Holliday, A. Goldsmith, N. Bambos, and P. Glynn. Distributed power and admission control for time-varying wireless networks. IEEE I*NTERNATIONAL* S*YMPOSIUM ON* I*NFORMATION* T*HEORY*, 2004.
46. Adam Czornik. On the generalized spectral subradius. *Linear Algebra and its Applications*, 407:242 – 248, 2005.
47. A. Moller and U.T. Jonsson. Stability of high order distributed power control. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 4963 –4970, December 2009.
48. A. Moller and U.T. Jonsson. Input Output Analysis of Power Control in Wireless Networks. In *Decision and Control, . Proceedings of the 49th IEEE Conference on*, pages 6451 – 6456, December 2010.
49. M. Abbas-Turki, F.de S. Chaves, H. Abou-Kandil, and J.M.T. Romano. Mixed H_2/H_∞ power control with adaptive qos for wireless communication networks. In *European Control Conference (ECC), Budapest, Hungary*, August 2009.
50. Chi Wan Sung and Kin Kwong Leung. Opportunistic power control for throughput maximization in mobile cellular systems. In *Communications, 2004 IEEE International Conference on*, volume 5, pages 2954–2958, June 2004.
51. H.Khalil. *Nonlinear* S*ystems*. Prentice Hall, Upper Saddle River, 2nd edition, 1996.
52. Charles R. Johnson. Sufficient conditions for d-stability. *Journal of Economic Theory*, 9(1):53 – 62, 1974.
53. D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation*. New Jersey 07458, USA, Prentice-Hall, 1989.
54. M. Edelstein. On fixed and periodic points under contractive mappings. *J. London Math. Soc.*, 37:74–79, 1962.

The Changing Role of Optimization in Urban Planning

James Keirstead and Nilay Shah

Abstract Most world cities are now planned in one way or another. Through the deliberate positioning of activity and transportation facilities, urban authorities hope to ensure the success of their cities in economic, social and environmental terms. Urban planning models are an important tool to help them in this task, and in this chapter, we examine the use of optimization techniques in urban planning modelling. Through a broad review of the field, we highlight the distinction between single-goal urban-environment models and multi-objective land use and transportation models. While it is shown that optimization no longer plays a stand-alone role in land use and transportation modelling, it does contribute to the overall modelling workflow. Furthermore, optimization forms the basis of two niche applications: excess commuting and sketch modelling. This last field holds the most promise for the future, enabling planners to establish minimum resource consumption benchmarks for their city as a means of comparison with other cities and to evaluate the ambition and feasibility of new plans.

Key words Cities • Urban planning • Mixed integer linear programming • Review

J. Keirstead
Department of Civil and Environmental Engineering, Imperial College London, SW7 2AZ, London, UK
e-mail: j.keirstead@imperial.ac.uk

N. Shah (✉)
Department of Chemical Engineering, Centre for Process Systems Engineering, Imperial College London, SW7 2AZ, London, UK
e-mail: n.shah@imperial.ac.uk

A. Chinchuluun et al. (eds.), *Optimization, Simulation, and Control*, Springer Optimization and Its Applications 76, DOI 10.1007/978-1-4614-5131-0_11,

1 Introduction

A city's character is greatly shaped by the organization of space and activities within its boundaries. This sense of place is partly an aesthetic attribute: by invoking the name of major world cities, we can quickly picture their structure and form such as the grid-iron streets and high-rises of New York or the hillside favelas of Rio de Janeiro. However, the city's social, economic and environmental performance is arguably the more important consequence of urban form.

For much of human history, cities evolved in an organic fashion "without preconceived planned intervention" [39, p. 10]. Natural determinants such as topography, climate and the availability of construction materials were major driving forces, shaping both architectural styles and activity location. The requirements of religion, politics, defence and logistics also played a role [39]. While the resultant forms may look random and uncoordinated, research has demonstrated a number of possible organic growth mechanisms including "preferential attachment" to existing settlements and transport networks [1, 14], economic processes (such as von Thünen's 1826 model of land rents) or analogies with physical processes such as diffusion-limited aggregation and dielectric breakdown [5].

The shift towards a more active form of urban planning often arose in response to the limitations imposed by haphazard urban growth. In Renaissance Rome for example, the tightly woven medieval structure of the city began to place significant constraints on the health and mobility of citizens and visitors. One of a number of planning popes, Sixtus V (1585–1590) located four major obelisks throughout the city to guide future planners in the construction of major thoroughfares that could connect prominent piazzas and churches. Similarly Haussmann's boulevards were an intentional effort to reshape Paris in response to changing defence requirements, inadequate sanitation and other factors [39]. While the specific constraints may vary over time and by location, the planning departments of modern cities essentially fulfil the same function: to create vibrant thriving urban areas subject to limitations of land, resources, finance and time. It is worth noting however that planned urban forms cannot be divorced from organic growth processes. Planned activity and transportation developments create opportunities for new patterns of urban living, which in turn need to be accommodated by new plans and construction. This cycle can be seen as the feedback loop which drives the growth of urban systems [52].

For the purpose of this chapter, we can broadly define urban planning as the policies that configure patterns of land use, associated activities and transportation. Urban planning is an interdisciplinary field, incorporating the expertise of architects, engineers, economists, sociologists and others. The planning process is necessarily a compromise between competing interests and multiple stakeholders, each of which may hold very different views about what constitutes a liveable neighbourhood or an effective strategic plan. In this context, it may seem that a technique as deterministic as optimization (or mathematical programming) has little to offer. However, this chapter will demonstrate that optimization techniques have been widely used in urban planning, although their precise contributions have shifted as other modelling techniques and the needs of analysts have changed.

This chapter is organized as follows. In Sect. 2, we provide an overview of the major uses of optimization modelling in urban planning over the past 50 years. The aim is not to provide a comprehensive review but to illustrate the range of applications, the specific techniques used, and to identify the reasons why mainstream urban planning analyses tend not to use these techniques now (or rather, do so indirectly). Section 3 then considers two current urban planning applications where optimization models are more commonly used, the fields of "excess commuting" and sketch planning. We review the structure of these models and offer comment on their formulation and applicability in various circumstances. Section 4 concludes by considering how optimization techniques for urban planning might evolve in future, concentrating on their use in the design of eco-cities.

2 Past Applications of Optimization in Urban Planning

The modern history of optimization might be said to begin with Dantzig's 1947 simplex algorithm for linear programming. Since then, advances in algorithms and computing technology have helped the field to expand, and mathematical programming models are now used in a variety of disciplines and formulations. This section provides an overview of the use of optimization in urban planning. First, we present a top-down review to identify major categories of practice and the types of optimization techniques employed. We then narrow the scope and work from key review articles to describe the major trends in the specific area of urban land use and transportation (LUT) planning.

2.1 Top-Down Review

We began our review by considering which general urban planning fields employ optimization techniques. To do this, we searched the ISI Web of Knowledge index[1] for the terms "`(optimization OR optimisation) AND (urban OR cities OR city) AND (planning)`" in both the topic and title fields. This led to 581 results, broken down into the subject areas shown in Table 1. While there is clearly a bias towards the more numerate subjects, the list of disciplines is very diverse. As noted in the introduction, the urban context attracts researchers from a wide variety of fields, and it is interesting to see that optimization offers at least some insight within all of these disciplines.

After inspecting the results, the query was further limited by adding "`AND land use`" to the search terms. This avoids a large number of papers that focus on topics not directly related to the question of urban land use. This includes work on the

[1] www.isiknowledge.com.

Table 1 Top ten subject areas for urban planning and optimization papers, as found by an ISI Web of Knowledge query

Subject area	Number of papers
Engineering	229
Environmental sciences and ecology	222
Computer science	123
Business and economics	110
Water resources	105
Mathematics	101
Public administration	69
Transportation	56
Public, environmental and occupational health	52
Agriculture	46
Total	581

See text for query phrasing

Table 2 Cross-tabulated categorization of selected urban planning and optimization papers; values indicate number of papers

	Optimization method					
Topic	LP	NLP	MOO	Other	Unknown	Total
Land use	4	2	13	4	6	29
Transport	2	0	2	1	2	7
Ecology	4	2	1	0	4	11
Water	0	2	3	2	1	8
Total	10	6	19	7	13	55

LP linear programming (including mixed-integer variants), *NLP* non-linear programming (including mixed-integer variants), *MOO* multi-objective optimization, *Other* hybrid methods such as cellular automata or agent-based simulation with an optimization component, as well as random and grey optimization, *Unknown* studies which do not specify the technique used

planning of large infrastructure systems without an explicit land use component (e.g. ant colony optimization applied to electricity networks, [16] or long-term water supply portfolio planning, [26]) and detailed operational optimization and control problems (e.g. for traffic light timing, [44]). The narrower search terms resulted in a more manageable 61 unique records. These papers were then categorized according to the optimization techniques used and the field of application. The cross-tabulated results shown in Table 2 therefore provide an indication of practice in this area. Six studies from the original sample were removed as they did not actually apply mathematical programming techniques, but rather referred to "optimization" in a non-technical manner (often as a synonym for improvement). From a methodological perspective, the table shows a mix of linear and non-linear, single and multi-objective formulations. The formulations are often, but not always, of mixed-integer form where integer variables are typically used to represent the classification of a discrete land use parcel [e.g. 10, 15]. Multi-objective approaches are commonly used as well and are typically solved by genetic algorithm [e.g. 3, 43], simulated annealing [e.g. 11] and to a lesser extent single aggregate objectives [e.g. 29].

Although the data set is diverse, a rough split in problem type can be seen. First, there are those studies which apply optimization techniques to examine the ecological impacts of urban development (the "ecology" and "water" categories above). In these cases, the natural environment imposes constraints on a city's growth and so might be called *urban-environment* planning models. For example, [10] describe how the growth of urban areas and pressures from agriculture can lead to fragmentation and degradation of nearby habitats, with negative impacts on biodiversity. They therefore use spatial optimization, based on an mixed-integer linear programming formulation, to allocate land function so that the minimum number of locations need to be actively restored after development. Constraints on the problem include overall areal targets for intact habitats and restrictions imposed by the local geography (e.g. soil types, vegetation). A similar approach is adopted by [38], who use non-linear programming to minimize impacts on water resources from urban growth policies.

The second set of problems focuses on urban planning in a more traditional economic or social context (the "land use" and "transportation" (LUT) categories); [2] is a typical example. Here a genetic algorithm is applied to determine future land-use and transportation provision for a growing city, subject to constraints on housing provision. The problem is one with multiple objectives such as minimizing cost, disruption, and traffic congestion. Multi-objective frameworks are significantly more common in this category, compared to urban-environment models that may have a narrower focus (42% of LUT models, 21% of urban-environment models, $\chi^2 = 5.55, p = 0.018$).

Two final points from this brief survey. First, researchers often combine optimization with other techniques, particularly for spatial analyses with cellular automata (e.g. for forestry planning, [37]) or agent-based models (e.g. for biodiversity planning involving multiple stakeholders, [27]). In these cases, the optimization routines can be embedded within heterogeneous agents to simulate the behaviour of individuals within a more complex interactive system. The second issue is that the selected papers span from 1979 to 2010 (a limitation of the ISI data set) and are therefore likely to be missing some of the early applications of these techniques to urban planning. The second part of this review will therefore provide more detailed perspective on the evolution of the field.

2.2 Optimization in LUT Models

For this second review, we limit our definition of urban planning to incorporate only the LUT sectors. Although environmental motivations are increasingly important, urban planning is still primarily concerned with creating economic and social opportunities [for an overview, see 41]. Quantitative work on the relationship between urban form and function typically falls under the general title of LUT modelling.

Review studies identify four major urban modelling approaches—regression, optimization, aggregate spatial models, and disaggregate individual models (including both random utility frameworks and activity-based models) [9, 20, 35]. Over time, the field has trended towards increasing behavioural realism and disaggregation. That is, whereas earlier studies could only simulate a few districts and had to aggregate all activity supply and demand within that zone, advances in computing power and model formulations mean that behaviours at the level of the individual or household can now be simulated (e.g. by modelling individual choices within an econometric random utility choice framework). The motivation for this shift can be explained by the operational use of these models, i.e. their deployment in real cities to answer policy questions such as how a city might expand over time [51], where people choose to live and work within cities [50], and which modes of transportation will be used to facilitate urban travel [20]. Greater behavioural fidelity allows these models to test sophisticated policy interventions and increases confidence that the salient processes have been effectively represented: "the value of more complex, behaviourally valid, microscopic models is not that one obtains microscopic forecasts, but that one obtains macroscopic forecasts based on microscopic principles" [48, p. 239].

However, this trend has meant that optimization has, over time, taken on a secondary role within urban LUT modelling. Several early studies used optimization as the primary technique to determine activity location and traffic flows within a city. For example, the earliest operational LUT optimization model appears to be TOPAZ from Australia [12, 46]. This model sought to allocate activities to discrete zones minimizing total cost from construction and travel. However, even in its early stages, limitations on data inputs and computational ability made these models impractical for everyday use [18, 32, 35]. Yet while optimization models have "all but disappeared" as stand-alone tools [52, p. 7], the techniques are still used in conjunction with more mainstream LUT modelling approaches, as so-called "combined" models [e.g. 9, 33, 42]. These tools use optimization to determine transportation costs endogenously, capturing spatial interactions and user behaviour in a more realistic fashion. More generally, optimization remains a useful technique within LUT models, for example to perform mean square error fitting of econometric models as a preliminary step in random utility choice modelling or to calculate market clearing equilibrium prices for land and transportation [52].

3 Present Applications: Excess Commuting and Sketch Planning

There are urban modelling niches within which optimization remains a valuable primary modelling methodology. It has been acknowledged that optimization techniques have the potential to find "extreme solutions" and to meet a specific objective during the "preparation of plans" (i.e. to be used in a normative fashion)

rather than forecasting detailed descriptive behavioural patterns over time [35, p. 330-331]. The use of optimization models for designing hypothetical optimal configurations, but not necessarily in assessments of existing cities, is also supported by [20]. Two such applications which will be discussed here are excess commuting and sketch layout modelling.

3.1 Excess Commuting

Excess commuting can be defined as commuting longer or further than suggested by the actual spatial layout of a city and assumptions about rational commuter behaviour [19]. Metrics based on this concept help analysts to compare the efficiency of urban layouts and suggest strategies for re-development and rationalization of spatial activity patterns. To estimate the level of excess commuting, original work in this field assumed a stylized monocentric city and found that, for a range of US and Japanese cities, average commuting journey distances were approximately 8 times greater than might be expected if commuters tried to minimize their average commute [17]. However, [53] reinterpreted this issue as a linear programming problem showing the level of excess commuting to be on the order of 11%. Research since 2000 has resulted in average estimates of excess commuting in the range of 50–70%, and the optimization framework is now the most common method of calculating benchmark minimum and maximum commutes for a given spatial layout [34].

The basic formulation of the problem, based on White's paper [53], is as follows:

$$\begin{aligned} \text{Minimize} \quad & \tfrac{1}{N}\sum_i\sum_j c_{i,j}n^*_{i,j} \\ \text{subject to} \quad & \sum_i n_{i,j} = \sum_i n^*_{i,j} = D_j \\ & \sum_j n_{i,j} = \sum_j n^*_{i,j} = O_i, \end{aligned}$$

where $c_{i,j}$ is the cost of commuting from zone i to j (e.g. distance or time), $n_{i,j}$ is the existing (exogenous) number of commuters travelling between zones i and j, $n^*_{i,j}$ is the optimized (endogenous) number of commuters, N is the total number of commuters, O_i is the number of workers living at zone i, and D_j is the number of workers employed in zone j. Readers familiar with operations research will recognize this as an example of the *transportation problem*, wherein the goal is to connect suppliers and customers with a network at minimum cost, subject to constraints on customer demand and supplier capacity [see 55]. It assumed that network capacities and transport requirements are specified at the outset. In the

urban context, this is analogous to knowing where people live and work and trying to determine the lowest cost routing for customers, i.e. the minimum flow of visits from home to each activity over the course of a year. This therefore results in an idealized measure of commuting flows that can be compared with observed flows and normalized to calculate the excess commuting statistic.

Optimization-based excess commuting has been applied in a variety of locations, but [34] highlight some common themes and outstanding questions. First there are methodological questions regarding biases in the calculation of the excess commute statistic. For example, measured data may need to be aggregated at different scales depending on availability and tractability. However the value of excess commute statistic depends on the chosen scale, with more aggregated spatial data resulting in a smaller estimate of excess commuting. This suggests that for accurate estimates, highly disaggregated data is preferred; however, this may create computational difficulties. A second issue is contextual, i.e. is the estimate a meaningful indicator of possible commuter flows? The technique described here relies upon an aggregate approach, i.e. considering travel to and from given urban areas for a particular activity class as a single homogeneous flow. However, more recent studies have looked to disaggregate individuals and households within these calculations, for example, to account for households with two working persons who need to live together but work in separate locations [7, 40]. Finally there are questions about the suitability of the measure for policy decisions, for example, on zoning regulations, housing policy, and road pricing.

The use of optimization modelling for the calculation of excess commuting is therefore an important application of the technique, even if theoretical and practical questions remain about its validity. However, a key point is that excess commuting measures are focused primarily on explaining observed behaviour in *existing* cities, where the locations of work, residence and other activities are exogenous inputs to the benchmark calculation. When considering new cities, these activity locations can be endogenous to the model so that transportation requirements are jointly determined with the location of individuals and activities. For this problem, we turn to a second current application of optimization: sketch models.

3.2 Sketch Models

Sketch models assess both activity location and transportation flows within an optimization framework; Fig. 1 provides a schematic overview of the technique. Clearly when compared with most operational urban planning models, sketch models provide an incomplete representation of LUT dynamics. When used in the context of master planning and benchmarking processes however, their reduced data requirements enables models to be tested quickly against multiple scenarios. The results can be then tested within a more rigorous LUT modelling framework as required.

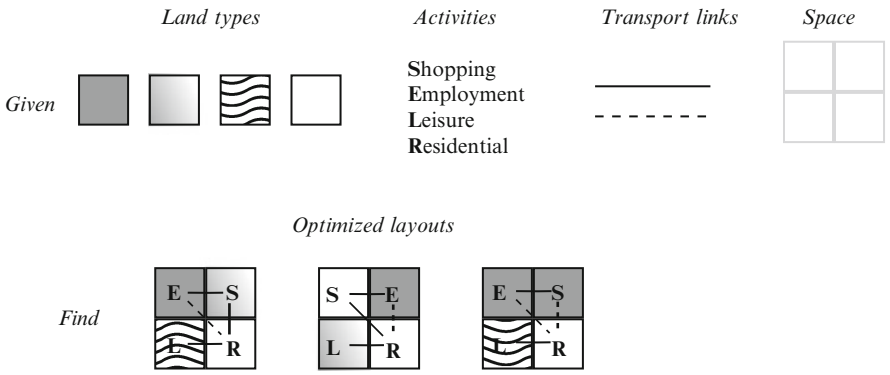


Fig. 1 A schematic description of a sketch model, based on [30]. The various combinations of land uses and transportation flows are assessed within an optimization framework according to objective functions which vary with the chosen application

Although [4] was not the first to use these techniques, his paper provides an excellent overview of the approach and illustrates a typical application wherein three goals are pursued in a multi-objective framework: minimization of land development costs, maximization of residential accessibility, and minimization of transportation energy costs. This work has been extended and revised by others, most notably in the sketch layout model [13, 30]. In its various incarnations, this model seeks to generate alternative master plan sketches as an input to participatory planning processes. Again a multi-objective approach is adopted and the model considers "harmony" (the similarity of adjacent land uses), "relevance" (the compatibility of adjacent land uses), and "traffic accessibility" (the shortest path between two cells). Related work in this area has examined the positioning of individual facilities within a city [e.g. shopping malls, 56], layout of space within buildings [47], the planning of development densities around transit stations [31], and the use of multi-objective optimization to generate a range of Pareto optimal layouts for discussion with planners [22].

The general formulation for sketch models is essentially a hybrid of two canonical operations research problems [see 55]. The first is the *transportation problem* discussed above. The second problem is the *assignment problem*, i.e. finding an optimal combination of tasks and agents where each pairing incurs a given cost. This problem can be formulated as follows:

$$\begin{aligned} \text{Minimize} \quad & \sum_{i,j} t_{i,j} x_{i,j} \\ \text{subject to} \quad & \sum_{i} x_{i,j} = 1 \quad \forall j \\ & \sum_{j} x_{i,j} = 1 \quad \forall i, \end{aligned}$$

where $x_{i,j}$ equals 1 if person i is assigned to job j, else 0, and $t_{i,j}$ is the cost of the assignment. Although it appears to be a mixed-integer problem, it can in fact be solved as an LP owing to the "integer-in-integer-out" properties of a network model [55]. In an urban context, the assignment problem can be seen as the task of allocating activity provision to different land areas, where "activity" might be interpreted variously as work, schools, residential housing and so on. The model can also be split so that building and activity types are assigned separately. In the case below for example, which focuses on minimizing urban energy consumption, there might be two building types that can support a single activity category.

The hybrid formulation is known as the *facility layout* problem, and while there are multiple forms of the problem's definition, the general aim is to determine the position of processes within a factory so that the combined costs of performing a task at a given work station and moving materials between each work station are minimized. Recent examples of this literature include [8, 28, 49]. As noted above, when planning a new city, there is no a priori reason for taking the location of housing and activities as fixed. Therefore a joint problem can be constructed which seeks to minimize the cost of assignment both function to land plots, and the travel required to move between those two locations.

The general formulation can be adapted with many application specific constraints. For example, in a factory layout problem, each piece of equipment might have a certain footprint and therefore require a certain amount of space. An analogous situation exists in the sketch planning case, where certain activities might require a minimum site area (e.g. for a school with a playground) or minimum total area (e.g. sufficient green space is provided for the whole city). An optimization-based sketch model can also handle constraints such as the capacity of transportation network links, housing requirements for the population, or prohibiting certain kinds of development on particular land plots.

3.3 An Example: Calculating a Minimum Energy Urban Layout

In our own work we have applied sketch models to the question of eco-cities, focusing particularly on urban energy consumption. Cities are major energy consumers, accounting for an estimated 67 % of global primary energy demand and 71 % of energy-related greenhouse gas emissions [21]. While there is some dispute about the precise allocation [45], it is clear that urban energy efficiency must be improved if economic and cultural opportunities are to be maintained while avoiding the worst environmental effects.

Urban energy consumption is the consequence of decisions taken at a variety of spatial and temporal scales. Using domestic energy consumption as an example, the temporal scale spans from short-term decisions such as when to use appliances (seconds to days), medium-term choices about which appliances to purchase (months

to years), and long-term decisions about the built fabric of the home (decades to centuries). For example, [6] note that, at current rates, the UK's housing stock could take approximately 1,300 years to be replaced completely. Variations in spatial scale also have a strong influence on energy consumption. End-use energy conversion technologies such as household gas boilers may be relatively easy to reposition or replace for improved efficiency but large infrastructure systems, such as resource distribution and transportation networks or the location of buildings and activities, are more persistent [e.g. 39]. Consequently the layout of urban environments is perhaps the most difficult aspect of improving urban energy efficiency.

For existing cities, urban expansion and retrofit projects can lead to improvements in energy consumption as seen in London's Canary Wharf and La Défense in Paris [54]. However, working within the constraints of existing infrastructures is expensive and difficult and so new construction arguably offers the greatest opportunities to create energy-efficient cities. In recent years, the eco-cities movement, both in the UK and abroad, has created visions of new sustainable urban areas [23], such as Masdar, the world's first "carbon-neutral zero-waste city" near Abu Dhabi in the United Arab Emirates [36]. However, these ambitions raise serious questions about the limits of low-energy urban forms, boundaries which must be identified if new developments are to set realistic goals and existing cities are to understand their improvement potential.

Not all of the energy issues listed above can be dealt with in an optimization model, or if they can be addressed, they may impose significant computational costs. However, a sketch model can be used to examine some key trade-offs such as the balance between transport capacity (i.e. the maximum number of trips that can occur on a route) and the maximum capacity of an activity site (i.e. the number of visitors that can be satisfied at a given location).

Our sketch model is described in full in [24] and contains both the core elements of an optimization-based sketch model, as described above, as well as some energy-specific features. As input, users provide information about:

- The population of the city
- The generic housing and transportation types that are available
- The spatial layout of the city (i.e. the location and size of the empty zones to be populated)
- The activities to be performed by the population

With the objective of minimizing "cost" (in this case, annual energy consumption), the model will then determine:

- The location of buildings and activities
- The location of network connections
- The number of daily trips from zone z to z' by mode m
- Other summary information (e.g. passenger km by mode)

Constraints on the problem generally fall into two categories: feasibility constraints (i.e. those that are necessary to obtain a valid solution, for example, that all

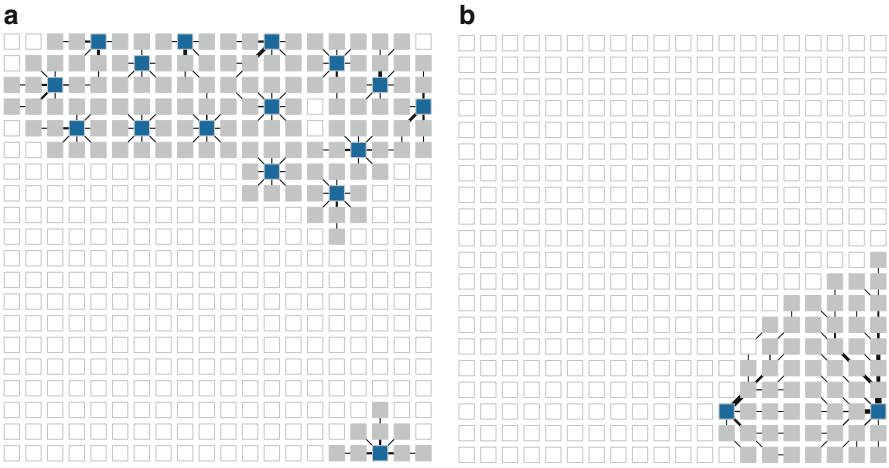


Fig. 2 Minimum energy layouts for the provision of work and housing under different assumptions. *Light grey cells* represent domestic housing, blue work locations. Transport links are shown in *black arrows*, with width proportional to the flows. Population of 100,000 in all cases. (**a**) Residential housing 60 dw/ha, each work site offers 3,200 jobs (**b**) Residential housing 130 dw/ha, each work site offers 48,000 jobs

citizens must be housed) and context constraints (i.e. additional restrictions to reflect planning laws or other user-specified concerns).

In this particular problem, the goal is to house 100,000 citizens and provide them with sufficient work. Each cell is 16 hectares and from UK statistics, it is assumed that 48 % of the population works. Figure 2a shows the results when we assume that the housing density is low (approximately 60 dwellings per hectare) and each work site can provide only 3,200 jobs (equivalent to a small office). However, in Figure 2b, higher-density housing is used (130 dwellings per hectare), and a single work site can employ 48,000 people (roughly equivalent to a dense central business district). In both cases, we have not added any binding constraints on transportation flows. The high-density case delivers an energy saving of approximately 15 %, accounting for both building and transport demands.

A closer inspection of these figures reveals the stylized nature of the analysis. In the low-density case, each work cluster is completely isolated; there are no traffic flows between centres as the model is able to satisfy work demand through these local "village" offices. Even in the high-density case, a clear divide between the two halves of the city can be seen. However, if one adds further activities besides work, the structure begins to change. In Fig. 3, a shopping activity is added and, when a single site is large enough to satisfy the city's resultant demand (e.g. a shopping mall), it is centrally located by the model so that all citizens can access it with minimal travel requirements.

The distributed structure of these results matches well with the predictions of Christaller's 1933 central place theory, which suggests that, for an unobstructed

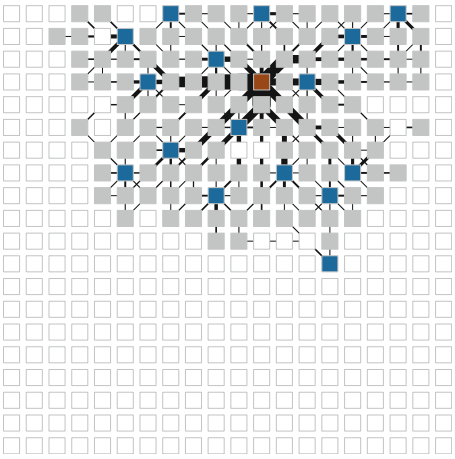


Fig. 3 A low-density layout for a hypothetical city with a central shop

landscape with an evenly distributed population, activity clusters should emerge at the centre of population areas subject to constraints on minimum market (threshold) and maximum distance (range) [41]. However, central place theory also predicts that individual local activity clusters feed into the demands for similar clusters at larger scales, in a hierarchical fashion. For example, a city may have distinct entertainment and business districts, but within each of these larger clusters, there will also be smaller local provision of these activities. Such a comparison therefore highlights the lack of multiple spatial scales within a simplified sketch model. To date, sketch models have tended to focus on a single spatial scale as the initial problem specification is built around discrete plots with an intended homogeneous purpose. This limitation is one reason why the results of such models are best used to inform planning discussion of smaller developments, where each zone contains a discrete activity, rather than to provide definitive plans or for the analysis of larger systems. Finally, a note about the optimization process itself. As a combinatorial optimization, the tractability of a sketch layout model is limited by the number of cells and activities to be positioned. Table 3 compares the capabilities of historic sketch models but note that these figures do not represent the limits of performance but the application size used for each study. While the table does show improvements over time, the problem remains fundamentally difficult. Looking at the solution in Fig. 2a for example, it can clearly be seen that a number of rotational and translational symmetries exist within the resulting structure. Therefore to improve performance and reduce degeneracy, it is useful for the user to provide some sensible constraints on the problem. This may include fixing an "anchor" activity at a given location within the model, in addition to using standard termination criteria for mixed-integer models such as a timeout or optimality gap.

Table 3 Comparison of previous urban layout optimization studies

Study	Case study	Computer set-up	Formulation	n_c	n_a
[4]	Germantown, WI, USA (7,000 people)	–	LP	9	11
[13]	Tanhai, Taiwan (300k people)	IBM/PC 486–33 with Turbo Pascal 6.0	MINLP	13	10
[30]	–	IBM Pentium II 300 MHz with Turbo Pascal 6.0	MINLP	13	7
[31]	Taipei central business district	–	LP	–	–
[56]	Dalian, China (2.23 million)	8 CPU cluster with C++	NLP	300	2
This paper	Generic UK	500 CPU cluster with GAMS/CPLEX 9.0 (though run on one core)	MILP	400	3

Other variables are used in these models, but only the core assignment problem variables, i.e. the number of cells (i.e. discrete zones within the model, n_c) and number of activities (i.e. land use categories, n_a) are shown here for an indicative comparison
Model formulations: *LP* linear programming, *NLP* non-linear programming, *MILP* mixed-integer linear programming, *MINLP* mixed-integer non-linear programming

4 Future Applications and Conclusions

This chapter has shown that optimization is a widely used technique in the urban modelling community. While in the past it was used on its own to determine LUT patterns, it has now fallen out of favour as a stand-alone technique with more behaviourally realistic models based on a disaggregated view of urban activities dominating current practice. As [9, p. 345] notes in his review, mathematical programming models have the advantage of "a simple mathematical form linked to system efficiency; however, the aggregate nature of the model means that there are inherent difficulties in representing the systematic properties of locations and the behavioural context of decision-makers." However, even in these problems, optimization is used to fit statistical models and in hybrid modelling applications, e.g. in conjunction with agent-based modelling.

Chang also observes that LUT modellers have focused "too much detail of the issues rather than the refinement of the foundational relationship." (p. 346). This suggests that a new look at the role of optimization modelling in urban planning might be in order. In Sect. 3, we showed that optimization remains a popular

technique in two niche applications: the determination of minimum commuting configurations and the rapid creation of sketch layouts early in the planning process. With the excess commuting literature pushing towards increased disaggregation and behavioural realism, sketch modelling seems like the most promising area for continued work on a purely optimization-based form of aggregate urban planning.

Early sketch modelling applications sought to contribute to the planning process by generating alternative plan ideas early in the planning process. This is still the general goal of such models, but the specific objective should be reworked slightly. Instead of focusing on realistic looking alternative plans that "rationally" balance multi-objectives, we would argue that there is significant scope for using sketch models to develop extreme scenarios with a specific goal in mind. In particular, the eco-cities movement has grand visions of low-impact urban settlements, driven by concerns over specific issues like carbon emissions. A resource-based sketch model could therefore be used to establish minimum benchmark values, i.e. patterns of development that meet basic goals of activity provision and housing with the lowest possible resource consumption. It is not envisioned that such plans would be built directly, but that by establishing a minimum benchmark, stakeholders could evaluate the ambition and difficulty of their actual designs in a more quantitative manner. In many ways, this application is similar to the urban-environment models highlighted in the review above. The emphasis is not on necessarily on multi-objective optimization, but on the pursuit of a single goal with the aim of identifying the limits of practice (although multi-objective optimization might still have a role as a goal like a "low-energy" city might have multiple energy-specific objectives such as carbon emissions and security of supply).

However, there appears to be at least three major obstacles or challenges in this field. First there is a question of scale, both spatial and temporal. Our analysis to date has focused on snapshot optimizations, as would be required to inform a single planning decision. However, resource infrastructure systems take decades to develop and must continually adapt to the needs of an evolving city. Multi-period optimization to look at minimum resource development pathways over time is therefore a promising area of research. The appropriate spatial scale, is also an issue. As identified above, most sketch models have tended to focus on a single spatial scale whereas the structure of cities consists of nested spatial scales. Hierarchical optimization methods might offer valuable insights here. This could offer performance improvements as well, for example, by solving a simplified relaxed version of the problem at a coarse spatial scale and then introducing integer variables to allocate homogeneous land functions at a local level.

The second major question is model fidelity. If sketch modelling is to be used to estimate the minimum resource consumption layouts for a city, what level of detail is needed by decision makers and can the models provide this? Taking the decision-makers perspective first of all, we can imagine a scenario where the goal is to establish a minimum energy baseline for a city. In such a scenario, building energy demands might be parameterized by means of normalized benchmarks (e.g. in kWh per square metre). However, the decision-maker may want to know if these demands can be reduced through demand side measures, such as flexible pricing,

or if higher per square demands are even relevant if the primary fuel source is low carbon. Clearly the more factors that become endogenous to the model, the more difficult it will be to construct and validate. The corollary to this problem is one of computational ability. Ideally, because sketch models represent a single optimized solution, they should be run multiple times to capture the uncertainty of input parameters and the range of possible outcomes; the goal should be to deliver a distribution of minimum resource benchmarks, not a single value [as in 29]. However, such a goal is in conflict with the use of a sketch model as a tool to quickly inform planning at an early stage. Multiple model runs will need to be solved and the mixed-integer formulation used here can result in slow solve times if not properly formulated. This can be partly resolved by parallel computing, but improved knowledge of the problem description and relevant heuristics will be valuable.

The third question is whether optimization is indeed the most effective technique for identifying the limits of feasible urban performance. Certainly its use in excess commuting and the simple examples shown here demonstrate that the basic idea is feasible, but it is uncertain what other techniques might be used to the same purpose. A specific question is whether or not the method works for existing cities. In new cities, a relatively unconstrained optimization makes sense; however in a city with substantial existing infrastructure, it is not clear whether the city has sufficient degrees of freedom to make a sketch model optimization meaningful. A more sensible approach in this case might be data envelopment analysis (which of course is also based on optimization) to identify the relative efficiency and performance of other cities.

Future research in this area should develop the concept of resource-based minimum benchmark urban plans and critically assess them in a variety of contexts (both locations and resource categories). One example is to combine these benchmark models with optimization-based models of resource supply systems. In [25], we examined the design of an eco-town by considering the layout and energy supply systems as separate optimization problems; however, the models could be combined to offer minimum energy benchmarks that consider both the supply and demand sides. While ultimately it may be found that a pure optimization-based approach is insufficient to capture the complexities of the urban environment, the history of the field suggests that optimization will continue to have an important role to play within the implementation of other techniques.

References

1. Albert R, Barabasi AL (2002) Statistical mechanics of complex networks. Reviews of Modern Physics 74(1):47–97
2. Balling R, Lowry M, Saito M (2003) Regional Land Use and Transportation Planning with a Genetic Algorithm. Journal of the Transportation Research Board 1831:210–218
3. Balling RJ, Taber JT, Brown MR, Day K (1999) Multiobjective Urban Planning Using Genetic Algorithm. Journal of Urban Planning and Development 125(2):86

4. Barber GM (1976) Land-use plan design via interactive multiple-objective programming. Environment and Planning A 8(6):625–636
5. Batty M, Longley P (1994) Fractal cities: a geometry of form and function. Academic Press, London
6. Boardman B, Darby S, Killip G, Hinnells M, Jardine C, Palmer J, Sinden G (2005) 40% House. Tech. rep., Environmental Change Institute, University of Oxford, Oxford
7. Buliung RN, Kanaroglou PS (2002) Commute minimization in the Greater Toronto Area: applying a modified excess commute. Journal of Transport Geography 10(3):177–186
8. Castillo I, Peters BA (2003) An extended distance-based facility layout problem. International Journal of Production Research 41(11):2451–2479
9. Chang J (2006) Models of the Relationship between Transport and Land-use: A Review. Transport Reviews 26(3):325–350
10. Crossman ND, Bryan BA, Ostendorf B, Collins S (2007) Systematic landscape restoration in the ruralurban fringe: meeting conservation planning and policy goals. Biodiversity and Conservation 16(13):3781–3802
11. Del Carmen Sabatini M, Verdiell A, Rodríguez Iglesias RM, Vidal M (2007) A quantitative method for zoning of protected areas and its spatial ecological implications. Journal of Environmental Management 83(2):198–206
12. Dickey J, Sharpe R (1974) Transportation and urban and regional development impacts. High Speed Ground Transportation Journal 8:71
13. Feng CM, Lin JJ (1999) Using a genetic algorithm to generate alternative sketch maps for urban planning. Computers, Environment and Urban Systems 23(2):91–108
14. Fisk D, Kerherve J (2006) Complexity as a cause of unsustainability. Ecological Complexity 3(4):336–343
15. Gabriel S, Faria J, Moglen G (2006) A multiobjective optimization approach to smart growth in land development. Socio-Economic Planning Sciences 40(3):212–248
16. Gomez J, Khodr H, De Oliveira P, Ocque L, Yusta J, Villasana R, Urdaneta A (2004) Ant colony system algorithm for the planning of primary distribution circuits. IEEE Transactions on Power Systems 19(2):996–1004
17. Hamilton B (1982) Wasteful commuting. Journal of Political Economy 90:1035–1053
18. Herbert J, Stevens B (1960) A model for the distribution of residential activity in urban areas. Journal of Regional Science 2:21
19. Horner MW (2002) Extensions to the concept of excess commuting. Environment and Planning A 34(3):543 – 566
20. Hunt JD, Kriger DS, Miller EJ (2005) Current operational urban land-use-transport modelling frameworks: A review. Transport Reviews 25(3):329–376
21. IEA (2008) World Energy Outlook. International Energy Agency, Paris
22. Jiang-Ping W, Qun T (2009) Urban planning decision using multi-objective optimization algorithm. ISECS International Colloquium on Computing, Communication, Control, and Management (pp. 392–394). IEEE. doi:10.1109/CCCM.2009.5267600
23. Joss S (2010) Eco-cities: a global survey 2009. Part A: eco-city profiles. URL http://www.westminster.ac.uk/schools/humanities/politics-and-international-relations/governance-and-sustainability/research/ecocities
24. Keirstead J, Shah N (2011) Calculating minimum energy urban layouts with mathematical programming and Monte Carlo analysis techniques. Computers, Environment and Urban Systems, 35(5), 368–377. doi:10.1016/j.compenvurbsys.2010.12.005
25. Keirstead J, Samsatli N, Shah N (2010) SynCity: an integrated tool kit for urban energy systems modelling. In: Bose R (ed) Energy Efficient Cities: Assessment Tools and Benchmarking Practices, World Bank, pp 21–42
26. Kirsch B, Characklis G, Dillard K, Kelley C (2009) More efficient optimization of long-term water supply portfolios. Water Resources Research 45(3):W03,414
27. Lagabrielle E, Botta A, Daré W, David D, Aubert S, Fabricius C (2010) Modelling with stakeholders to integrate biodiversity into land-use planning Lessons learned in Réunion Island (Western Indian Ocean). Environmental Modelling & Software 25(11):14

28. Lahmar M, Benjaafar S (2005) Design of distributed layouts. IIE Transactions 37(4):303–318
29. Ligmann-Zielinska A, Jankowski P (2010) Exploring normative scenarios of land use development decisions with an agent-based simulation laboratory. Computers, Environment and Urban Systems 34(5):409–423
30. Lin JJ, Feng CM (2003) A bi-level programming model for the land usenetwork design problem. The Annals of Regional Science 37(1):93–105
31. Lin JJ, Gau CC (2006) A TOD planning model to review the regulation of allowable development densities around subway stations. Land Use Policy 23(3):353–360
32. Los M (1978) Combined residential location and transportation models. Tech. rep., Centre for Transport Research, University of Montreal, Montreal
33. Lowry MB, Balling RJ (2009) An approach to land-use and transportation planning that facilitates city and region cooperation. Environment and Planning B: Planning and Design 36(3):487–504
34. Ma KR, Banister D (2006) Excess Commuting: A Critical Review. Transport Reviews 26(6):749–767
35. Mackett R (1985) Integrated land use-transport models. Transport Reviews 5(4):325–343
36. Masdar City (2010) Welcome to Masdar City. URL http://www.masdarcity.ae/en/index.aspx
37. Mathey AH, Krcmar E, Dragicevic S, Vertinsky I (2008) An object-oriented cellular automata model for forest planning problems. Ecological Modelling 212(3–4):359–371
38. Mejia AI, Moglen GE (2009) Spatial Patterns of Urban Development from Optimization of Flood Peaks and Imperviousness-Based Measures. Journal of Hydrologic Engineering 8(4):1
39. Morris A (1994) A History of Urban Form: Before the Industrial Revolutions. Longman, Harlow
40. O'Kelly M, Lee W (2005) Disaggregate journey-to-work data: implications for excess commuting and jobs- housing balance. Environment and Planning A 37(12): 2233–2252
41. Pacione M (2009) Urban geography: a global perspective, 3rd edn. Routledge, London
42. Pfaffenbichler P, Shepherd S (2002) A Dynamic Model to Appraise Strategic Land-Use and Transport Policies. European Journal of Transportation and Infrastructure Research 2(3/4):255–283
43. Reichold L, Zechman EM, Brill ED, Holmes H (2010) Simulation-Optimization Framework to Support Sustainable Watershed Development by Mimicking the Predevelopment Flow Regime. Journal of Water Resources Planning and Management 136(3):366–375
44. Sadoun B (2008) On the simulation of traffic signals operation. Simulation 84(6):285
45. Satterthwaite D (2008) Cities' contribution to global warming: notes on the allocation of greenhouse gas emissions. Environment and Urbanization 20(2):539–549
46. Sharpe E, Brotchie J, Ahern P (1975) Evaluation of alternative growth patterns for Melbourne. In: Karlqvist A, Lundqvist L, Snickars F (eds) Dynamic Allocation of Urban Space, Saxon House, p 259
47. Sharpe R, Marksjö B, Mitchell J, Crawford J (1985) An interactive model for the layout of buildings. Applied Mathematical Modelling 9(3):207–214
48. Timmermans H (2003) The saga of integrated land use-transport modeling: How many more dreams before we wake up. Keynote paper 10th International Conference on Travel Behaviour Research pp 219–248
49. Urban TL, Chiang WC, Russell RA (2000) The integrated machine allocation and layout problem. International Journal of Production Research 38(13):2911
50. Waddell P (2000) A behavioral simulation model for metropolitan policy analysis and planning: residential location and housing market components of UrbanSim. Environment and Planning B: Planning and Design 27(2):247–263
51. Wang C, Wu J (2010) Natural amenities, increasing returns and urban development. Journal of Economic Geography, lbq020. doi:10.1093/jeg/lbq020
52. Wegener M (2004) Overview of land use transport models. In D. A. Hensher & K. Button (Eds.), Handbook of transport geography and spatial systems. Kidlington: Pergamon, Elsevier Science Ltd pp. 127–146

53. White MJ (1988) Confirmations and Contradictions: Urban Commuting Journeys Are Not "Wasteful". The Journal of Political Economy 96(5):1097–1110
54. de Wilde S, van Den Dobbelsteen A (2004) Space use optimisation and sustainability - environmental comparison of international cases. Journal of Environmental Management 73(2):91–101
55. Williams HP (1999) Model Building in Mathematical Programming, 4th edn. Wiley, Chichester
56. Yu B, Yang Z, Cheng C (2007) Optimizing the distribution of shopping centers with parallel genetic algorithm. Engineering Applications of Artificial Intelligence 20(2):215–223

Parametric Optimization Approach to the Solow Growth Theory

Rentsen Enkhbat and Darkhijav Bayanjargal

Abstract We extend the classical growth theory model assuming that production function is an arbitrary continuously differentiable function on its domain and the saving rate and depreciation rate of capital depend on time. Then the per capita consumption maximization problem reduces to one dimensional parametric maximization problem. We propose a new finite method for solving the problem using Lipschitz condition. Some test problems have been solved numerically.

Key words Growth theory • Nonconvex • Parametric optimization problem

1 Introduction

The production function model was applied to the study of growth problems by Solow [6]. Solow developed a growth theory model within a neoclassical economic framework. The Solow growth model assumes the maximization of per capita consumption under economic equilibria or steady state [1,4]. This model from view point of optimization problem was considered in [3]. In paper [3], we proposed some global optimization methods and algorithms for solving the per capita consumption maximization problem using quasiconcave production functions [2,5].

This chapter is organized as follows. In Sect. 1 we introduced general Solow growth model and classical assumptions. In Sect. 2 we consider the Solow growth model for nonconvex production function and population growth function with variable rates. The per capita consumption maximization problem is formulated as a parametric optimization problem. Section 3 is devoted to numerical results.

R. Enkhbat (✉) • D. Bayanjargal
The School of Economics Studies, National University of Mongolia, Ulaanbaatar, Mongolia
e-mail: enkhbat@ses.edu.mn; bayanjargal@ses.edu.mn

A. Chinchuluun et al. (eds.), *Optimization, Simulation, and Control*, Springer Optimization and Its Applications 76, DOI 10.1007/978-1-4614-5131-0_12,

Now consider briefly Solow growth model. The production function relating output Y to capital K and labor L is

$$Y(t) = f(K(t), L(t)). \tag{1}$$

We assume a fraction s of income is saved and invested. Then the standard capital accumulation equation is

$$\begin{cases} K'(t) = sf(K(t), L(t)) - \mu K(t) \\ K(t_0) = K_0, \end{cases} \tag{2}$$

where s the savings (saving rate), $0 \leq s \leq 1$, and μ depreciation rate of capital and the consumption C is

$$C(t) = (1-s)f(K(t), L(t)). \tag{3}$$

Let $f(K,L)$ be a concave, differentiable homogeneous production function. Assume that the labor grows at exponential rate η which means that

$$L = L_0 e^{\eta t}. \tag{4}$$

Define k as per capita capital function:

$$k(t) = \frac{K(t)}{L(t)}.$$

Then

$$k'(t) = \left(\frac{K(t)}{L(t)}\right)' = \frac{K'L - L'K}{L^2} = \frac{1}{L}\left(K' - \frac{L'}{L}K\right) = \frac{1}{L}\left(K' - \eta K\right).$$

If we substitute $f(K,L)$ into this equation, we have

$$\begin{aligned} k' &= \frac{1}{L}\left(s(t)f(K,L) - \mu K - \eta K\right) \\ &= s(t)f\left(\frac{K}{L}, 1\right) - \mu\frac{K(t)}{L(t)} - \eta\frac{K(t)}{L(t)} \\ &= s(t)\varphi(k) - (\mu + \eta)k, \end{aligned}$$

where $\varphi(k)$ is the Solow per capita production function. Then per capita capital accumulation equation is

$$\begin{cases} k' = s(t)\varphi(k) - (\mu + \eta)k \\ k(0) = k_0. \end{cases} \tag{5}$$

Economic equilibria condition is:

$$k' = s(t)\varphi(k) - (\mu + \eta)k = 0. \tag{6}$$

Now we consider per capita consumption function

$$c(t) = \frac{C(t)}{L(t)} = \frac{(1-s)f(K,L)}{L} = (1-s)\varphi(k). \tag{7}$$

Assume that s is a constant function.

Let us consider the per capita consumption maximization problem subject to economic equilibria. That is

$$c = (1-s)\varphi(k) \to \max, \tag{8}$$

$$s\varphi(k) - (\mu + \eta)k = 0. \tag{9}$$

Then problems (8) and (9) are equivalent to the following one-dimensional problem:

$$\max_k \left[\varphi(k) - (\mu + \eta)k\right].$$

The solution k^* satisfies the equation

$$\varphi'(k^*) = (\mu + \eta). \tag{10}$$

So-called the golden rule of level of accumulation is determined from (9) as in [6]:

$$s^* = \frac{\varphi'(k^*)k^*}{\varphi(k^*)}. \tag{11}$$

2 Economic Growth with Nonconvex Production Functions

In general, we can consider a capital accumulation equation

$$\begin{cases} K'(t) = s(t)f(K(t),L(t),t) - \mu K(t) \\ K(t_A) = K_A, \end{cases} \tag{12}$$

when $f(K,L,t)$ and $L(t)$ are arbitrary continuously differentiable given functions on their domains and $s = s(t), \mu = \mu(t)$ are functions of t, $t \in [t_A, t_B]$.

Then a per capita consumption function is

$$c(t) = \frac{C(t)}{L(t)} = \frac{(1-s(t))f(K(t),L(t),t)}{L(t)}. \tag{13}$$

Also, a per capita capital function is

$$k(t) = \frac{K(t)}{L(t)}. \tag{14}$$

Economic equilibria condition is written as

$$k'(t) = \left(\frac{K(t)}{L(t)}\right)' = \frac{K'(t)L(t) - L'(t)K(t)}{L^2} = 0 \tag{15}$$

or

$$\frac{K'}{K} = \frac{L'}{L},$$

which means that capital and labor growth rate must be equal.

Taking into account (12), we can write (15) as follows:

$$K'(t)L(t) - L'(t)K(t) = \big(s(t)f(K,L,t) - \mu(t)K(t)\big)L(t) - L'(t)K(t) = 0.$$

From this equation we find $s(t)$:

$$s(t) = \frac{\mu KL + L'K)}{Lf(K,L,t)}. \tag{16}$$

Denote by K_B the maximum of function $K(t)$ defined by (12), i.e.,

$$K_B = \max_{t_A \le t \le t_B} K(t), \tag{17}$$

and introduce the function $\phi(K,L,t)$ as follows:

$$\phi(K,L,t) = f(K,L,t) - \frac{\mu KL + L'K}{L}. \tag{18}$$

Definition 1. If the following condition

$$\phi(K,L,t) \ge 0, \ \ \forall t \in [t_A, t_B],$$

holds, then the interval $[K_A, K_B]$ is called an economic efficient interval.

From Definition 1 and (16), we can easily notice that $0 \le s(t) \le 1$.

Now we consider per capita consumption maximization problem on a given interval $[t_A, t_B]$:

$$c(t) = \frac{(1 - s(t))f(K,L,t)}{L} \to \max, \ \ t \in [t_A, t_B],$$

subject to

$$s(t) = \frac{\mu KL + L'K}{Lf(K,L,t)}.$$

This problem is equivalent to the following one dimensional parametric maximization problem:

$$F(K,t) = \frac{f(K,L,t)}{L} - \frac{\mu KL + L'K}{L^2} \to \max_K, \quad t \in [t_A, t_B]. \tag{19}$$

We can write function $f(K,L,t)$ as

$$f(K,L,t) = \frac{\mu KL + L'K}{L} + \phi(K,L,t).$$

Further, we assume that $[K_A, K_B]$ is an efficient interval.

Then the problem (19) can be rewritten as

$$F(K,t) = \frac{f(K,L,t)}{L} - \frac{\mu KL + L'K}{L^2} = \frac{\phi(K,L,t)}{L} \to \max_K, \quad t \in [t_A, t_B]. \tag{20}$$

Problem (20) is a hard parametric optimization problem.

Lemma 1. *The function $F(K,t)$ satisfies the Lipschitz condition with respect to t with constant M for each $K \in [K_A, K_B]$, i.e.,*

$$\left|F(K,\hat{t}) - F(K,t)\right| \le M\left|\hat{t} - t\right|, \quad \forall t \in [t_A, t_B].$$

Proof. Since $f(K,L,t)$ is a continuously differentiable function with respect to t, using Taylor expansion formula, we can write down:

$$f(K,L,t+\Delta t) - f(K,L,t) = \frac{\partial f(K,L,t+\theta\Delta t)}{\partial t}\Delta t,$$

where $0 < \theta < 1, t + \theta\Delta t \in [t_A, t_B]$.

Now we have the following estimation:

$$\begin{aligned}\left|f(K,L,t+\Delta t) - f(K,L,t)\right| &= \left|\frac{\partial f(K,L,t+\theta\Delta t)}{\partial t}\right|\left|\Delta t\right| \\ &\le \max_{\substack{K_A \le K \le K_B \\ t_A \le t \le t_B}} \left|\frac{\partial f(K,L,t)}{\partial t}\right|\left|\Delta t\right|.\end{aligned}$$

By setting $M = \max_{\substack{K_A \le K \le K_B \\ t_A \le t \le t_B}} \left| \frac{\partial f(K,L,t)}{\partial t} \right|$, we obtain

$$\left|F(K,\hat{t}) - F(K,t)\right| \le M\left|\hat{t} - t\right|$$

which proves the lemma. □

Lemma 2. *Assume that the production function $f(K,L,t)$ is a continuously differentiable function with respect to t. Then for a given $\epsilon > 0$, there exists a discretization*

$$t_A = t_0 < t_1 < \cdots < t_i < t_{i+1} < \cdots < t_N = t_B$$

such that

$$|F(K^*(t),t) - F(K^*(t_i),t_i)| < \epsilon \text{ for all } t \in [t_A,t_B] \text{ and certain } t_i,$$

where

$$F(K^*(t),t) = \max_{K \in [K_A,K_B]} F(K,t), \ \ t \in [t_A,t_B].$$

Proof. We discretize $[t_A,t_B]$ in the following way:

$$t_A = t_0, t_i = t_0 + i\frac{t_B - t_A}{N}, i = 1,2,\ldots,N.$$

Clearly, for any $t \in [t_A,t_B]$, there exists $j \in \{1,2,\ldots,N\}$ such that $t \in [t_j,t_{j+1}]$. Consequently,

$$|t - t_j| < \frac{t_B - t_A}{N}. \tag{21}$$

Due to Lemma 1, there exists $M > 0$ such that

$$|f(K(\hat{t}),L(\hat{t}),\hat{t}) - f(K(t),L(t),t)| < M|\hat{t} - t|, \ \forall t,\hat{t} \in [t_A,t_B].$$

Define $\epsilon > 0$ as follows:

$$\epsilon = M\frac{t_B - t_A}{N}. \tag{22}$$

Now take any $t \in [t_A,t_B]$ and compute

$$|F(K^*(t),t) - F(K^*(t_j),t_j)| \le M|t - t_j| \le M\frac{t_B - t_A}{N} = \epsilon$$

which proves the lemma. □

The above lemma allows us to find $\epsilon-$ approximate solution of problem (20) by solving a finite number of nonlinear optimization problems.

3 Numerical Results

To reinforce the theoretical results, we solve the following problems numerically on $t \in [1,5]$ for the given parameters of $\eta = 0.012, L_0 = 2, N = 40, \epsilon = 0.001$.

Example 1.

$$F(K(t),L(t)) = \frac{\phi(K(t),L(t))}{L(t)} \to \max_K, \ \ t \in [1,5],$$

where

$$\phi(K,L) = -L^2K^4 + 8LK^3 - 9K^2 + 5, \ \ K \in [1;4],$$

and the labor grows at exponential rate η. The numerical results of the example are shown in Table 1.

Example 2.

$$F(K(t),L(t)) = \frac{\phi(K(t),L(t))}{L(t)} \to \max_K, \ \ t \in [1,5],$$

where

$$\begin{aligned}\phi(K,L) = {} & 0.000108K^5 - 0.00596LK^4 + 0.11365K^3 \\ & -0.889572K^2 + 2.986324K, \ \ K \in [0;6],\end{aligned}$$

and the labor grows at exponential rate η. The numerical results of example 2 are given in Table 2.

Table 1 Example 1

t	L	K^*	t	L	K^*	t	L	K^*
1.0	2.0404	2.5100	2.4	2.0983	2.4407	3.8	2.1579	2.3732
1.1	2.0445	2.5049	2.5	2.1025	2.4358	3.9	2.1622	2.3686
1.2	2.0486	2.4999	2.6	2.1068	2.4309	4.0	2.1666	2.3638
1.3	2.0527	2.4949	2.7	2.111	2.4260	4.1	2.1709	2.3591
1.4	2.0568	2.4899	2.8	2.1152	2.4242	4.2	2.1753	2.3543
1.5	2.0609	2.4850	2.9	2.1194	2.4164	4.3	2.1796	2.3497
1.6	2.065	2.4801	3.0	2.1237	2.4115	4.4	2.184	2.3449
1.7	2.0692	2.4750	3.1	2.1279	2.4067	4.5	2.1883	2.3403
1.8	2.0733	2.4701	3.2	2.1322	2.4019	4.6	2.1927	2.335
1.9	2.0775	2.4651	3.3	2.1365	2.3971	4.7	2.1971	2.3310
2.0	2.0816	2.4603	3.4	2.1407	2.3924	4.8	2.2015	2.3263
2.1	2.0858	2.4554	3.5	2.145	2.3876	4.9	2.2059	2.3217
2.2	2.09	2.4504	3.6	2.1493	2.3828	5.0	2.2103	2.3355
2.3	2.0941	2.4456	3.7	2.1536	2.3780			

Table 2 Example 2

t	L	K^*	t	L	K^*	t	L	K^*
1.0	2.0404	2.4270	2.4	2.0983	2.4068	3.8	2.1579	2.3870
1.1	2.0445	2.4256	2.5	2.1025	2.4054	3.9	2.1622	2.3857
1.2	2.0486	2.4241	2.6	2.1068	2.4039	4.0	2.1666	2.3842
1.3	2.0527	2.4227	2.7	2.111	2.4025	4.1	2.1709	2.3828
1.4	2.0568	2.4212	2.8	2.1152	2.4011	4.2	2.1753	2.3814
1.5	2.0609	2.4198	2.9	2.1194	2.3997	4.3	2.1796	2.3800
1.6	2.065	2.4183	3.0	2.1237	2.3982	4.4	2.184	2.3786
1.7	2.0692	2.4168	3.1	2.1279	2.3969	4.5	2.1883	2.3773
1.8	2.0733	2.4154	3.2	2.1322	2.3954	4.6	2.1927	2.3759
1.9	2.0775	2.4139	3.3	2.1365	2.3940	4.7	2.1971	2.3745
2.0	2.0816	2.4125	3.4	2.1407	2.3927	4.8	2.2015	2.3732
2.1	2.0858	2.4110	3.5	2.145	2.3912	4.9	2.2059	2.3718
2.2	2.09	2.4097	3.6	2.1493	2.3898	5.0	2.2103	2.3704
2.3	2.0941	2.4082	3.7	2.1536	2.3884			

Table 3 Example 3

t	L	K^*	t	L	K^*	t	L	K^*
1.0	2.0404	1.2131	2.4	2.0983	1.1874	3.8	2.1579	1.1619
1.1	2.0445	1.2112	2.5	2.1025	1.1856	3.9	2.1622	1.1602
1.2	2.0486	1.2097	2.6	2.1068	1.1838	4.0	2.1666	1.1583
1.3	2.0527	1.2075	2.7	2.111	1.1819	4.1	2.1709	1.1564
1.4	2.0568	1.2057	2.8	2.1152	1.1801	4.2	2.1753	1.1547
1.5	2.0609	1.2039	2.9	2.1194	1.1783	4.3	2.1796	1.1528
1.6	2.065	1.2021	3.0	2.1237	1.1765	4.4	2.184	1.1510
1.7	2.0692	1.2002	3.1	2.1279	1.1747	4.5	2.1883	1.1492
1.8	2.0733	1.1984	3.2	2.1322	1.1729	4.6	2.1927	1.1472
1.9	2.0775	1.1966	3.3	2.1365	1.1710	4.7	2.1971	1.1456
2.0	2.0816	1.1947	3.4	2.1407	1.1692	4.8	2.2015	1.1438
2.1	2.0858	1.1929	3.5	2.145	1.1674	4.9	2.2059	1.1420
2.2	2.09	1.1910	3.6	2.1493	1.1655	5.0	2.2103	1.1402
2.3	2.0941	1.1893	3.7	2.1536	1.1638			

Example 3.

$$F(K(t),L(t)) = \frac{\phi(K(t),L(t))}{L(t)} \to \max_K, \;\; t \in [1,4],$$

where

$$\phi(K,L) = \log(K^2 + LK)/K, \;\; K \in [1;3],$$

and the labor grows at exponential rate η.

The numerical results of example 3 are shown in Table 3.

References

1. David Romer, *Advanced Macroeconomics*, University of California, Berkeley, 1996.
2. R.Enkhbat, *Quasiconvex Programming*, Lambert Publisher, Germany, 2009.
3. R.Enkhbat, D.Bayanjargal and A.Griewank, *Global Optimization Approach to the Solow Growth Theory*, Advanced Modeling and Optimization, An Electronic International Journal, pp.133–140, Vol.12, Number 2, 2010.
4. H.Gregory Mankiw, *Macro Economics*, Harvard University, Worth Publisher, New York, 2010.
5. R.Horst, Panos M.Pardalos and N.Thoai, *Introduction to Global Optimization*, Dordrecht, The Netherlands, 1995.
6. Robert M.Solow, *A Contribution to the Theory of Economic Growth*, Quarterly Journal of Economics, (The MIT Press)70(1):65–94, 1956.

Cyclical Fluctuations in Continuous Time Dynamic Optimization Models: Survey of General Theory and an Application to Dynamic Limit Pricing

Toichiro Asada

Abstract In this chapter, we reconsider the analytical results on the existence of cyclical fluctuations in continuous time dynamic optimization models with two state variables and their applications to dynamic economic theory. In the first part, we survey the useful analytical results which were obtained by Dockner and Feichtinger (J Econom 53–1:31–50, 1991), Liu (J Math Anal Appl 182:250–256, 1994) and Asada and Yoshida (Chaos, Solitons and Fractals 18:525–536, 2003) on the general theory of cyclical fluctuations in continuous time dynamic optimizing and non-optimizing models. In the second part, we provide an application of these analytical results to a particular continuous time dynamic optimizing economic model, that is, a model of dynamic limit pricing with two state variables, which is an extension of Gaskins (J Econom Theor 3:306–322, 1971) prototype model.

Key words Cyclical fluctuations • Continuous time • Dynamic optimization models • Hopf Bifurcation • Dynamic limit pricing

1 Introduction

It is well known that the typical continuous time dynamic optimization model with only one state variable, which is very popular in economics, does not produce the cyclical fluctuations but it produces the monotonic convergence to the equilibrium point. On the other hand, some economic theorists provided various types of continuous time dynamic optimization models with two state variables which entail cyclical fluctuations. Some examples of such works are [3, 4, 7, 8].

T. Asada (✉)
Chuo University, 742-1 Higashinakano, Hachioji, Tokyo 102-0393, Japan
e-mail: asada@tamacc.chuo-u.ac.jp

A. Chinchuluun et al. (eds.), *Optimization, Simulation, and Control*, Springer Optimization and Its Applications 76, DOI 10.1007/978-1-4614-5131-0_13,

The above-mentioned works showed the existence of closed orbits as the optimal trajectories analytically as well as numerically by applying the Hopf Bifurcation theorem.[1]

All of the above-mentioned works are the studies of particular economic models rather than the systematic investigations of the general continuous time dynamic optimization models with two state variables. On the other hand, [11] provided an exhaustive classification of the nature of the solution of such a general model including the conditions for the occurrence of the Hopf Bifurcation. [13, 14] are examples of the applications of [11] theorem to the economic models. Asada and Yoshida [6] discussed on the analytical results of [11] from a particular point of view.

In this chapter, we reconsider the analytical results on the existence of cyclical fluctuations in continuous time dynamic optimization models with two state variables and their applications to dynamic economic theory. Our strategy is to take up a particular economic model from the viewpoint of an application of the general theory of dynamic optimization. In Sect. 2, we survey the useful analytical results which were obtained by [6, 11, 18] on the general theory of cyclical fluctuations in continuous time dynamic optimizing and non-optimizing models. In Sect. 3, we provide an application of these analytical results to a particular continuous time dynamic optimization model, that is, a model of dynamic limit pricing with two state variables, which is an extension of [16] prototype model. Section 4 is devoted to an interpretation of the analytical results obtained in Sect. 3.

2 Survey of General Theory

In this section, we survey some useful analytical results on the existence of cyclical fluctuations in continuous time dynamic optimization and non-optimization models. First, let us quote the following "Hopf Bifurcation theorem" that describes a set of sufficient conditions for the existence of the closed orbits in a general n-dimensional system of nonlinear differential equations (cf. [15] Chap. 24 and [2] Mathematical Appendix).

Theorem 1 (Hopf Bifurcation theorem). *Let $\dot{x} = f(x;\epsilon), x \in R^n$, $\epsilon \in R$ be an n-dimensional system of differential equations depending upon a parameter ϵ. Suppose that the following conditions (H1)–(H3) are satisfied:*

(H1) The system has a smooth curve of equilibria given by $f(x*(\epsilon);\epsilon) = 0$.

[1] This does not necessarily mean that every continuous time dynamic optimization model with two state variables produces cyclical fluctuations. For example, [5] proved analytically that [19] continuous time dynamic optimization model of endogenous growth with two state variables entails only the monotonic convergence to the equilibrium point.

(H2) *The characteristic equation* $|\lambda I - Df(x*(\epsilon_0);\epsilon_0)| = 0$ *has a pair of pure imaginary roots* $\lambda(\epsilon_0), \bar{\lambda}(\epsilon_0)$ *and no other roots with zero real parts, where* $Df(x*(\epsilon_0);\epsilon_0)$ *is the Jacobian matrix of the above system at* $(x*(\epsilon_0),\epsilon_0)$ *with the parameter value* ϵ_0.

(H3) $\left.\frac{d\{\mathrm{Re}\,\lambda(\epsilon)\}}{d\epsilon}\right|_{\epsilon=\epsilon_0} \neq 0$, *where* $\mathrm{Re}\,\lambda(\epsilon)$ *is the real part of* $\lambda(\epsilon)$.

Then, there exists a continuous function $\epsilon(\gamma)$ *with* $\epsilon(0) = \epsilon_0$, *and for all sufficiently small values of* $\gamma \neq 0$, *there exists a continuous family of non-constant periodic solution* $x(t,\gamma)$ *for the above dynamical system, which collapses to the equilibrium point* $x*(\epsilon_0)$ *as* $\gamma \to 0$. *The period of the cycle is close to* $2\pi/\mathrm{Im}\,\lambda(\epsilon_0)$, *where* $\mathrm{Im}\,\lambda(\epsilon_0)$ *is the imaginary part of* $\lambda(\epsilon_0)$.

The point $\epsilon = \epsilon_0$ that satisfies all of the above conditions (H1)–(H3) is called the "Hopf Bifurcation point." An important necessary condition for the occurrence of Hopf Bifurcation is that the characteristic equation of the above system has a pair of pure imaginary roots at $\epsilon = \epsilon_0$. It is well known that the typical continuous time dynamic optimization model with single state variable has two characteristic roots and at least one of which has positive real part, so that the Hopf Bifurcation cannot occur in such a model. But, [6, 11] proved analytically that the existence of Hopf Bifurcation is at least potentially possible if we consider the continuous time dynamic optimization model with two state variables.

Following [6, 11], let us consider the following typical continuous time dynamic optimization problem with two state variables.

$$\text{Maximize} \int_0^\infty F(k_1,k_2,u_1,u_2,\ldots,u_n)\mathrm{e}^{-rt}\mathrm{d}t \tag{1}$$

subject to

$$\dot{k}_1 = f(k_1,k_2,u_1,u_2,\ldots,u_n), \quad \dot{k}_2 = g(k_1,k_2,u_1,u_2,\ldots,u_n;\epsilon), \tag{2}$$

$$k_1(0) = k_{10} = \text{given}, \quad k_2(0) = k_{20} = \text{given}, \tag{3}$$

where $k_i (i = 1,2)$ are two state variables, $u_j (j = 1,2,\ldots,n)$ are control variables, r is the rate of discount that is a positive parameter, and ϵ is another parameter.[2] We assume that the functions F, f, and g are at least twice continuously differentiable.

We can solve this problem by means of Pontryagin's maximum principle (cf. [9, 12]). First, let us define the current value Hamiltonian as

$$\begin{aligned} H = {} & F(k_1,k_2,u_1,u_2,\ldots,u_n) + \mu_1 f(k_1,k_2,u_1,u_2,\ldots,u_n) \\ & + \mu_2 g(k_1,k_2,u_1,u_2,\ldots,u_n;\epsilon), \end{aligned} \tag{4}$$

[2] We can introduce other parameters which affect functions F and f, but the formulation in the text is sufficient for our purpose.

where μ_1 and μ_2 are two costate variables which correspond to two state variables k_1 and k_2, respectively. Then, a set of necessary conditions of the optimality becomes as follows.

$$\text{(i)} \qquad \dot{k}_i = \partial H/\partial \mu_i \quad (i=1,2)$$

$$\text{(ii)} \qquad \dot{\mu}_i = r\mu_i - \partial H/\partial k_i \quad (i=1,2)$$

$$\text{(iii)} \qquad \underset{(u_1,u_2,\ldots,u_n)}{\text{Max}} H \tag{5}$$

$$\text{(iv)} \qquad \lim_{t\to\infty} k_i \mu_i e^{-rt} = 0 \quad (i=1,2).$$

The conditions (5)(i) are equivalent to the dynamic constraints (2). The conditions (5)(ii) are a set of differential equations which describe the dynamics of the costate variables. We suppose that the conditions (5)(iii) are equivalent to the following first-order conditions[3]:

$$\partial H/\partial u_j = 0 \quad (j=1,2,\ldots,n). \tag{6}$$

This is a set of simultaneous equations with respect to the control variables. We assume that its solution is uniquely determined, and it can be expressed by the following continuously differentiable functions:

$$u_j = u_j(k_1,k_2,\mu_1,\mu_2;\epsilon) \ \ (j=1,2,\ldots,n). \tag{7}$$

The conditions (5)(iv) are called the "Transversality conditions."

Substituting the relationships (7) into (5)(i) and (5)(ii), we obtain the following four-dimensional system of linear or nonlinear differential equations:

$$\text{(i)} \qquad \dot{k}_1 = G_1(k_1,k_2,\mu_1,\mu_2;\epsilon)$$

$$\text{(ii)} \qquad \dot{k}_2 = G_2(k_1,k_2,\mu_1,\mu_2;\epsilon)$$

$$\text{(iii)} \qquad \dot{\mu}_1 = G_3(k_1,k_2,\mu_1,\mu_2;r,\epsilon)$$

$$\text{(iv)} \qquad \dot{\mu}_2 = G_4(k_1,k_2,\mu_1,\mu_2;r,\epsilon). \tag{8}$$

We shall consider the dynamics of this system around the equilibrium point by *assuming* that there exists a meaningful equilibrium solution $(k_1^*,k_2^*,\mu_1^*,\mu_2^*)$ of this system such that $\dot{k}_1 = \dot{k}_2 = \dot{\mu}_1 = \dot{\mu}_2 = 0$.

[3]We assume that the second-order conditions are also satisfied.

Let us write the (4×4) Jacobian matrix of this system *at the equilibrium point* as J. Then, we can write the characteristic equation of this system as

$$\Delta(\lambda) \equiv |\lambda I - J| = \lambda^4 + a_1\lambda^3 + a_2\lambda^2 + a_3\lambda + a_4 = 0, \tag{9}$$

$$a_1 = -\text{trace}J, \quad a_2 = M_2, \quad a_3 = -M_3, \quad a_4 = \det J, \tag{10}$$

where M_j is the sum of all principal j-th order minors of $J(j = 2,3)$. [4]

Dockner and Feichtinger [11] proved that the following relationships are satisfied in case of this particular Jacobian matrix J.

$$\text{trace}J = 2r, \quad -M_3 + rM_2 - r^3 = 0. \tag{11}$$

Following [11], let us write

$$K \equiv M_2 - r^2. \tag{12}$$

Then, we can rewrite Eq. (11) as

$$\text{trace}J = 2r, \quad -M_3 + rK = 0. \tag{13}$$

Then, we have the following expression substituting Eqs. (12) and (13) into a set of relationships (10).

$$a_1 = -\text{trace}J = -2r < 0, \quad a_2 = r^2 + K, \quad a_3 = -rK, \quad a_4 = \det J. \tag{14}$$

It is worth to note that we have

$$\text{trace}J = \sum_{j=1}^{4} \lambda_j = 2r > 0, \tag{15}$$

where $\lambda_j(j = 1,2,3,4)$ are the characteristic roots of Eq. (9). Therefore, this system has at least one root with positive real part.

Furthermore, [11] proved that the following set of conditions (DF) is equivalent to the condition (H2) in Theorem 1 in this chapter.

$$\det J > (K/2)^2, (K/2)^2 + r^2(K/2) - \det J = 0. \tag{DF}$$

More accurately, they proved the following quite useful theorem.

[4] See mathematical appendix of [2].

Theorem 2 ([11]). *The characteristic equation* $\Delta(\lambda) \equiv |\lambda I - J| = 0$ *of the particular Jacobian matrix J of the system (8) has the following properties (i)–(iv).*

(i) The characteristic equation has two positive real roots and two negative real roots if and only if

$$K < 0, \quad 0 < \det J \leqq (K/2)^2. \tag{16}$$

(ii) The characteristic equation has a pair of complex roots with positive real part and a pair of complex roots with negative real part if and only if

$$\det J > (K/2)^2, \quad \det J - (K/2)^2 - r^2(K/2) > 0. \tag{17}$$

(iii) The characteristic equation has three roots with positive real parts and one negative real root if and only if

$$\det J < 0. \tag{18}$$

(iv) The characteristic equation has a pair of complex roots with positive real part and a pair of pure imaginary roots if and only if the condition (DF) is satisfied.

Dockner and Feichtinger [11] expressed the result of this theorem visually by using Fig. 1.

Next, let us turn to the investigation of the conditions for the occurrence of the Hopf Bifurcation in a general system of nonlinear differential equations without restricting to the particular dynamic optimization model. It is worth noting that the following "Liu's theorem" provides us very powerful result that is applicable to general *n*-dimensional system of differential equations.

Theorem 3 ([18]). *Consider the following characteristic equation with* $n \geqq 3$ *:*

$$\lambda^n + b_1\lambda^{n-1} + b_2\lambda^{n-2} + \cdots + b_{n-1}\lambda + b_n = 0. \tag{19}$$

This characteristic equation has a pair of pure imaginary roots and $(n-2)$ *roots with negative real parts if and only if the following set of conditions are satisfied :*

$$A_j > 0 \quad \textit{for all} \quad j \in \{1, 2, \ldots, n-2\}, \ A_{n-1} = 0, \ b_n > 0, \tag{20}$$

where $A_j (j = 1, 2, \ldots, n-1)$ *are Routh-Hurwitz terms defined as*

$$A_1 = b_1, \quad A_2 = \begin{vmatrix} b_1 & b_3 \\ 1 & b_2 \end{vmatrix}, \quad A_3 = \begin{vmatrix} b_1 & b_3 & b_5 \\ 1 & b_2 & b_4 \\ 0 & b_1 & b_3 \end{vmatrix}, \ldots,$$

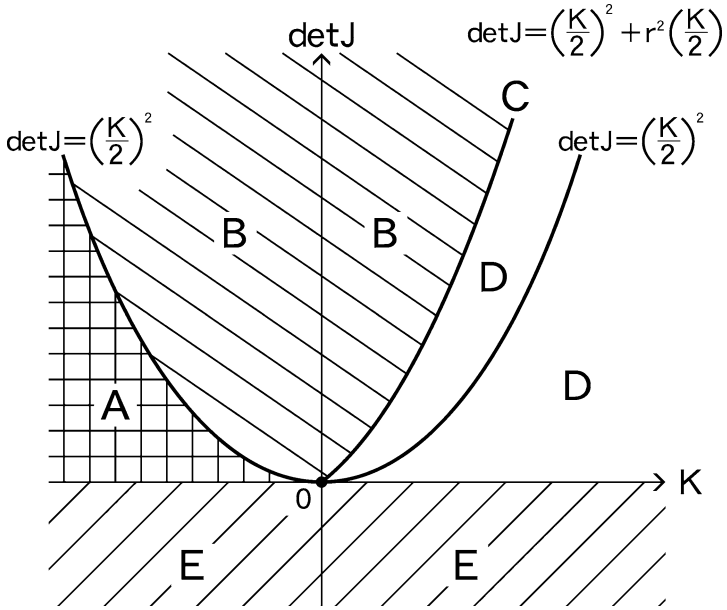


Fig. 1 Classification of the nature of the roots of characteristic equation (9). (A) Two Positive real roots and two negative real roots (real roots type saddle point). (B) A pair of complex roots with positive real part and a pair of complex roots with negative real part (complex roots type saddle point). (C) A pair of complex roots with positive real part and a pair of pure imaginary roots (Hopf Bifurcation curve). (D) Four roots with positive real parts(totally unstable). (E) Three parts with positive real parts and one negative real root. (*Source*: Dockner and Feichtinger (1991), p. 36; Feichtinger et al. (1994), p. 356)

$$A_{n-1} = \begin{vmatrix} b_1 & b_3 & b_5 & b_7 & \cdots & 0 & 0 \\ 1 & b_2 & b_4 & b_6 & \cdots & 0 & 0 \\ 0 & b_1 & b_3 & b_5 & \cdots & 0 & 0 \\ 0 & 1 & b_2 & b_4 & \cdots & 0 & 0 \\ 0 & 0 & b_1 & b_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & b_n & 0 \\ 0 & 0 & 0 & 0 & \cdots & b_{n-1} & 0 \\ 0 & 0 & 0 & 0 & \cdots & b_{n-2} & b_n \\ 0 & 0 & 0 & 0 & \cdots & b_{n-3} & b_{n-1} \end{vmatrix}. \tag{21}$$

Although this "Liu's theorem" is quite useful in the sense that it can be applicable to the general n-dimensional system of differential equations, it has the following deficiency. The Hopf Bifurcation in which all the characteristic roots *except* a pair of purely imaginary ones have *negative* real parts is called the "simple" Hopf Bifurcation. Liu's theorem is applicable only to the case of "simple" Hopf Bifurcation. But, in the typical dynamic optimization model, usually there exists at least one characteristic root that has positive real part. This means that Liu's theorem is inapplicable to the typical dynamic optimization model. On the other hand, [6] provided the following complete mathematical characterization of the criteria for the occurrence of the Hopf Bifurcation including the "non simple" as well as the "simple" case, although their analysis is restricted to four-dimensional system.[5]

Theorem 4 ([6]). *(1) Consider the characteristic equation*

$$\lambda^4 + b_1\lambda^3 + b_2\lambda^2 + b_3\lambda + b_4 = 0. \tag{22}$$

(i) The characteristic equation (22) has a pair of pure imaginary roots and two roots with nonzero real parts if and only if either of the following set of conditions (A) or (B) is satisfied

$$b_1 b_3 > 0, \quad b_4 \neq 0, \quad \Phi \equiv b_1 b_2 b_3 - b_1^2 b_4 - b_3^2 = 0. \tag{A}$$

$$b_1{=}b_3{=}0, \quad b_4 < 0. \tag{B}$$

(ii) The characteristic equation (22) has a pair of pure imaginary roots and two roots with negative real parts if and only if the following condition (C) is satisfied.

$$b_1 > 0, \quad b_3 > 0, \quad b_4 > 0, \quad \Phi \equiv b_1 b_2 b_3 - b_1^2 b_4 - b_3^2 = 0. \tag{C}$$

(2) Consider the characteristic equation

$$\lambda^4 + b_1(\epsilon)\lambda^3 + b_2(\epsilon)\lambda^2 + b_3(\epsilon)\lambda + b_4(\epsilon) = 0, \tag{23}$$

where it is assumed that the coefficients $b_j(j = 1,2,3,4)$ are the continuously differentiable functions of a parameter ϵ. Then, we have the following properties (i) and (ii).

(i) Suppose that we have $b_1(\epsilon_0)b_3(\epsilon_0) > 0, b_4(\epsilon_0) \neq 0$, and

$\Phi(\epsilon_0) \equiv b_1(\epsilon_0)b_2(\epsilon_0)b_3(\epsilon_0) - b_1(\epsilon_0)^2 b_4(\epsilon_0) - b_3(\epsilon_0)^2 = 0$ *at the point $\epsilon = \epsilon_0$.*

[5] Theorem 4(1) was referred to by ([15], p. 483) as "Asada-Yoshida Theorem".

Then, the condition (H3) in Theorem 1 is equivalent to the following condition (D).

$$\left.\frac{d\Phi(\epsilon)}{d\epsilon}\right|_{\epsilon=\epsilon_0} \neq 0 \tag{D}$$

(ii) Suppose that we have $b_1(\epsilon_0) = 0$, $b_3(\epsilon_0) = 0$, *and* $b_4(\epsilon_0) < 0$ *at the point* $\epsilon = \epsilon_0$. *Then, the condition (H3) in Theorem 1 is equivalent to the following condition (E).*

$$\left[b_2(\epsilon_0) + \sqrt{b_2(\epsilon_0)^2 - 4b_4(\epsilon_0)}\right] b_1'(\epsilon_0) - 2b_3'(\epsilon_0) \neq 0. \tag{E}$$

Asada and Yoshida [6] proved the following proposition by applying Theorem 4(1)(i) to the particular characteristic equation (9).[6]

Proposition 1 ([6]). *(i) The characteristic equation (9) of the particular system of differential equations (8) has a set of pure imaginary roots and two roots with nonzero real parts if and only if the following set of conditions (AY) is satisfied:*

$$K > 0, \quad (K/2)^2 + r^2(K/2) - \det J = 0. \tag{AY}$$

(ii) A set of conditions (AY) is equivalent to a set of conditions (DF) by [11].

Proof. (i) First, it follows from the relationships (14) that

$$\Phi \equiv a_1a_2a_3 - a_1^2a_4 - a_3^2 = 4r^2[(K/2)^2 + r^2(K/2) - \det J]. \tag{24}$$

Second, a set of conditions (A) in Theorem 4(1)(i) is equivalent to the following set of conditions in case of the particular characteristic equation (9).

$$a_3 < 0, \quad a_4 \neq 0, \quad \Phi = 0. \tag{25}$$

We can see from the relationships (14) that the condition $a_3 < 0$ is equivalent to the condition $K > 0$. Furthermore, the condition $\Phi = 0$ is equivalent to the condition $(K/2)^2 + r^2(K/2) - \det J = 0$. If these two conditions are satisfied, we also have $a_4 \neq 0$ because of the fact that $a_4 = \det J = (K/2)^2 + r^2(K/2) > 0$.

(ii) First, let us suppose that a set of conditions (DF) is satisfied. In this case, we have

$$\det J = (K/2)^2 + r^2(K/2) > (K/2)^2, \tag{26}$$

which means that $K > 0$. This proves the causality (DF) $\Longrightarrow$(AY).

[6] We reproduce the proof here. The method of proof is quite simple and straightforward.

Next, let us suppose that a set of conditions (AY) is satisfied. Also in this case, we have the relationship (26), which means that a set of conditions (DF) is satisfied. This proves the causality (AY)$\Longrightarrow$(DF). □

Remark 1. Comparing Theorem 2(iv) and Proposition 1, we can see that the particular characteristic equation (9) has a pair of pure imaginary roots and two complex roots with positive real parts if a set of conditions (AY) is satisfied. In this case, the condition (D) in Theorem 4(2)(i) is equivalent to the condition

$$\frac{\mathrm{d}}{\mathrm{d}\epsilon}[(K/2)^2 + r^2(K/2) - \det J]\Big|_{\epsilon=\epsilon_0} \neq 0. \tag{27}$$

In the next section, we shall apply the analytical results which were surveyed in this section to an extended version of [16] model of dynamic limit pricing.

3 An Application to Dynamic Limit Pricing

3.1 Gaskins' Prototype Model of Dynamic Limit Pricing

First, let us summarize the prototype model of dynamic limit pricing that was originated by [16]. We consider a partial equilibrium model of an industry in which one dominant large firm and many small fringe firms exist. The demand function is expressed by the following linear decreasing function:

$$q = a - bp \quad ; \quad a > 0,\ b > 0, \tag{28}$$

where q is the demand for the product of this industry, p is the price of this product., and a, b are two parameters of the demand function.[7]

The dominant large firm acts as the price leader (the price setter) subject to the threat of entry by the fringe firms. Fringe firms behave as price takers and the entry dynamics of the fringe firms are expressed by the differential equation

$$\dot{x} = \alpha(p - \bar{p}) \quad ; \quad \alpha{>}0,\ \bar{p} > 0, \tag{29}$$

where x is the total output of fringe firms and α, $\bar{p}$ are parameters of the entry dynamics. It is assumed that the dominant large firm selects its output level corresponding to $(q - x)$ and the average cost of the dominant large firm (c) is constant such that $0 < c < \bar{p}$. Then, the discounted present value of the dominant large firm becomes

[7] Gaskins [16] used more general demand function that is not necessarily linear, but we use the linear demand function for simplicity of the analysis following [10] Chap. 10.

$$W = \int_0^{\infty} (p-c)(a-bp-x)\mathrm{e}^{-rt}\mathrm{d}t, \tag{30}$$

where r is the rate of discount, which is a positive parameter.

The dominant large firm is supposed to select the dynamic path (p) of price that maximizes W subject to the dynamic constraint (29) and given initial value $x(0)$. Although this is a typical dynamic optimization problem of single agent with one state variable (x), we can interpret that this is implicitly a kind of Stackerberg differential game in which the dominant large firm acts as the leader and fringe firms act as followers (cf. [4]).[8]

The current value Hamiltonian of this dynamic optimization problem can be written as

$$H = (p-c)(a-bp-x) + \mu\alpha(p-\bar{p}), \tag{31}$$

where μ is the costate variable corresponding to the dynamic constraint (29). A set of necessary conditions for optimality becomes as

$$\begin{aligned}
&\text{(i)} && \dot{x} = \partial H/\partial\mu, \\
&\text{(ii)} && \dot{\mu} = r\mu - \partial H/\partial x, \\
&\text{(iii)} && \underset{p}{\mathrm{Max}} H, \\
&\text{(iv)} && \lim_{t\to\infty} x\mu\mathrm{e}^{-rt} = 0.
\end{aligned} \tag{32}$$

Solving Eq. (32)(iii) with respect to μ, we have $\mu = \mu(p)$. Substituting this relationship into equations (i) and (ii) in (32), we obtain the following two dimensional system of differential equations with single transversality condition, where the initial value of the state variable $x(0)$ is predetermined, but the initial value of the control variable $p(0)$ is *not* predetermined.

$$\begin{aligned}
&\text{(i)} && \dot{x} = F_1(p) \\
&\text{(ii)} && \dot{p} = F_2(x,p) \\
&\text{(iii)} && \lim_{t\to\infty} x\mu(p)\mathrm{e}^{-rt} = 0.
\end{aligned} \tag{33}$$

[8] As for the exhaustive exposition of the theory of differential game, see [12].

Gaskins [16] proved that the economically meaningful equilibrium point such that $\dot{x} = \dot{p} = 0$ exists under some reasonable conditions, and it becomes a saddle point, namely, the (2×2) Jacobian matrix of this system at the equilibrium point has one positive real root and one negative real root. This means that there exists only one initial value $p(0)$ that ensures the convergence to the equilibrium point corresponding to the given initial value $x(0)$. Only the convergent path satisfies the transversality condition (33)(iii).

In sum, in Gaskins' prototype model, cyclical fluctuations do not occur, but only the monotonic convergence to the equilibrium point occurs.

3.2 *Cyclical Fluctuations in an Extended Gaskins Model of Dynamic Limit Pricing*

It is possible to extend and develop Gaskins' prototype model in several ways. For example, [4, 17] extended Gaskins model by introducing the investment behaviors of firms. In particular, [4] provided an example of the occurrence of cyclical fluctuations in such an extended model by means of numerical simulations. In this subsection, we shall present another simple extension of Gaskins model that can produce cyclical fluctuations, which is an example of the direct application of the analytical results summarized in Sect. 2 of this chapter.

Instead of the dynamic constraint (29), let us adopt the following new formulation.

$$\dot{x} = \alpha(p^e - \bar{p}) \quad ; \quad \alpha>0,\ \bar{p} > 0, \tag{34}$$

$$\dot{p^e} = \beta(p - p^e) \quad ; \quad \beta > 0, \tag{35}$$

where p^e is the expected price, which is the price expected by fringe firms. Equation (35) means that the dynamic of expected price is governed by a formula of adaptive expectation hypothesis, and β is the speed of adaptation that can be interpreted as the *reciprocal* of the average time lag of expectation adaptation.[9]

The dynamic optimization problem of the dominant large firm is to select the dynamic path of price (p) that maximizes W in Eq. (30) subject to two dynamic constraints (34), (35) with given initial values of two state variables $x(0)$ and $p^e(0)$. In this case, the current value Hamiltonian becomes

$$H = (p-c)(a-bp-x) + \mu_1\alpha(p^e - \bar{p}) + \mu_2\beta(p - p^e), \tag{36}$$

[9]In the appendix, we reinterpret this equation by means of a continuously distributed lag model of expectation formation.

where μ_1 and μ_2 are two costate variables which correspond to two state variables x and p^e, respectively.

A set of necessary conditions for optimality becomes

$$\text{(i)} \qquad \dot{x} = \partial H/\partial \mu_1 = \alpha(p^e - \bar{p}),$$

$$\text{(ii)} \qquad \dot{p}^e = \partial H/\partial \mu_2 = \beta(p - p^e),$$

$$\text{(iii)} \qquad \dot{\mu}_1 = r\mu_1 - \partial H/\partial x = r\mu_1 + p - c,$$

$$\text{(iv)} \qquad \dot{\mu}_2 = r\mu_2 - \partial H/\partial p^e = (r+\beta)\mu_2 - \mu_1\alpha,$$

$$\text{(v)} \qquad \operatorname*{Max}_p H,$$

$$\text{(vi)} \qquad \lim_{t\to\infty} x\mu_1 \mathrm{e}^{-rt} = 0, \quad \lim_{t\to\infty} p^e \mu_2 \mathrm{e}^{-rt} = 0. \tag{37}$$

Now, let us turn to the condition (37)(v). The first-order condition for the maximization of H with respect to p becomes[10]

$$\partial H/\partial p = -2bp + a - x + bc + \mu_2\beta = 0. \tag{38}$$

Solving this equation with respect to p, we have

$$p = \frac{1}{2b}(a - x + bc + \mu_2\beta). \tag{39}$$

Substituting Eq. (39) into Eq. (37)(i)–(iv), we obtain the following four-dimensional system of linear differential equations:

$$\text{(i)} \qquad \dot{x} = \alpha(p^e - \bar{p}) \equiv G_1(p^e; \alpha)$$

$$\text{(ii)} \qquad \dot{p}^e = \beta\{\frac{1}{2b}(a - x - bc + \mu_2\beta) - p^e\} \equiv G_2(x, p^e, \mu_2; \beta)$$

$$\text{(iii)} \qquad \dot{\mu}_1 = r\mu_1 + \frac{1}{2b}(a - x + bc + \mu_2\beta) - c \equiv G_3(x, \mu_1, \mu_2; r, \beta)$$

$$\text{(iv)} \qquad \dot{\mu}_2 = (r+\beta)\mu_2 - \mu_1\alpha \equiv G_4(\mu_1, \mu_2; r, \alpha, \beta). \tag{40}$$

Next, we shall consider the nature of the equilibrium solution $(x^*, p^{e*}, p^*, \mu_1^*, \mu_2^*)$ that satisfies $\dot{x} = \dot{p}^e = \dot{\mu}_1 = \dot{\mu}_2 = 0$. It is easy to see that we have

$$p^{e*} = p^* = \bar{p} > 0. \tag{41}$$

[10] Since $\partial^2 H/\partial p^2 = -2b < 0$, the second-order condition is always satisfied.

Other three equilibrium values are determined by the following linear system of equations.

$$\begin{bmatrix} -1 & 0 & \beta \\ -1 & 2br & \beta \\ 0 & -\alpha & r+\beta \end{bmatrix} \begin{bmatrix} x \\ \mu_1 \\ \mu_2 \end{bmatrix} = \begin{bmatrix} 2b\bar{p}+bc-a \\ bc-a \\ 0 \end{bmatrix}. \tag{42}$$

It is easy to see that the solution of this system of equations becomes

$$x^* = \frac{(a-2b\bar{p}-bc)r(r+\beta)-\bar{p}\alpha\beta}{r(r+\beta)} = (a-2b\bar{p}-bc) - \frac{\bar{p}\alpha\beta}{r(r+\beta)}$$
$$< a-2b\bar{p}-bc, \tag{43}$$

$$\mu_1^* = \frac{-\bar{p}}{r} < 0, \tag{44}$$

$$\mu_2^* = \frac{-\alpha\bar{p}}{r(r+\beta)} = \frac{\alpha\mu_1^*}{r+\beta} < 0. \tag{45}$$

Proposition 2. *We have $x^* > 0$ for all $\beta > 0$ if the parameter a (upper limit of demand) is fixed at sufficiently large positive value and the parameter α (adjustment speed of entry) is fixed at sufficiently small positive value.*

Proof. It is easy to see that we have $x^* > 0$ *if and only if* the inequality

$$Z(\beta) \equiv (a-2b\bar{p}-bc)r(r+\beta)-\bar{p}\alpha\beta > 0 \tag{46}$$

is satisfied. Incidentally, we have

$$Z(0) = (a-2b\bar{p}-bc)r^2, \tag{47}$$

$$Z'(\beta) = (a-2b\bar{p}-bc)r-\bar{p}\alpha. \tag{48}$$

Therefore, we have $Z(0) > 0$ and $Z'(\beta) > 0$ if a is sufficiently large and α is sufficiently small. In this case, we obtain $Z(\beta) > 0$ for all $\beta > 0$, which means that we have $x^* > 0$ for all $\beta > 0$. □

Now, let us study the dynamic property of this model by assuming as follows.

Assumption 1. The combination of the parameter values (a, α) is at the level such that $x^* > 0$ for all $\beta > 0$.

The Jacobian matrix of this system becomes

$$J = \begin{bmatrix} 0 & \alpha & 0 & 0 \\ -\frac{\beta}{2b} & -\beta & 0 & \frac{\beta^2}{2b} \\ -\frac{1}{2b} & 0 & r & \frac{\beta}{2b} \\ 0 & 0 & -\alpha & r+\beta \end{bmatrix}. \tag{49}$$

We can write the characteristic equation of this system as

$$\Delta(\lambda) \equiv |\lambda I - J| = \lambda^4 + a_1\lambda^3 + a_2\lambda^2 + a_3\lambda + a_4 = 0, \tag{50}$$

where

$$a_1 = -\text{trace}J = -2r < 0, \tag{51}$$

$$\begin{aligned} a_2 &= M_2 = \text{sum of all principal second-order minors of } J \\ &= \begin{vmatrix} 0 & \alpha \\ -\frac{\beta}{2b} & -\beta \end{vmatrix} + \begin{vmatrix} 0 & 0 \\ -\frac{1}{2b} & r \end{vmatrix} + \begin{vmatrix} 0 & 0 \\ 0 & r+\beta \end{vmatrix} \\ &\quad + \begin{vmatrix} -\beta & 0 \\ 0 & r \end{vmatrix} + \begin{vmatrix} -\beta & \frac{\beta^2}{2b} \\ 0 & r+\beta \end{vmatrix} + \begin{vmatrix} r & \frac{\beta}{2b} \\ -\alpha & r+\beta \end{vmatrix} \\ &= r^2 + \beta\left(\frac{\alpha}{b} - r - \beta\right), \end{aligned} \tag{52}$$

$$a_3 = -M_3 = -(\text{sum of all principal third-order minors of } J), \tag{53}$$

$$a_4 = \det J = \frac{\alpha\beta r(r+\beta)}{2b} \equiv \det J(\beta) > 0. \tag{54}$$

Since this dynamic optimization model with two state variables is only a particular case of the model that was explained in Sect. 2, we can apply Theorem 2 in Sect. 2 to this model. To this purpose, let us consider the following three relationships.

$$K \equiv M_2 - r^2 = -\beta^2 + (\frac{\alpha}{b} - r)\beta \equiv K(\beta), \tag{55}$$

$$\begin{aligned} \Omega(\beta) &\equiv (K/2)^2 - \det J \\ &= \frac{\beta}{2}\left[\frac{1}{2}\beta^3 + \left(r - \frac{\alpha}{b}\right)\beta^2 + \left\{\frac{1}{2}\left(r - \frac{\alpha}{b}\right)^2 - \frac{\alpha r}{b}\right\}\beta - \frac{\alpha r^2}{b}\right], \end{aligned} \tag{56}$$

$$\begin{aligned} \Psi(\beta) &\equiv (K/2)^2 + r^2(K/2) - \det J \\ &= \beta\left[\beta^3 + \frac{1}{2}\left(r - \frac{\alpha}{b}\right)\beta^2 + \frac{\alpha}{b}\left(\frac{\alpha}{b} - 4r\right)\beta - r^3\right]. \end{aligned} \tag{57}$$

Now, we can prove the following important results by applying Dockner and Feichtinger's theorem (Theorem 2 in Sect. 2).

Proposition 3. *Suppose that* $0 < r < \frac{\alpha}{b}$.
Then, we have the following properties (i)–(ii).

(i) The characteristic equation (50) has a pair of complex roots with positive real part and a pair of complex roots with negative real part for all sufficiently small values of $\beta > 0$.
(ii) Equation (50) has two positive real roots and two negative real roots for all sufficiently large values of $\beta > 0$.

Proof. Suppose that $0 < r < \frac{\alpha}{b}$. Then, the function $K(\beta)$ becomes a differentiable function that has the following property (P_1) because of Eq. (55).

$$\begin{aligned} K(0) = 0, \quad K'(\beta) > 0 \quad \text{for all } \beta \in \left[0, \frac{\alpha/b - r}{2}\right), \\ K'\left(\frac{\alpha/b - r}{2}\right) = 0, \quad K'(\beta) < 0 \quad \text{for all } \beta \in \left(\frac{\alpha/b - r}{2}, \infty\right), \\ K\left(\frac{\alpha}{b} - r\right) = 0, \quad \lim_{\beta \to \infty} K(\beta) = -\infty. \end{aligned} \tag{P_1}$$

On the other hand, the functions $\det J(\beta)$, $\Omega(\beta)$ and $\Psi(\beta)$ become the differential functions which have the following properties (P_2)–(P_4) because of the Eqs. (54), (56), and (57):

$$\det J(0) = 0, \quad \det J'(\beta) > 0 \quad \text{for all } \beta \in [0, \infty), \quad \lim_{\beta \to \infty} \det J(\beta) = \infty. \tag{P_2}$$

$$\Omega(0) = 0, \quad \Omega'(0) = -\frac{\alpha r^2}{2b} < 0, \quad \lim_{\beta \to \infty} \frac{\Omega(\beta)}{\beta^4} = \frac{1}{4} > 0. \tag{P_3}$$

$$\Psi(0) = 0, \quad \Psi'(0) = -r^3 < 0. \tag{P_4}$$

These properties (P_1)–(P_4) imply the following results.

The combination $(K, \det J)$ is located at the origin of Fig. 1 when $\beta = 0$. As β increases from $\beta = 0$, this combination moves to the north-east direction continuously until it reaches the point $\beta = \frac{\alpha/b - r}{2}$, and the property ($P_4$) implies that this combination is located at the region B of Fig. 1 for all sufficiently small values of $\beta > 0$. After the point $\beta = \frac{\alpha/b - r}{2}$ this combination moves to the north-west direction continuously and indefinitely according as the further increase of β. At the point $\beta = \frac{\alpha}{b} - r$ this combination is located at the vertical axis of Fig. 1. On the other hand, $\lim_{\beta \to \infty} \frac{\Omega(\beta)}{\beta^4} > 0$ implies that $\Omega(\beta)$ becomes positive for all sufficiently large values of $\beta > 0$. This means that the combination is located at the region A of Fig. 1 for all sufficiently large values of $\beta > 0$. □

Proposition 4. *Suppose that* $0 < r < \frac{\alpha}{b}$ *and* r *is sufficiently small. Then, there exist the parameter values* $B_j (j = 1, 2, 3, 4)$ *such that* $0 < \beta_1 < \frac{\alpha/b - r}{2} < \beta_2 < \frac{\alpha}{b} - r < \beta_3 \leqq \beta_4 < \infty$ *which satisfy the following properties (i)–(iv).*

(i) The characteristic equation (50) has a pair of complex roots with positive real part and a pair of complex roots with negative real part for all $\beta \in (0, \beta_1) \smile (\beta_2, \beta_3)$.

(ii) Equation (50) has four roots with positive real parts for all $\beta \in (\beta_1, \beta_2)$.

(iii) Equation (50) has a pair of complex roots with positive real part and a pair of pure imaginary roots at two points $\beta = \beta_1$ *and* $\beta = \beta_2$.

(iv) Equation (50) has two positive real roots and two negative real roots for all $\beta \in [\beta_4, \infty)$.

Proof. Suppose that $0 < r < \frac{\alpha}{b}$. In this case, it follows from the method of the proof of Proposition 3 that there exist the parameter values $\beta_j (j = 1,2,3)$ such that $0 < \beta_1 < \frac{\alpha/b-r}{2} < \beta_2 < \frac{\alpha}{b} - r < \beta_3$ with the properties that (i) the trajectory of the combination $(K, \det J)$ is located at the region B in Fig. 1 for all $\beta \in (0, \beta_1) \smile (\beta_2, \beta_3)$, (ii) it is located at the region D in Fig. 1 for all $\beta \in (\beta_1, \beta_2)$, and (iii) it crosses the curve C at two points $\beta = \beta_1$ and $\beta = \beta_2$, *if and only if* the inequality $\Psi(\frac{\alpha/b-r}{2}) > 0$ is satisfied, where we have

$$\Psi\left(\frac{\alpha/b-r}{2}\right) = \left(\frac{\alpha/b-r}{2}\right)\left[\left(\frac{\alpha/b-r}{2}\right)^3 - \frac{1}{2}\left(\frac{\alpha}{b}-r\right)\left(\frac{\alpha/b-r}{2}\right)^2 + \left(\frac{\alpha}{b}\right)\left(\frac{\alpha}{b}-4r\right)\left(\frac{\alpha/b-r}{2}\right) - r^3 2\right] \tag{58}$$

from Eq. (57). It follows from Eq. (58) that

$$\lim_{r\to 0}\Psi\left(\frac{\alpha/b-r}{2}\right) = \frac{9}{2}\left(\frac{\alpha}{2b}\right)^4 > 0, \tag{59}$$

which means that we have $\Psi(\frac{\alpha/b-r}{2}) > 0$ for all sufficiently small values of $r > 0$ by continuity. This proves (i)–(iii) of Proposition 4. Proposition 4 (iv) directly follows from Proposition 3. □

Proposition 5. *Suppose that* $r \geqq \frac{\alpha}{b}$. *Then, there exists a parameter value* $\beta_0 \in (0, \infty)$ *that satisfy the following properties (i)–(ii).*

(i) The characteristic equation (50) has a pair of complex roots with positive real part and a pair of complex roots with negative real part for all $\beta \in (0, \beta_0)$.

(ii) Equation (50) has two positive real roots and two negative real roots for all $\beta \in [\beta_0, \infty)$.

Proof. Suppose that $r \geqq \frac{\alpha}{b}$. Then, the differentiable function $K(\beta)$ has the following property (P_1').

$$\begin{aligned} &K(0) = 0, \quad K'(0) = 0, \quad K'(\beta) < 0 \quad \text{for all } \beta > 0, \\ &\lim_{\beta\to\infty} K(\beta) = -\infty \quad \text{if } r = \frac{\alpha}{b}, \text{ and,} \\ &K(0) = 0, \quad K'(\beta) < 0 \quad \text{for all } \beta \geqq 0, \\ &\lim_{\beta\to\infty} K(\beta) = -\infty \quad \text{if } r > \frac{\alpha}{b}. \end{aligned} \tag{P_1'}$$

On the other hand, the properties (P_2) and (P_3) in the proof of Proposition 3 apply also in this case.

The properties (P_1') and (P_2) mean that the combination $(K, \det J)$ is located at the origin of Fig. 1 when $\beta = 0$, and this combination moves to the north-west direction continuously and indefinitely as β increases. The property (P_3) implies that this combination is located at the region B of Fig. 1 for all sufficiently small values of $\beta > 0$, and it is located at the region A of Fig. 1 for all sufficiently large values of $\beta > 0$. This means that there exists a parameter value $\beta_0 \in (0, \infty)$ that satisfy the property (i) of Proposition 5, and we have

$$\Omega(\beta_0) = 0, \quad \Omega'(\beta_0) = \frac{\beta_0}{2}\left[\frac{3}{2}\beta_0^2 + 2\left(r - \frac{\alpha}{b}\right)\beta_0 + \frac{1}{2}\left(r - \frac{\alpha}{b}\right)^2 - \frac{\alpha r}{b}\right] > 0. \quad (60)$$

In other words, the switching of the regions $B \longrightarrow A$ (we call it "forward switching") occurs at the point $\beta = \beta_0$. Next, let us consider whether the "backward switching" (the switching of the regions $A \longrightarrow B$) occurs according as the further increase of the parameter value β. For this purpose, let us suppose tentatively that there exists another switching point $\beta^* \in (\beta_0, \infty)$ such that

$$\Omega(\beta^*) = 0, \quad \Omega'(\beta*) = \frac{\beta*}{2}\left[\frac{3}{2}\beta *^2 + 2\left(r - \frac{\alpha}{b}\right)\beta * + \frac{1}{2}\left(r - \frac{\alpha}{b}\right)^2 - \frac{\alpha r}{b}\right]. \quad (61)$$

Comparing Eqs. (60) and (61), we can see that $\beta^* > \beta_0 > 0$ and $r \geqq \frac{\alpha}{b}$ imply

$$\Omega'(\beta^*) > \Omega'(\beta_0) > 0, \quad (62)$$

which contradicts that the point $\beta = \beta^*$ is a "backward" switching point, because at the "backward" switching point the inequality $\Omega'(\beta) < 0$ must be satisfied. This proves that the "backward switching" cannot occur so that the property (ii) of Proposition 5 is satisfied in case of $r \geqq \frac{\alpha}{b}$. □

Figure 2 summarizes the results of Propositions 3–5. In the regions (B) and at the points (C) in this figure, the cyclical fluctuations occur. In the next section, we shall try to provide an interpretation of the analytical results obtained in this section.

4 An Interpretation of the Analytical Results

Figure 2 provides us a convenient characterization of the solution of the extended dynamic limit pricing model that was presented in Sect. 3.2. This figure shows that the characteristic equation of this system has two positive real roots and two negative real roots (regions (A) in this figure) irrespective of the value of the rate of discount $r > 0$ if the adjustment speed of adaptive expectation $\beta > 0$ is sufficiently large (if the time lag of the expectation adaptation $\tau = 1/\beta$ is sufficiently small). In this case, the equilibrium point of the system becomes a real roots type saddle point, and the

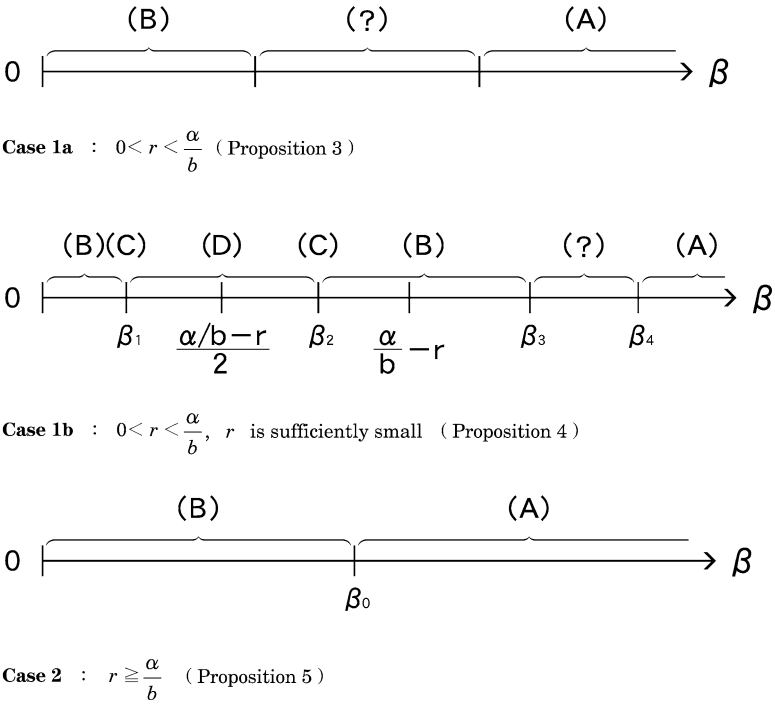


Fig. 2 Classification of the nature of the roots of characteristic equation (50)

number of the positive roots is equal to the number of the not-pre-determined costate variables in a system of four-dimensional linear differential equations (40). This means that the dominant firm can select the initial values of the costate variables which ensure the monotonic convergence to the equilibrium point. *If and only if* the convergent path is selected, the transversality conditions (37) (vi) are satisfied. This situation is illustrated in Fig. 3.[11] It is worth noting that the solution path in Fig. 3 is qualitatively the same as that of [16] original model of dynamic limit pricing that was explained in Sect. 3.1, which can be considered to be the limit case of $\beta \to \infty$ $(\tau \to 0)$.

Figure 2 also shows that the characteristic equation of this system has a pair of complex roots with positive real part and a pair of complex roots with negative real part (regions (B) in this figure) irrespective of the value of $r > 0$ if $\beta > 0$ is sufficiently small (if $\tau = 1/\beta$ is sufficiently large). In this case, the equilibrium point becomes a complex roots type saddle point, and also in this case the number of the roots with positive real parts is equal to the number of the not-pre-determined costate variables. Therefore, also in this situation the dominant firm can select the

[11]Note that Eq. (39) means that the initial value of price $p(0)$ is determined if the initial value of a state variable $x(0)$ is given and the initial value of a costate variable $\mu_2(0)$ is selected.

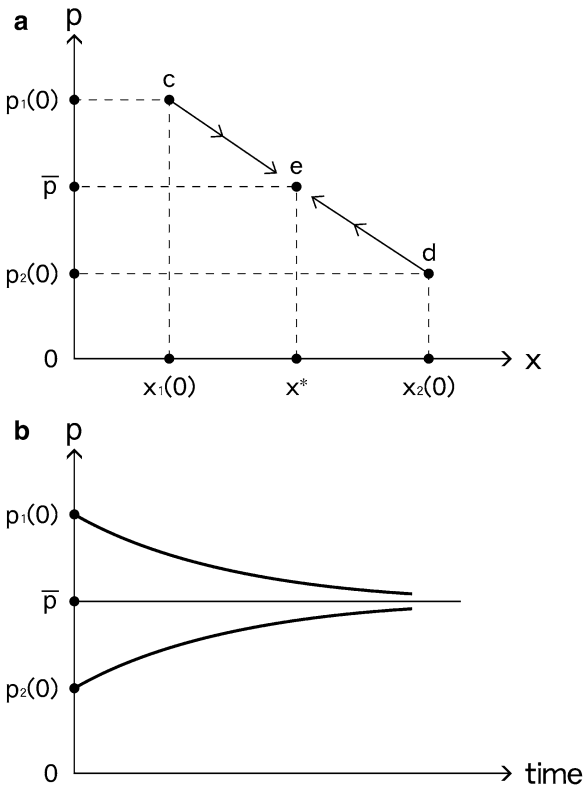


Fig. 3 Monotonic convergence (region (A) in Fig. 2)

convergent path, which satisfies the transversality conditions. In this case, however, the cyclical fluctuations occur even if the dominant firm selects the convergent path. This situation is illustrated in Fig. 4.

Case 1b of Fig. 2 provides us an additional important information in case of the sufficiently small values of the rate of discount $r > 0$. In this case, the region of cyclical convergence (B) is interrupted by the region (D) at which the characteristic equation has four roots with positive real parts. If the parameter values are located at the region (D), it is impossible to satisfy the transversality conditions unless the initial values of two state variables are given at the equilibrium levels. In this case, a system of four-dimensional linear differential equations (40) fails to characterize the optimal solution.

Next, let us pay attention to two boundary points between the regions (B) and (D) in Case 1b of Fig. 2, namely, the points $\beta_1$and β_2. At these points, the characteristic equation has a pair of complex roots with positive real parts and a pair of pure imaginary roots. These points correspond to the (degenerated) Hopf Bifurcation points in a system of linear differential equations. Also in this case, the number of the roots with positive real parts is equal to the number of the not-predetermined costate variables. Hence, the dominant firm can select the non-divergent

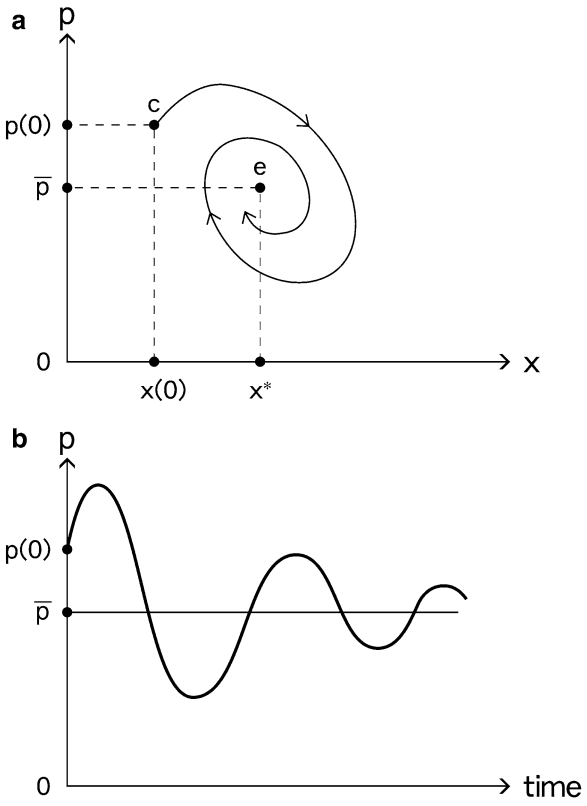


Fig. 4 Cyclical convergence (region (B) in Fig. 2)

dynamic path. In this case, however, the non-divergent path does not converge to the equilibrium point, but it becomes a closed orbit around the equilibrium point. The combination (p,x) continues to move along the closed orbit without becoming nonpositive if the initial values of the state variables are not extremely far from the equilibrium point, and the dynamic path along the closed orbit satisfies the transversality conditions (37)(vi). This means that the closed orbit becomes the optimal path in this case. This situation is illustrated in Fig. 5.

It must be noted that the Hopf Bifurcations in this model are "degenerated" types because of the linearity of the dynamic system. This means that the probability of the occurrence of the closed orbit becomes "measure zero" in the half line β in Case 1b of Fig. 2. Nevertheless, the (converging) cyclical fluctuations occur at the wide range of the parameter value $\beta > 0$ in this extended dynamic limit pricing model.

Acknowledgements This chapter is based on the paper that was written in March 2010 while the author was staying at School of Finance and Economics, University of Technology Sydney (UTS) as a visiting professor under the "Chuo University Leave Program for Special Research Project", and an earlier version of this chapter was tentatively published as Discussion Paper Series No. 139 of the Institute of Economic Research, Chuo University, Tokyo, Japan (April 2010). This research was financially supported by the Japan Society for the promotion of Science (Grant-in-Aid (C)

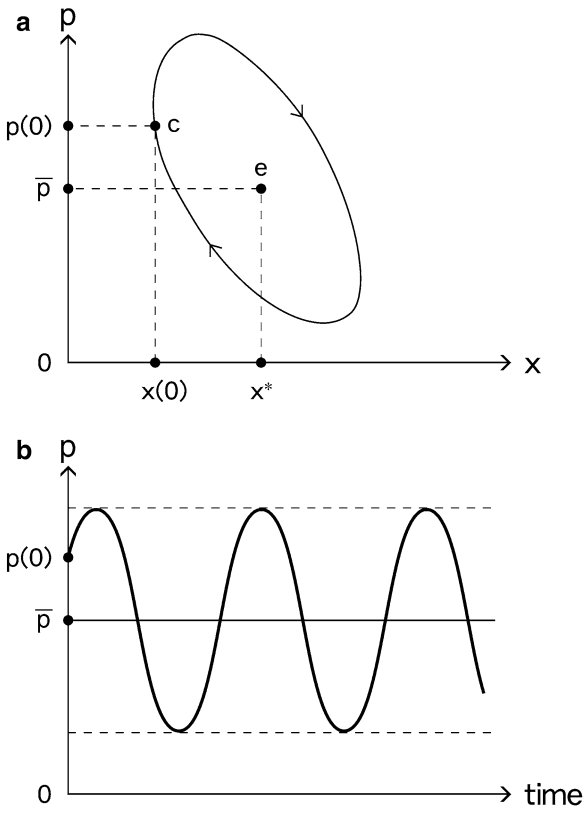


Fig. 5 Closed orbit (Points β_1 and β_2 in Fig. 2)

20530160) and Chuo University. Grant for Special Research Section 2 of this chapter is based on [1], although Sects. 3 and 4 and Appendix are not based on [1]. Needless to say, only the author is responsible for possible remaining errors. The author is grateful to Dr. Masahiro Ouchi of Nihon University, Tokyo, Japan for preparing LATEX version of this chapter.

Appendix

In this appendix, we reinterpret Eq. (35) in the text by means of a continuously distributed lag model of expectation formation following the procedure that was adopted by [20, 21]. Let us assume that the expected price is the weighted average of actual past prices, that is,

$$p^e(t) = \int_{-\infty}^{t} p(s)\omega(s)\mathrm{d}s, \tag{A1}$$

where $\omega(s)$ is a weighting function such that

$$\omega(s) \geqq 0, \quad \int_{-\infty}^{t} \omega(s)\mathrm{d}s = 1. \tag{A2}$$

In particular, we assume that our model is described by means of the following "simple exponential distributed lag" (cf. [20] Chap. 6 and [21]).[12]

$$\omega(s) = (1/\tau)\mathrm{e}^{-(1/\tau)(t-s)} \geqq 0 \quad ; \quad \tau > 0. \tag{A3}$$

Substituting (A3) into (A1), we obtain

$$p^e(t)\mathrm{e}^{(1/\tau)t} = (1/\tau)\int_{-\infty}^{t} p(s)\mathrm{e}^{(1/\tau)s}\mathrm{d}s. \tag{A4}$$

Differentiating (A4) with respect to t we obtain

$$\dot{p}^e(t) = (1/\tau)\{p(t) - p^e(t)\},$$

which is equivalent to Eq. (35) in the text if we write $\beta = 1/\tau$. We can interpret τ as the average time lag of expectation adaptation.

References

1. Asada, T. (2008) : "On the Existence of Cyclical Fluctuations in Continuous Time Dynamic Optimization Models : General Theory and its Application to Economics." Annals of the Institute of Economic Research, Chuo University 39, pp. 205–222. (in Japanese)
2. Asada, T., C. Chiarella, P. Flaschel and R. Franke (2003) : Open Economy Macrodynamics : An Integrated Disequilibrium Approach. Springer, Berlin.
3. Asada, T. and W. Semmler (1995) : "Growth and Finance : An Intertemporal Model." Journal of Macroeconomics 17–4, pp. 623–649.
4. Asada, T. and W. Semmler (2004) : "Limit Pricing and Entry Dynamics with Heterogeneous Firms." M. Gallegati, A. P. Kirman and M. Marsili eds. The Complex Dynamics of Economic Interaction : Essays in Economics and Econophysics, Springer, Berlin, pp. 35–48.
5. Asada, T, W. Semmler and A. Novak (1998) : "Endogenous Growth and Balanced Growth Equilibrium." Research in Economics 52–2, pp. 189–212.
6. Asada, T. and H. Yoshida (2003) : "Coefficient Criterion for Four-dimensional Hopf Bifurcation : A Complete Mathematical Characterization and Applications to Economic Dynamics." Chaos, Solitons and Fractals 18, pp. 525–536.
7. Benhabib, J. and K. Nishimura (1979) : "The Hopf Bifurcation and the Existence and Stability of Closed Orbits in Multisector Models of Optimal Economic Growth." Journal of Economic Theory 21. pp. 421–444.
8. Benhabib, J. and A. Rustichini (1990) : "Equilibrium Cycling with Small Discounting." Journal of Economic Theory 52, pp. 423–432.
9. Chiang, A. (1992) : Elements of Dynamic Optimization. McGraw-Hill, New York.

[12] We have $\int_{-\infty}^{t}(1/\tau)\mathrm{e}^{-(1/\tau)(t-s)}\mathrm{d}s = (1/\tau)\mathrm{e}^{-(1/\tau)t}\int_{-\infty}^{t}\mathrm{e}^{(1/\tau)s}\mathrm{d}s = \mathrm{e}^{-(1/\tau)t}[\mathrm{e}^{(1/\tau)s}]_{s=-\infty}^{s=t} = 1.$

10. Dixit, A. K. (1990) : Optimization in Economic Theory(Second Edition). Oxford University Press, Oxford.
11. Dockner, E. and G. Feichtinger (1991) : "On the Optimality of Limit Cycles in Dynamic Economic Systems." Journal of Economics 53–1, pp. 31–50.
12. Dockner, E., S.Jorgensen, N. Van Long and G. Sorger (2000) : Differential Games in Economics and Management Science. Cambridge University Press, Cambridge.
13. Faria, J. R. and J. P. Andrade (1998) : "Investment, Credit, and Endogenous Cycles." Journal of Economics 67–2, pp. 135–143.
14. Feichtinger, G., A. Novak and F. Wirl (1994) : "Limit Cycles in Intertemporal Adjustment Models." Journal of Economic Dynamics and Control 18, pp. 353–380.
15. Gandolfo, G. (2009) : Economic Dynamics (Fourth Edition). Springer, Berlin.
16. Gaskins, D. W. (1971) : "Dynamic Limit Pricing : Optimal Pricing Under Threat of Entry." Journal of Economic Theory 3, pp. 306–322.
17. Judd, K. and B. Petersen (1986) : "Dynamic Limit Pricing and Internal Finance." Journal of Economic Theory 39, pp. 368–399.
18. Liu, W. M. (1994) : "Criterion of Hopf Bifurcation without Using Eigenvalues." Journal of Mathematical Analysis and Applications 182, pp. 250–256.
19. Romer, P. (1990) : "Endogenous Technological Change." Journal of Political Economy 98, pp. 71–102.
20. Shinkai, Y. (1970) : Economic Analysis and Differential-Difference Equations. Toyo Keizai Shinpo-sha, Tokyo. (in Japanese)
21. Yoshida, H. and T. Asada (2007) : "Dynamic Analysis of Policy Lag in a Keynes-Goodwin Model : Stability, Instability, Cycles and Chaos." Journal of Economic Behavior and Organization 62, pp. 441–469.

Controlling of Processes by Optimized Expertsystems

Wolfram-M. Lippe

Abstract Expertsystems are characterised by storing knowledge, normally in if-then-rules. The human Intelligence implies the ability to comprehend, reason, memorise, learn, adapt and create. The attribute of certainty or precision does not exist in human perception and cognition. Perception and cognition through biological sensors, pain reception and other similar biological events are characterised by many uncertainties. A person can linguistically express perceptions experienced through the senses, but these perceptions cannot be described using conventional statistic theory. The perception and cognition activity of the brain is based on relative grades of information acquired by the human sensory systems. These are the reasons that fuzzy logic has been applied very successfully in many areas where conventional model–based approaches are difficult or not cost-effective to implement. Therefore fuzzy-rule based Expertsystems have many advantages over classical expertsystems. Hybrid neuro-fuzzy-Expertsystems combine the advantages of fuzzy systems, which deal with explicit knowledge which can be explained and understood, and neural networks which deal with implicit knowledge which can be acquired by learning. Different methods are known for combining fuzzy-rule-based-systems with neural networks. But all these methods have some disadvantages and restrictions. We suggest a new model enabling the user to represent a given fuzzy-rule-base by a neural network and to adapt its components as desired.

Key words Expertsystems • Fuzzy • Artificial neural networks • If-then-rules • Optimisation

W.-M. Lippe (✉)
Institute for Computer Science, University of Muenster, Einsteinstr. 62,
D-48149 Muenster, Germany
e-mail: lippe@uni-muenster.de

A. Chinchuluun et al. (eds.), *Optimization, Simulation, and Control*, Springer Optimization and Its Applications 76, DOI 10.1007/978-1-4614-5131-0_14,

1 Introduction

Hybrid neuro-fuzzy-Expertsystems combine the advantages of fuzzy systems, which deal with explicit knowledge which can be explained and understood, and neural networks which deal with implicit knowledge which can be acquired by learning. Neural network learning provides a good way to adjust the expert's knowledge and automatically generate additional fuzzy-rules and membership functions, to meet certain specifications and reduce design time and costs.

Fuzzy reasoning can be applied in many areas, e.g. for fuzzy control, fuzzy diagnosis, and fuzzy modelling. Essentially fuzzy Expertsystems consist of four components: the fuzzification unit, the rule-base to store the knowledge, the inference-engine and (if needed) the defuzzyfication unit. The fuzzy-rules of the rule-base are of the following general type:

$$R: \underbrace{\text{IF } x_1 = A_1 \text{ AND} \ldots x_n = A_n}_{\text{premise}} \text{ THEN } \underbrace{y_j = B_j}_{\text{conclusion}}$$

Fuzzy-rule-based systems differ in the form of the conclusion (fuzzyfied *(Mamdani-like)* or crisp *(Sugeno-like)*) and in different calculation methods. These can be divided into methods used for fuzzification, defuzzification and evaluation of the rules.

The final output is calculated by the following steps:

1. Fuzzyfication of the incoming data
2. Calculating the degree of acceptance of the premise
3. Calculating the results of the rules *(evaluating the conclusions)*
4. Calculating the output values rsp. output-fuzzy-sets
5. Defuzzification of the output-fuzzy-sets *(if necessary)*

There are two methods to create the fuzzy-rules:

In *data-driven methods* the rules are constructed through analysing given examples by the help of mathematical methods. In *expert-driven methods* the rules are specified by exploring the knowledge of a human expert.

The advantage of fuzzy-rule-based systems lies in the simple understanding of the stored knowledge and the similarity to the human-like reasoning. One disadvantage is that these systems are not adaptive. Therefore before using these systems, all rules, fuzzy sets and methods have to be specified completely. The quality of a system depends on the quality of this specification. Furthermore, due to the dynamic nature of economic and financial applications, rules and membership functions must be adaptive to the changing environment in order to continue to be useful.

Artificial neural networks (ANN) are adaptive systems. The advantage of ANNs lies in the processing of training examples for the construction, no explicit algorithm is necessary. So they can be synthesised without making use of the detailed, explicit knowledge of the underlying process.

A disadvantage of ANNs is the lack of verification; they are “black boxes”. The I/O behaviour is known for the training and test data but not for other data. Furthermore an adequate training set is required. Normally it is not possible to use well-known information for the construction of an ANN. The late 1990s witnessed the development of hybrid systems, which combine the advantages of two or more SoftComputing techniques. Evolutionary algorithms were used to optimize fuzzy-rule-based systems as well as combinations of ANNs and Fuzzy systems. These neuro-fuzzy-systems have many advantages over other combinations because they combine the power of adaptivity and learning with clearness and simple understanding. So systems with neuro-fuzzy components may be found in many fields such as stock market prediction, intelligent information systems, and data mining and s.o.

Expertsystems are characterised by storing knowledge, normally in if-then-rules. The human intelligence implies the ability to comprehend, reason, memorise, learn, adapt and create. The attribute of certainty or precision does not exist in human perception and cognition. Perception and cognition through biological sensors, pain reception and other similar biological events are characterised by many uncertainties. A person can linguistically express perceptions experienced through the senses, but these perceptions cannot be described using conventional statistic theory. The perception and cognition activity of the brain is based on relative grades of information acquired by the human sensory systems. These are the reasons that fuzzy logic has been applied very successfully in many areas where conventional model-based approaches are difficult or not cost-effective to implement. Therefore fuzzy-rule-based Expertsystems have many advantages over classical Expertsystems.

In this chapter we discuss the existing tools and present a new tool for modelling an existing fuzzy-rule-based system by an ANN which can be improved by special learning rules in a training phase and which is able to handle all the possibilities to optimise the given rule-base. The tool is based on the simulation of a given and nonoptimal fuzzy-rule-based-system by an equivalent neural network to optimise this neural network in a training phase (can be done on the job) and recreating the (optimised) rules from the improved neural network, if wanted. The possible improvements of a given rule-base can be a modification or deletion of existing fuzzy sets, creation of new fuzzy sets, modification or deletion of existing fuzzy-rules or the creation of new rules. Some examples for well-known models working in this way are e.g. NEFCON, Lin/Lee, NARA and ANFIS. Unfortunately none of the existing models can handle all the possibilities listed above. This new tool exists in two versions: one for Mamdani-like systems and one for Sugeno-like systems.

2 Existing Neuro-Fuzzy-Expertsystems

In this chapter we will discuss the construction and properties of the existing systems. NARA (introduced by H. Takagi 1991/92) and ANFIS (adaptive-network-

based-fuzzy-inference-system, introduced by J.S.R. Jang 1992/93) are the oldest ones, and their restrictions and disadvantages are well known. So we will focus on Lin/Lee-systems and NEFCON systems.

2.1 Lin/Lee System

At the beginning of the 1990s C.T. Lin and C.S. Lee developed a procedure by which Mamdani controllers can be optimised. For this purpose at first a given controller is transformed into a functional equivalent five-layered ANN. Therefore, fuzzy sets are used for weights in the generated network instead of usual real numbers. In the subsequent training period, the weights (and therefore the fuzzy sets) are adapted by means of a special learning rule based on the backpropagation procedure. Furthermore, it is possible to generate new output partition sets and to change the conclusions of rules.

The generated five-layered network is constructed as follows:

The input layer contains one neuron for every input value. In layer 2 there is one neuron for every available linguistic term of the input partitions. This is connected in each case with the input neuron which is assigned to the same input dimension. As a weight the input partitions set, which represents the respective linguistic term, is used. The used fuzzy sets are Gaussian or triangular sets. Layer 3 contains one neuron for every rule. This is connected exactly with the neurons from layer 2, which stand for the linguistic terms of the premise of this rule. Here, no weights are used. In layer 4 there is, analogously to layer 2, exactly one neuron for every available linguistic term of the output partitions. This is connected in each case with all neurons from layer 3. Its associated rule has this term as a conclusion. In this process no weights are used. In the output layer there is one neuron for every output dimension. This is connected with all neurons from layer 4. Its linguistic term belongs to this output dimension. In each case the output partition set, that is assigned to the linguistic term, is used as a weight.

The calculation of the network output corresponds to the calculation of the output of the given fuzzy controller. In layer 1 every neuron transmits its input unmodified. Every neuron from layer 2 calculates the membership grade of its input value to the fuzzy set, which is used as a weight of the connection with layer 1. In layer 3, every neuron calculates the minimum of its input values. This gives the "fulfillment grade" of the premise of the rule that it represents. In layer 4 every neuron calculates the minimum between 1 and the sum of its input values.

Thereby an inclusive OR operation is realised to determine the cut height of the corresponding fuzzy set. In layer 5 every neuron j calculates its output as follows:

$$O_j = \frac{\sum_{i \in S_4} m_{5,i,j} w_{5,i,j} z_{4,i}}{\sum_{i \in S_4} w_{5,i,j} z_{4,i}}$$

whereby the sum runs in each case only through the neurons i from layer 4 with which the current output neuron is connected, $z_{4,i}$ is the respective output and $m_{5,i,j}$ the modal value and $w_{5,i,j}$ is the range of the fuzzy set of the connections with layer 4. In this way a centroid defuzzification is approximated.

In the procedure of Lin and Lee a hybrid learning rule is used which is based on backpropagation procedure, i.e. the weights of the network (and with it the fuzzy sets) are trained with a backpropagation procedure. In addition, if required new neurons are generated in layer 4, where at the same time suitable connections with layer 3 and layer 5 are generated. This structural change of the network corresponds to the production of new output partitions sets, as well as to the formation of new conclusions for some rules.

A disadvantage of this procedure is that no fuzzy sets can be deleted. As the new generated fuzzy sets are compared only to the fuzzy sets already available in the network, but not among each other, the number of the sets in one single step often increases dramatically. During this process fuzzy sets are perhaps generated, which are not actually necessary for a correct fuzzy feedback control.

Another disadvantage is that there is no possibility to generate also new input-fuzzy-sets either or to provide new rules. If at the initialisation of the network less input-fuzzy-sets or rules than absolutely necessary are used, then the system is not able to provide an optimally functioning fuzzy controller.

2.2 NEFCON System

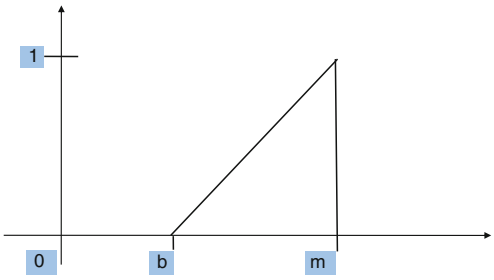
The NEFCON model was developed in the middle of the 1990s at the University of Technology of Brunswick by D. Nauck, F. Klawonn and R. Kruse. As with the Lin and Lee method it is able to optimise a given Mamdani controller by the fact that this controller is transformed at first in a functionally equivalent ANN which is then trained and optimised afterwards with special learning procedures.

A NEFCON system is a three-layered neural network which can represent a fuzzy controller. The essential difference to a traditional neural network lies in the fact that fuzzy sets are used as weights for the connections instead of real numbers (similar to Lin and Lee method).

The input layer exists of n neurons which only pass its input value. In layer 2, the *rule layer*, there is one neuron for every rule. The connections of layer 1 are named in each case with a linguistic term. Thereby for the connections of the neurons $E_1, \ldots, E_n$ of layer 1 to neuron R_k of layer 2 those linguistic terms used in the premise of rule R_k are used.

For the input and output exclusively special fuzzy sets are used; triangular sets are only provided as inputs and prong sets as outputs (“bisected triangle”). A prong set is shown in Fig. 1.

The calculation of the network output corresponds to the calculation of the output of a fuzzy controller. In layer 1 every neuron outputs its (real) input again. In layer 2 every neuron R_k calculates the grade of fulfillment of the premise of rule R_k.



$$\mu_{\overline{B}}(x) = \begin{cases} \frac{x-b}{m-b} & : x \in [b.m] \; if \; b < m \\ \frac{b-x}{b-m} & : x \in [m,b] \; if \; m < b \\ 0 & : else \end{cases}$$

Fig. 1 Prong sets

The modification of fuzzy sets is realised in six steps. If there is no suitable control base then a corresponding rule-base can automatically be created within the NEFCON model with the help of another learning method. A condition for the application of this method is that at least more or less suitable fuzzy sets are defined for the input and output spaces. Here, the number of the defined fuzzy sets in particular must be correct. A more precise adaptation of the fuzzy sets (i.e. fine adjustment) occurs subsequent to the rule-base learning with the method described above. In addition, as with the learning method for the adaptation of the fuzzy sets, the right algebraic sign of the control variable must be known.

The idea behind the used procedure is, first of all, to provide all (!) the rules which are able to be generated by means of the fuzzy sets previously defined by the user. For the premise every possible combination of fuzzy sets on the input dimensions is used; as a conclusion, in addition, every given fuzzy set is used on the output space. False and superfluous rules are removed iteratively, until an appropriated rule-base remains. This process is executed in two phases. In the first phase in each case all rules whose contribution to the result has the false algebraic sign are deleted. In the second phase, of the rules remaining, those with the same premise are pooled in each case to a set of rules. Then, for every run from each of these sets, a rule is chosen, that is used for the calculation of the result. Subsequently the error quotient of every used rule is stored and added up. Afterwards, the rule with the slightest error quotient is selected from all sets. The other rules are deleted, also those rules that are only rarely *active*.

A disadvantage of the NEFCON system is the condition that the membership functions of fuzzy sets of the output space have to be monotonous on its porter, whereby the range of the used fuzzy sets is constrained. This is unfavourable, because especially non-monotonic triangular sets and Gaussian sets are often used for fuzzy controllers. Fuzzy controllers, that use sets of these types are basically not appropriate for optimisation with the NEFCON system. Being able to determine just one output value is another disadvantage. Therefore, it is not easy to assign and optimise an arbitrary created fuzzy controller to the NEFCON system.

Furthermore there is no possibility to check available rules and to correct if necessary, without creating missing fuzzy sets. Therefore, the application is only reasonable for those fuzzy controllers that fulfil the mentioned conditions (monotony, one output value), whereas at least the number of required fuzzy sets must be known. In case of a fuzzy set that is necessary for the correct control, being forgotten, the NEFCON system cannot generate it and therefore by the presented procedures cannot provide fuzzy controllers that function in every situation.

3 An Example

The algorithm for modelling a given fuzzy-rule-based Expertsystem to control a process by our approach, which is able to optimise an Expertsystem without any restrictions and without the disadvantages of the other systems, is demonstrated by a well-known example, the overturned pendulum, shown in Fig. 2.

The input space consists of the position $\Theta(X_1)$ and the speed $\lambda(X_2)$, the output is the power $F(Y_1)$. The partition of the input space and the output space is given by

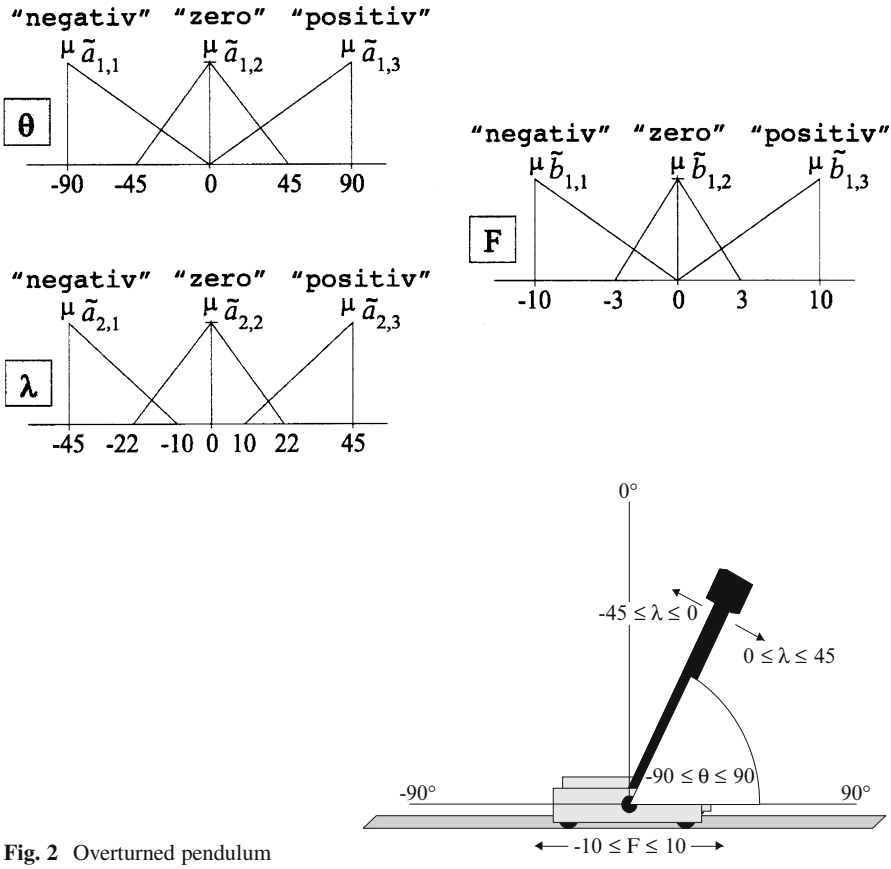


Fig. 2 Overturned pendulum

The fuzzy-rule-base is given by

IF	(x_1 IS neg.)	AND	(x_2 IS zero)	THEN	(y_1 IS neg.)
IF	(x_1 IS neg.)	AND	(x_2 IS pos.)	THEN	(y_1 IS zero)
IF	(x_1 IS zero)	AND	(x_2 IS neg.)	THEN	(y_1 IS neg.)
IF	(x_1 IS zero)	AND	(x_2 IS zero)	THEN	(y_1 IS zero)
IF	(x_1 IS zero)	AND	(x_2 IS pos.)	THEN	(y_1 IS pos.)
IF	(x_1 IS pos.)	AND	(x_2 IS neg.)	THEN	(y_1 IS zero)
IF	(x_1 IS pos.)	AND	(x_2 IS zero)	THEN	(y_1 IS pos.)

4 The Optimization Process

In a first step the fuzzy-rules of a given Expertsystem is transformed into a functionally equivalent neural network. The network is a simple feedforward network and consists of four layers. The first hidden layer calculates the premises. The weights between the input layer and the first hidden layer are the fuzzy sets of the premises. The second hidden layer calculates the conclusions, and the last step calculates the output-fuzzy-set. The correctness of the transformation can be shown by a formal proof. After this step a training phase can be started to improve the given net resp. the given rules.

The corresponding equivalent neural network for the rule-base of the example is given in Fig. 3.

5 The Training Rules

There are two different techniques for an improvement of the rule-base rsp. the corresponding neural network: A fine tuning of the fuzzy sets can be done by classical backpropagation techniques. A “rough” tuning can be done by the following learning rules:

5.1 Training the Rules

5.1.1 Surplus Rules

For all training inputs we inspect all rules with the same conclusion to find surplus rules: If a rule has—for each training example—the minimum degree of acceptance, it has no influence on the result, so it can be removed. If a rule has—for each training example—the maximum degree of acceptance, it determines the intersection-altitude alone, so all other inspected rules can be removed.

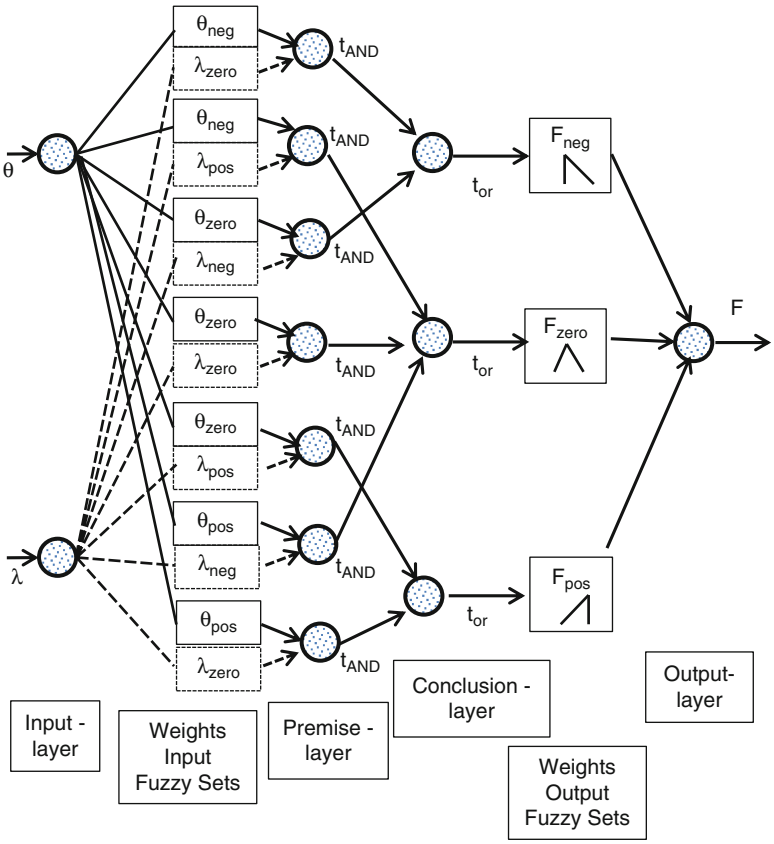


Fig. 3 Neural network for the rule–base

5.1.2 Faulty Rules

Case 1. If a rule, which includes every input space in its premise, has a large degree of acceptance, the system is exactly in the situation, which the rule describes. If in that case the result calculated solely with this rule has a large error, this rule is incorrect and gets a new conclusion. The new conclusion is the output-fuzzy-set of the corresponding output dimension which leads to the minimum distance between the result calculated solely with this rule and correct output.

Case 2. The case 1 method can not be used for rules, which do not use every input space in their premise, because the correct output depends on the entire condition of the system. In this case we proceed as follows:

- Examine—for each rule—all training examples which cause a degree of acceptance above a threshold S_1.

- Calculate for each of these training-examples solely with this rule the result and its error.
- If a rule has—for each training example which causes a degree of acceptance above S_1—an error above a threshold S_2, the conclusion-fuzzy-set of this rule lies in a wrong area.
- Incorrect rules get corrected by supplying them with a new conclusion (either an existing fuzzy set, or a new one is created).

5.1.3 New Rules

If the degree of acceptance of each rule for given input values $(s_1, \ldots, s_n)$ is below a threshold, there is no rule for this situation. Therefore, one has to be created. The premise is defined by choosing for each input value s_i the fuzzy set of the input dimension X_i, which causes the maximum membership degree of s_i. The conclusion is found in the same way with the correct output values.

5.2 Training the Fuzzy Sets

(Cannot be done separately but during part I methods)

5.2.1 Correcting Rules: Creating Sets

Case 1.
- When correcting a rule, which uses all inputs in its premise, the output-fuzzy-set, which leads to the minimum distance between the result calculated solely with this rule and the correct result, is chosen as conclusion.
- If the minimum distance to the correct output is above a threshold, there is no correct output-fuzzy-set for the current situation. Therefore one has to be created. Mean is the correct output, width is e.g. the distance to the mean of the next fuzzy set.

Case 2.
- When correcting a rule, which does not use all inputs in its premise, the output-fuzzy-set is chosen, which yields the minimum error most of the times.
- If the minimum error is not once below a threshold S_2, there is no correct fuzzy set for this rule. Therefore one has to be created. Mean is the intersection from the correct output of all training examples, which cause a degree of acceptance above S_2, width is e.g. the distance to the mean of the next fuzzy set.

5.2.2 Creating Rules: Creating Sets

When creating new rules the input-fuzzy-set is chosen, which leads to the maximum membership degree. If one input s_i causes—for each fuzzy-set of X_i— a membership

degree below a threshold, there is no correct fuzzy set. Therefore one has to be created. Mean is s_i width is e.g. the distance to the mean of the next fuzzy set. If necessary, for the conclusion, a new fuzzy set is created analogously.

5.2.3 Removing Sets

- Fuzzy sets, on which no rules apply, can be removed.
- If two neighboring fuzzy sets lead to the same conclusion every time, they can be combined.

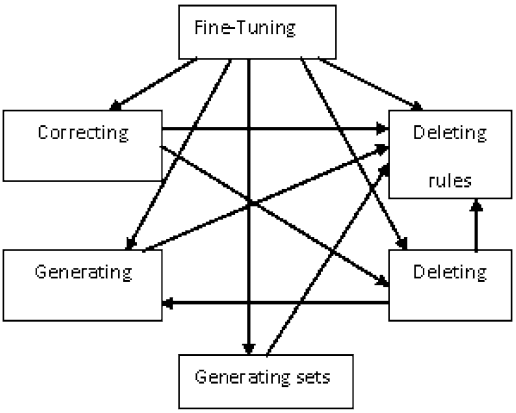
5.2.4 Tuning Sets

- The error of the neurons in layer 2 can be calculated directly: In layer 2 each neuron calculates the degree of acceptance E of the premise from the rule, which it presents. By that it holds that the more the conclusion is correct, the larger the degree of acceptance has to be.
- If the correct output is exactly the mean of the conclusion-fuzzy-set, the degree of acceptance has to be 1; if the correct output is outside of the conclusion-fuzzy-set, the degree of acceptance has to be 0. Therefore it is advisable, to define the correct degree of acceptance by the membership degree g of the correct output in the conclusion-fuzzy-set. Then the error is calculated by $F = gE$.
- If the degree of acceptance is too large, each input value has too large a membership degree in its corresponding fuzzy set, because the membership-degrees are combined by means of fuzzy-t-norm. Therefore all applied input-fuzzy-sets get shifted away from the input values and are curtailed.
- If the degree of acceptance is too low, only the applied fuzzy sets which cause too low membership degrees get shifted in the direction of the input values.

5.2.5 Output

- The output-fuzzy-sets are adapted as follows: the squared error of an output value is defined by $F = (s-t)^2$, s is the output and t the correct output.
- Because the influence of an output-fuzzy-set on the results depends on its intersection-altitude M, the intersection-altitude is considered when adapting the output-fuzzy-sets. The correct output yields the "correct" output-fuzzy-set to be the one, whose mean has the minimum distance to the correct output. The "correct" output-fuzzy-set gets shifted in the direction of the correct output value. The "wrong" output-fuzzy-sets get curtailed.

Because of some interconnections between the different learning steps an application should be done in the following order:



6 Conclusion

In this Chapter we focussed on modelling and optimising fuzzy-rule-based Expertsystems. The if-then-rules of the rule-base are transformed into a functionally equivalent neural network. After that connectionist methods for modifications and optimisations are used. In contrast to existing systems, the user may specify which components are to be adapted. The user may choose to adjust all components of the rule-base separately or simultaneously. There are no restrictions concerning the types of fuzzy sets or the defuzzification method. The systems can handle all the possibilities to optimise a fuzzy-rule-base. The system was tested in many concrete applications. The results were extremely positive.

References

1. Davoian, K., Lippe, W.-M.: Exploring the Role of Activation Function Type in Evolutionary Artificial Neural Networks. In: Proc. Int. Conf. on Data Mining '08 (DMIN08), 2008
2. Jang, J.S.R.: ANFIS: Adaptive Network based Fuzzy Inference System. IEEE Transactions on Systems, Man and Cybernetics 23, (3), 665685, 1993
3. Kang H.-J. et al.: A New Approach to Adaptive Fuzzy Control. Proc. FUZZ-IEEE98, pp. 268–273, 1998.
4. Kang Y. et al.: Optimization of Fuzzy-Rules: Integrated approach for Classification Problems. LNCS 3984, pp. 665–674. Springer (2006)
5. Kolodziej, C., Priemer, R.: Design of a Fuzzy Controller Based on Fuzzy Closed-Loop Specifications. Proc. ANNIE96, pp. 249-255, 1996
6. Li W. : Optimization of a Fuzzy Controller Using Neural Networks. Proc. FUZZ-IEEE94, pp. 223–228, 1994
7. Lin, C.T., Lee, C.S.G.: Neural Fuzzy Control Systems with Structure and Parameter Learning. World Scientific, 1994
8. Lippe, W.-M.: Soft-Computing Neuronale Netze, Fuzzy Logic und Evolutionre Algorithmen. Springer (2006)

9. Moraga, C., Salas R.: A new aspect for the optimization of Fuzzy if-then-rules. Proc. 35th Int. Symposium on Multiple Valued Logic, pp. 160–165, 2005
10. Nauck, D., Kruse, R.: NEFCON-1: An XWindow based Simulator for Neural Fuzzy-Controllers. in R. Kruse, J. Gebhardt and R. Palm: Fuzzy-Systems in Computer Science. Vieweg, Braunschweig, pp. 141–151
11. Nauck, D., Kruse, R.: NEFCON-1: An XWindow based Simulator for Neural Fuzzy-Controllers. Proc. IEEE Int. Conf. Neural Networks 1994 at IEEE WCCI '94, pp. 1638–1643,1994.
12. Perng, C.-F. et al.: Self-Learning Fuzzy Controller with a Fuzzy Superviser. Proc. FUZZ-YEEE98, pp. 331–357, 1998
13. Shi, Y., Mizumoto, M., Yubazaki, N.: An Improvement of Fuzzy Rules Generation Based on Fuzzy c-means Clustering Algorithm. Japanese Journal of Fuzzy Theory and Systems, vol. 9–4., pp. 395–407, 1997.
14. Silipo, R.: Extracting Information from Fuzzy Models. Proc. 9th Int. Conf. of the North American Fuzzy Information Processing Society, pp. 44.48, 2000
15. Takagi, H., Hagashi, I.: NN-driven fuzzy reasoning. Int. Journal of Approximate Reasoning 5; (3), 191–212, 1991
16. Takagi, H., Susuki, N., Koda, T., Kojima, Y.: Neural Networks designed on approximate reasoning architecture and their applications. IEEE Trans. Neural Networks 3, (5), 752–760, 1992
17. Yeung, D. et al. Fuzzy Production Rule Refinement using Multilayer Perceptrons, Proc. FUZZ-IEEE94, pp. 211 218, 1994

Using Homotopy Method to Solve Bang–Bang Optimal Control Problems

Zhijie Gao and Hexi Baoyin

Abstract According to the Pontryagin maximum principle, some optimal control problem can result in a bang-bang control law. In despite of what method is used in the optimization procedure for the bang-bang control, fixing switching points of the bang-bang control is very intractable. In this chapter, the smoothing technique presented by Bertrand et al. for solving bang-bang optimal control problems is introduced, but its convergence is quite slow. To overcome this flaw, based upon a method termed homotopy method, this chapter presents an integration switching method which can converge very fast. Finally, two numerical examples are solved illustrating the interest of our method, and the simulation results are provided to demonstrate the effectiveness of our method.

Key words Homotopy method • Maximum principle • Bang–bang control

1 Introduction

There are three types of method to optimize continuous thrust spacecraft trajectories: direct method, indirect method, and hybrid method. The direct method discretizes an optimal control problem into a parameter optimization problem and then uses nonlinear programming method[1, 2] to solve the original problem. The indirect method uses Pontryagin maximum principle[3] to convert an optimal control problem into a two-point boundary value problem (TPBVP)[4, 5]. For most

Z. Gao
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
e-mail: gzj08@mails.tsinghua.edu.cn

H. Baoyin (✉)
Department of Aerospace Engineering, Tsinghua University, Beijing 100084 China
e-mail: baoyin@tsinghua.edu.cn

A. Chinchuluun et al. (eds.), *Optimization, Simulation, and Control*, Springer Optimization and Its Applications 76, DOI 10.1007/978-1-4614-5131-0_15,

of engineering problems, however, the TPBVP is too sensitive to initial value of the costate to guess properly. Meanwhile the hybrid method combines the previous two methods, uses maximum principle to depict out some properties of the optimal control law, and then discretizes the problem to solve a nonlinear programming instead of solving a TPBVP[6]. According to maximum principle, some optimal control problem result in a bang–bang control law. In despite of what method is used in the optimization procedure for the bang–bang control, fixing switching points of the bang–bang control is very intractable. To handle this difficulty, Bertrand et al.[7] present a smoothing technique to fix the switching points by shooting methods but a drawback there is its convergence is quite slow, usually need thousands of steps. To overcome this flaw, this chapter presents an integration switching method which can converge very fast, usually need only few steps. Finally, two numerical examples are solved illustrating the interest of our method.

2 Two-Point Boundary Value Problem

Consider a continuous thrust spacecraft moves in a centric gravitation field. Its dynamic equations can be written as

$$\begin{cases} \dot{r} = v \\ \dot{v} = -\frac{\mu}{r^3} r + \frac{T}{m}\alpha \\ \dot{m} = -\frac{T}{I_{sp} g_0}, \end{cases} \tag{1}$$

where T is the thrust magnitude with constraint $0 \le T \le T_{\max}$ and α is the unit vector of the thrust. At fixed initial time t_0, it has fixed initial position vector r_0, velocity vector v_0, and mass m_0, and at fixed final time t_f, it must arrive at a fixed final position vector r_f and velocity vector v_f.The objective function to be minimized is

$$\int_{t_0}^{t_f} T \mathrm{d}t. \tag{2}$$

It is equivalent to fuel consumption. According to the calculus of variations, the corresponding Hamiltonian can be formed as

$$H = T + \lambda_r^T v + \lambda_v^T \left(-\frac{\mu}{r^3} r + \frac{T}{m}\alpha \right) - \lambda_m \frac{T}{I_{sp} g_0}, \tag{3}$$

where λ_r, λ_v, and λ_m are costates associated with the states r, v, and m, respectively. The Pontryagin maximum principle tells that the optimal control should satisfy

$$\alpha^* = -\frac{\lambda_v}{\|\lambda_v\|}, T^* = T_{\max} u, \tag{4}$$

where u is decided by switch function ρ as

$$\begin{cases} u = 0, & \text{if} \rho > 0 \\ u = 1, & \text{if} \rho < 0 \\ 0 < u < 1, & \text{if} \rho = 0 \end{cases} \tag{5}$$

$$\rho = 1 - \frac{\|\lambda_v\|}{m} - \frac{\lambda_m}{I_{sp} g_0}. \tag{6}$$

That is to say, the optimal thrust magnitude is either zero or the maximum, which is termed bang–bang control. Then the dynamic equations of both states and costates can be formed as

$$\begin{cases} \dot{r} = v \\ \dot{v} = -\frac{\mu}{r^3} r - \frac{\lambda_v}{\|\lambda_v\|} \frac{T_{\max}}{m} u \\ \dot{m} = -\frac{T_{\max} u}{I_{sp} g_0} \\ \dot{\lambda}_r = \frac{\mu}{r^3} \lambda_v - \frac{3\mu r \cdot \lambda_v}{r^5} r \\ \dot{\lambda}_v = -\lambda_r \\ \dot{\lambda}_m = -\|\lambda_v\| \frac{T_{\max}}{m^2} u. \end{cases} \tag{7}$$

The boundary conditions are

$$r(t_0) = r_0, v(t_0) = v_0, m(t_0) = m_0, \tag{8}$$

$$r(t_f) = r_f, v(t_f) = v_f, \lambda_m(t_f) = 0. \tag{9}$$

Denote the column vector $[\lambda_r; \lambda_v; \lambda_m]$ by λ, which has seven components. Then the goal is to find a certain λ_0 that when combining with conditions (8) and governed by Eq. (7), the moment it propagates from t_0 to t_f, conditions (9) are satisfied. It can be summarized as a column vector of shooting function

$$S(\lambda_0) = \begin{bmatrix} r(t_f, \lambda_0) - r_f \\ v(t_f, \lambda_0) - v_f \\ \lambda_m(t_f, \lambda_0) \end{bmatrix} = 0 \tag{10}$$

which has seven components to determine a column vector with seven unknowns.

In principle, if the initial guess values are given properly, and Eq. (7) are integrated accurately enough, solving Eq. (10) through some classic nonlinear equation solvers such as Newton–Raphson method is not difficult. However, on the one hand, the initial guess values are usually not easy to be given in convergence field. On the other hand, integrating Eq. (7) with high-enough accuracy is very

difficult because when acted by bang–bang control, the right sides of Eq. (7) are discontinuous but burst at the moment the switch function takes the value zero.

3 Homotopy Method

In this section, the smoothing technique presented by Bertrand et al. for solving bang–bang optimal control problems will be introduced. And the performance index is modified based on a method termed homotopy method. The optimal thrust then becomes continuous, so that it is not too difficult to integrate the dynamic equations. Especially, the initial guess values can be updated from a series of cases easier to solve.

Instead of the purely fuel consumption expressed by Eq. (2), the performance index is modified to be

$$T_{\max}\int_{t_0}^{t_f}\left[\frac{T}{T_{\max}}-\varepsilon\frac{T}{T_{\max}}\left(1-\frac{T}{T_{\max}}\right)\right]\mathrm{d}t, \tag{11}$$

where $0 \leq \varepsilon \leq 1.0$. For $\varepsilon = 0$, it is equal to Eq. (2); while for $\varepsilon = 1.0$, it becomes quadratic, which corresponds to continuous optimal thrust that has wider convergence domain. The Hamiltonian becomes

$$H = T - \varepsilon T\left(1-\frac{T}{T_{\max}}\right)+\lambda_v^T v+\lambda_r^T\left(-\frac{\mu}{r^3}r+\frac{T}{m}\alpha\right)-\lambda_m\frac{T}{I_{sp}g_0}. \tag{12}$$

Then, Eq. (5) should be formed as

$$\begin{cases} u = 0, \text{if}\rho > \varepsilon \\ u = 1, \text{if}\rho < -\varepsilon \\ u = \frac{1}{2}-\frac{\rho}{2\varepsilon}, \text{if}\,|\rho| \leq \varepsilon. \end{cases} \tag{13}$$

The normalized thrust magnitude u expressed by Eq. (5) is discontinuous with respect to switch function, while it becomes continuous right now. Since switch functions are always continuous with respect to time, u is also continuous with respect to time.

Note that it is still indifferent at the moment when ρ takes the value zero and $\pm\varepsilon$, which still brings trouble to integrate dynamic equations through classic integrators such as fourth-order Runge–Kutta method. Nevertheless, it is easier to solve the nonlinear equations (10) for the cases with ε near to 1, meaning that $\rho > \varepsilon$ and $\rho < -\varepsilon$ both never happen. Importantly, the solution to ε_k can be used as initial guess value to solve the case with ε_{k+1}, where $1.0 \geq \varepsilon_k > \varepsilon_{k+1} \geq 0.0$. In practice, we find that when ε nears to zero that corresponds to the initial problem of minimal

fuel consumption, directly using integrators with adaptive step such as ode45 in Matlab is hard to converge. The thrust u is still continuous, but it varies rapidly between the maximum and zero at the moment ε nears to zero. Thus, Bertrand et al. [7] did not use quadratic penalty but use logarithmic barrier to solve their examples. The performance index is modified to be

$$T_{\max}\int_{t_0}^{t_f}\left\{\frac{T}{T_{\max}}-\varepsilon\ln\left[\frac{T}{T_{\max}}\left(1-\frac{T}{T_{\max}}\right)\right]\right\}\mathrm{d}t, \tag{14}$$

where the normalized thrust magnitude u is determined by

$$u=\frac{2\varepsilon}{\rho+2\varepsilon+\sqrt{\rho^2+4\varepsilon^2}} \tag{15}$$

which is not only continuous but also differentiable with respect to switch function ρ and time. But at the commutation points, its derivative is still large so as to influence computation efficiency. Thus, they decrease ε with very small step to guarantee convergence.

4 Fourth-Order Runge–Kutta Integrator with Switch Function Detection

To avoid the deficiency of logarithmic barrier in computation efficiency, we keep the quadratic penalty but add a special integration process when the value of switch function passes through the range $[-\varepsilon,\varepsilon]$. To solve the initial value problem of ordinary differential equation

$$\dot{x}=f(t,x),t_0\le t\le t_f,x(t_0)=x_0 \tag{16}$$

the fourth-order Runge–Kutta algorithm is well known as

$$\begin{aligned}
x_{k+1}&=x_k+\frac{h}{6}(k_1+2k_2+2k_3+k_4),\\
k_1&=f(t_k,x_k),\\
k_2&=f\left(t_k+h/2,x_k+h/2k_1\right),\\
k_3&=f\left(t_k+h/2,x_k+h/2k_2\right),\\
k_4&=f(t_k+h,x_k+hk_3).
\end{aligned} \tag{17}$$

Here denote them by a formula

$$x_{k+1} = \mathrm{RK4}\left(@f, t_k, x_k, h\right), \tag{18}$$

where x denote the vector $[x; v; m; \lambda_r; \lambda_v; \lambda_m]$ with 14 dimensions. Since in Eq. (13) the value of the normalized thrust magnitude u has three possibilities, the right-hand side function of Eq. (7)

$$f = \begin{cases} v \\ -\frac{\mu}{r^3} r - \frac{\lambda_v}{\|\lambda_v\|} \frac{T_{\max}}{m} u \\ -\frac{T_{\max}}{I_{sp} g_0} u \\ \frac{\mu}{r^3} \lambda_v - \frac{3\mu r \cdot \lambda_v}{r^5} r \\ -\lambda_r \\ -\|\lambda_v\| \frac{T_{\max}}{m^2} u \end{cases} \tag{19}$$

can be formed to three types, denoted by f_1, f_2, and f_3, corresponding to u=0, $1/2 - \rho/(2\varepsilon)$, and 1, i.e. $\rho > \varepsilon$, $|\rho| \leq \varepsilon$, and $\rho < -\varepsilon$, respectively. Though the thrust magnitude becomes continuous, the derivative does not exist at the point $\rho = \pm\varepsilon$. Since switch function as well as its derivative is generally continuous, the computation from x_k to x_{k+} can be classed into three cases: $\rho_k > \varepsilon$, $|\rho_k| \leq \varepsilon$, and $\rho_k < -\varepsilon$, their flow charts are sketched as Figs. 1–3, respectively, where $\rho = \mathrm{SF}(x)$ denotes the switch function with respect to states and costates. For the step h small enough, the switch function is always assumed to be linear with respect to time during the interval.

5 Numerical Example

To solve nonlinear Eq. (10), among which the integrator RK4 with switch function detection is used, a set of good initial guess values for the costates are necessary. Although the convergence domain of the optimal control problem modified by adding a quadratic penalty is enlarged much, the initial guess values are still important. On the one hand, if the guess values are not in the convergence domain, converged results cannot be obtained. On the other hand, if the guess values are given on a convergence domain of local minimal, what obtained is local optimal, but not global optimal. Therefore, the PSODE is used to find a set of good initial values for the costates, while the fuel consumption needs to be minimized, and the constraints on final states are added to the performance index by multiplying a set of penalty factors:

$$\mathrm{obj} = -m\left(t_f\right) + \mu_r \left\|r\left(t_f\right) - r_f\right\|^2 + \mu_v \left\|v\left(t_f\right) - v_f\right\|^2, \tag{20}$$

Fig. 1 Flow chart for $\rho_k > \varepsilon$

$$
\begin{aligned}
&\boldsymbol{x}_{k+1} = \mathrm{RK4}\left(@\boldsymbol{f}_1, t_k, \boldsymbol{x}_k, h\right)\\
&\rho_{k+1} = \mathrm{SF}\left(\boldsymbol{x}_{k+1}\right)\\
&\text{if } \left(\rho_{k+1} < \varepsilon\right)\\
&\quad h_1 = \frac{\varepsilon - \rho_k}{\rho_{k+1} - \rho_k} h\\
&\quad \boldsymbol{x}_{k+1} = \mathrm{RK4}\left(@\boldsymbol{f}_1, t_k, \boldsymbol{x}_k, h_1\right)\\
&\quad \text{if } \left(\rho_{k+1} \geq -\varepsilon\right)\\
&\qquad \boldsymbol{x}_{k+1} = \mathrm{RK4}\left(@\boldsymbol{f}_2, t_k + h_1, \boldsymbol{x}_{k+1}, h - h_1\right)\\
&\quad \text{else}\\
&\qquad h_2 = \frac{-2\varepsilon}{\rho_{k+1} - \rho_k} h\\
&\qquad \boldsymbol{x}_{k+1} = \mathrm{RK4}\left(@\boldsymbol{f}_2, t_k + h_1, \boldsymbol{x}_{k+1}, h_2\right)\\
&\qquad \boldsymbol{x}_{k+1} = \mathrm{RK4}\left(@\boldsymbol{f}_3, t_k + h_1 + h_2, \boldsymbol{x}_{k+1}, h - h_1 - h_2\right)\\
&\quad \text{end}\\
&\quad \rho_{k+1} = \mathrm{SF}\left(\boldsymbol{x}_{k+1}\right)\\
&\text{end}
\end{aligned}
$$

where μ_r and μ_v are the penalty factors on final position and velocity, respectively. The optimization variables, i.e. initial costates, are searched on large domain, for example, $[-50,000, 50,000]$. It is very difficult to obtain the solution that exactly satisfies the constraints on final states. Therefore, we set penalty factors to be very small to pay more attention to the fuel consumption than to the constraints on final states. When scaling the length quantity by AU and the time by Julian years, the factors μ_r and μ_v are both set to be 0.0001. After getting the initial costates that violate the constraints not too seriously, Matlab's `fsolve` is used to solve nonlinear equation, and the terms "TolFun" and "TolX" are both set to be $1.0e-8$. The fixed step is set to be 0.0005 Julian years.

5.1 Example 1

From the Earth to rendezvous with Venus. Depart at 2005-10-7 0:0:0.0 (UTC), and the flight time is 1,000 Julian days. I_{sp}=3,800 s, $T_{\max} = 0.33$ N, and m_0=1,500 kg. The costate initial values for $\varepsilon = 1$ obtained from PSODE is [1,1276, 981.99, 5,000, −479.2, 2,193.3, −475.14, −9.5438], which is very rough because the position error is 0.008AU and the velocity error is 1.47 km/s. While it can be used as initial guess values for the method presented.

Fig. 2 Flow chart for $|\rho_k| \le \varepsilon$

$$
\begin{aligned}
&\boldsymbol{x}_{k+1} = \mathrm{RK4}\left(@\boldsymbol{f}_2, t_k, \boldsymbol{x}_k, h\right)\\
&\rho_{k+1} = \mathrm{SF}\left(\boldsymbol{x}_{k+1}\right)\\
&\text{if}\left(\rho_{k+1} > \varepsilon\right)\\
&\quad h_1 = \frac{\varepsilon - \rho_k}{\rho_{k+1} - \rho_k} h\\
&\quad \boldsymbol{x}_{k+1} = \mathrm{RK4}\left(@\boldsymbol{f}_2, t_k, \boldsymbol{x}_k, h_1\right)\\
&\quad \boldsymbol{x}_{k+1} = \mathrm{RK4}\left(@\boldsymbol{f}_1, t_k + h_1, \boldsymbol{x}_{k+1}, h - h_1\right)\\
&\quad \rho_{k+1} = \mathrm{SF}\left(\boldsymbol{x}_{k+1}\right)\\
&\text{elseif}\left(\rho_{k+1} < -\varepsilon\right)\\
&\quad h_1 = \frac{-\varepsilon - \rho_k}{\rho_{k+1} - \rho_k} h\\
&\quad \boldsymbol{x}_{k+1} = \mathrm{RK4}\left(@\boldsymbol{f}_2, t_k, \boldsymbol{x}_k, h_1\right)\\
&\quad \boldsymbol{x}_{k+1} = \mathrm{RK4}\left(@\boldsymbol{f}_3, t_k + h_1, \boldsymbol{x}_{k+1}, h - h_1\right)\\
&\quad \rho_{k+1} = \mathrm{SF}\left(\boldsymbol{x}_{k+1}\right)\\
&\text{else}\\
&\text{end}
\end{aligned}
$$

The value of ε is decreased from 1.0 to 0.000001 through only eight steps, of which the solution to $\varepsilon = 0.001$ is accurate enough. The series of results are listed in Table 1, and the thrust profiles are depicted in Fig. 4. So, the final result for $\varepsilon = 0.000001$ is accurate enough to be regarded as an optimal solution, of which the thrust profile and corresponding switch function are depicted in Fig. 5. The value of switch function less than zero maps the maximal thrust magnitude, while the value more than zero maps null thrust. The pitch and yaw angles are depicted in Fig. 6. Here, the minimal fuel consumption is 209.422 kg, and that obtained by Bertrand et al. is 210 kg. For comparison, the thrust profiles obtained by Bertrand et al. are shown by Fig. 7. The middle results are different from our solutions because the logarithmic barrier was used there, while the quadratic penalty is used here. Nevertheless, the final results are the same. The result of $\varepsilon = 0.001$, which is decreased from 1.0 through only six steps, is accurate enough. In every step, the required iteration number is small. However, the final result, corresponding to $\varepsilon = 0.00001$, obtained by Bertrand et al. was decreased from 0.1 through 1,000 steps. Therefore, our method is more effective in computations.

Fig. 3 Flow chart for $\rho_k < -\varepsilon$

$$
\begin{aligned}
&\boldsymbol{x}_{k+1} = \mathrm{RK4}\left(@\boldsymbol{f}_3, t_k, \boldsymbol{x}_k, h\right)\\
&\rho_{k+1} = \mathrm{SF}\left(\boldsymbol{x}_{k+1}\right)\\
&\text{if}\left(\rho_{k+1} > -\varepsilon\right)\\
&\quad h_1 = \frac{-\varepsilon - \rho_k}{\rho_{k+1} - \rho_k} h\\
&\quad \boldsymbol{x}_{k+1} = \mathrm{RK4}\left(@\boldsymbol{f}_3, t_k, \boldsymbol{x}_k, h_1\right)\\
&\quad \text{if}\left(\rho_{k+1} \le \varepsilon\right)\\
&\qquad \boldsymbol{x}_{k+1} = \mathrm{RK4}\left(@\boldsymbol{f}_2, t_k + h_1, \boldsymbol{x}_{k+1}, h - h_1\right)\\
&\quad \text{else}\\
&\qquad h_2 = \frac{2\varepsilon}{\rho_{k+1} - \rho_k} h\\
&\qquad \boldsymbol{x}_{k+1} = \mathrm{RK4}\left(@\boldsymbol{f}_2, t_k + h_1, \boldsymbol{x}_{k+1}, h_2\right)\\
&\qquad \boldsymbol{x}_{k+1} = \mathrm{RK4}\left(@\boldsymbol{f}_1, t_k + h_1 + h_2, \boldsymbol{x}_{k+1}, h - h_1 - h_2\right)\\
&\quad \text{end}\\
&\quad \rho_{k+1} = \mathrm{SF}\left(\boldsymbol{x}_{k+1}\right)\\
&\text{end}
\end{aligned}
$$

Table 1 Iteration results

n	ε	Iter.	Δm (kg)			Costate	Initial	Values		
1	1.0	53	225.02	256.5	−1,122.8	2,268.2	−50.618	76.744	−177.68	0.7779
2	0.5	14	217.8	3,021.4	−821.32	3,585.6	−141.23	526.63	−87.968	0.9717
3	0.3	14	215.95	4,622.9	−483.92	3,929.1	−197.44	776.91	3.0777	1.0419
4	0.1	22	212.9	6,353.7	58.851	3,886.3	−258.89	1,033.4	135.46	1.1034
5	10^{-2}	27	209.57	7,117.9	497.46	3,435.6	−284.65	1,132.2	228.65	1.1123
6	10^{-3}	9	209.42	7,123.7	512.47	3,411.6	−284.47	1,132.4	230.86	1.1117
7	10^{-5}	4	209.42	7,123.8	512.60	3,411.3	−284.47	1,132.4	230.88	1.1117
8	10^{-6}	2	209.42	7,123.8	512.60	3,411.3	−284.47	1,132.4	230.88	1.1117

5.2 Example 2

The third segment of our GTOC3 (Third Global Trajectory Optimization Competition) results, where the spacecraft departures from asteroid 2006 JY26 at 59806.8411 (MJD) to rendezvous with asteroid 2000 SG344 at 60470.0672 (MJD). $I_{sp} = 3,000$ s, $T_{\max} = 0.15$ N, and $m_0 = 1,807.546$ kg.

The costate initial values for $\varepsilon = 1$ obtained from PSODE is [−22,625, 3,538.7, 3,381.6, 4,215.7, 639.59, −499.99, −11.744], which is very rough because the position error is 0.00034 AU and the velocity error is 3.62 km/s. While it can be used as initial guess values for the method presented.

The value of ε is decreased from 1.0 to 0.000001 through only nine steps, of which the solution to $\varepsilon = 0.001$ is accurate enough. The series of results are listed in Table 2, and the thrust profiles are depicted in Fig. 8. So, the final result for $\varepsilon =$

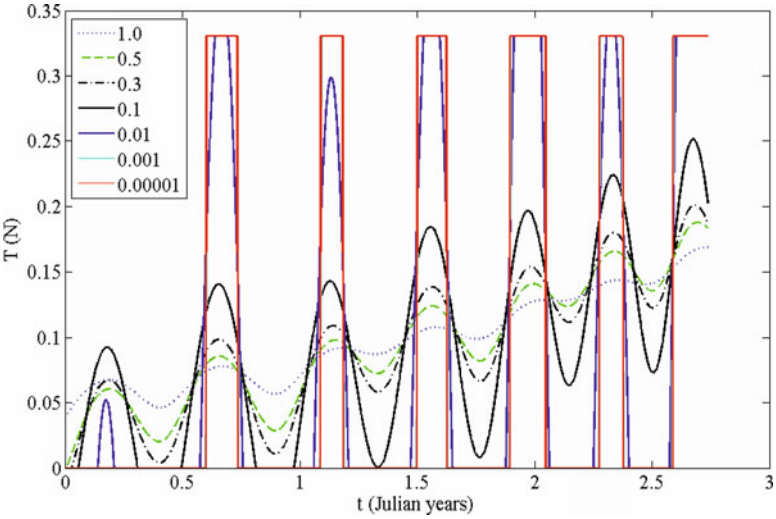


Fig. 4 Thrust profiles of iteration results

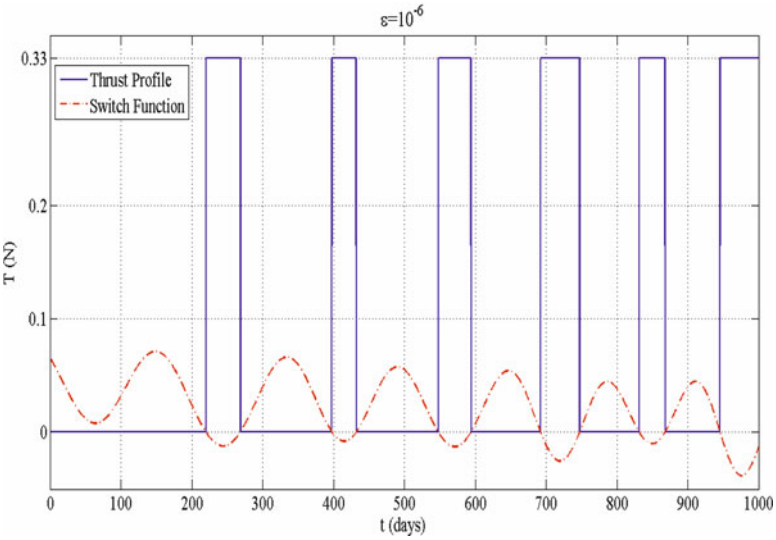


Fig. 5 Thrust profile and switch function of optimal result

0.000001 is accurate enough to be regarded as an optimal solution, of which the thrust profile and corresponding switch function are depicted in Fig. 9. The value of switch function less than zero maps the maximal thrust magnitude, while the value more than zero maps null thrust. The pitch and yaw angles are depicted in Fig. 10. Here, the minimal fuel consumption is 167.48 kg, and that obtained through direct method is 182.76 kg. For comparison, the thrust profiles obtained through direct

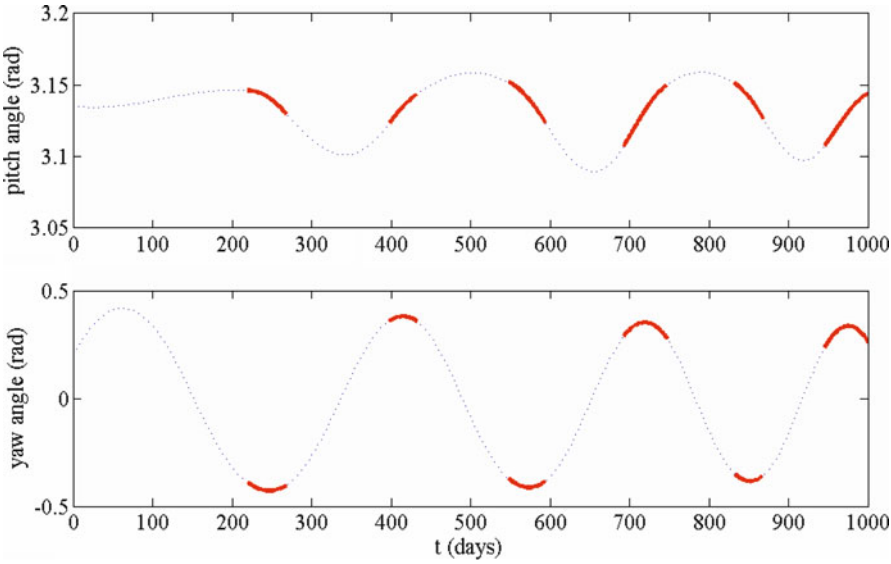


Fig. 6 Steering angles of optimal result

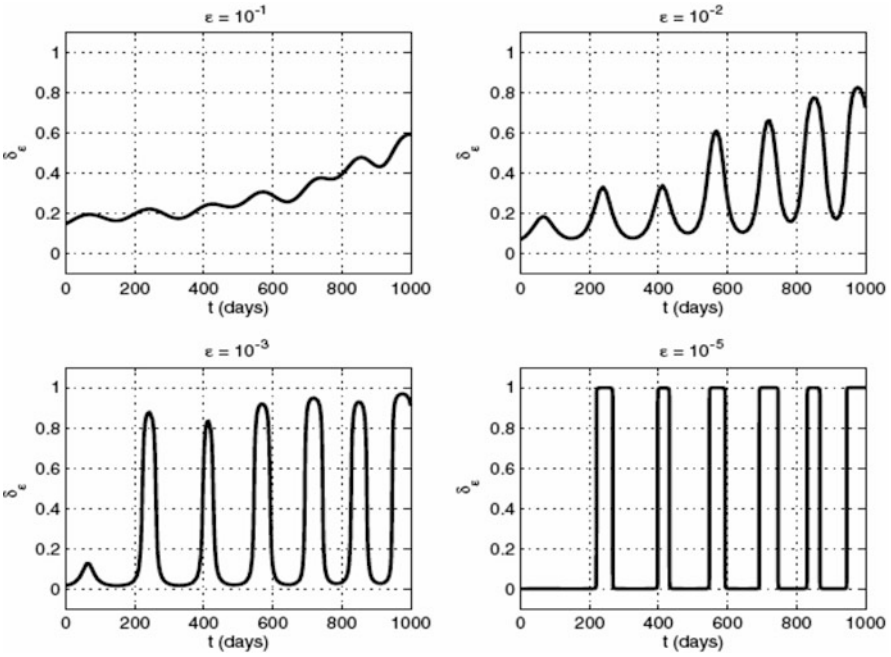


Fig. 7 Thrust profiles of Bertrand's results

Table 2 Iteration results

n	ε	Iter.	Δm (kg)			Costate	Initial	Values		
1	1.0	19	173.99	−8,963.8	−559.18	−1,139.9	1,649.2	0.1788	−1,128.1	0.7384
2	0.5	16	171.81	−7,360.6	525.51	−1,297.9	1,331.7	146.87	−1,120.3	0.6674
3	0.3	11	170.41	−6,552.2	1,109.0	−1,484.6	1,169.9	226.96	−1,165.5	0.6379
4	0.2	10	169.61	−6,115.2	1,401.4	−1,639.7	1,082.1	267.32	−1,213.9	0.6233
5	0.1	13	168.71	−5,589.4	1,719.0	−1,873.3	977.36	310.58	−1,295.4	0.6066
6	10^{-2}	20	167.56	−5,066.5	1,971.5	−2,101.0	882.35	338.57	−1,387.9	0.5888
7	10^{-3}	33	167.48	−5,044.7	1,970.5	−2,126.4	878.77	337.85	−1,393.7	0.5877
8	10^{-4}	2	167.48	−5,044.2	1,970.5	−2,127.0	878.69	337.85	−1,393.8	0.5877
9	10^{-6}	1	167.48	−5,044.2	1,970.5	−2,127.0	878.69	337.85	−1,393.8	0.5877

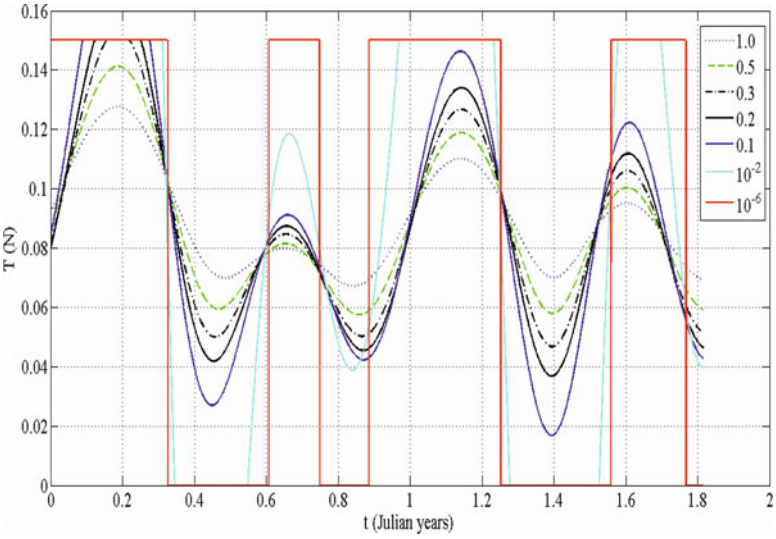


Fig. 8 Thrust profiles of iteration results

method are shown by Fig. 11, which is different from the optimal results somewhat largely.

Acknowledgments The authors are supported by the National Natural Science Foundation of China (Grants No: 60625304, 60621062).

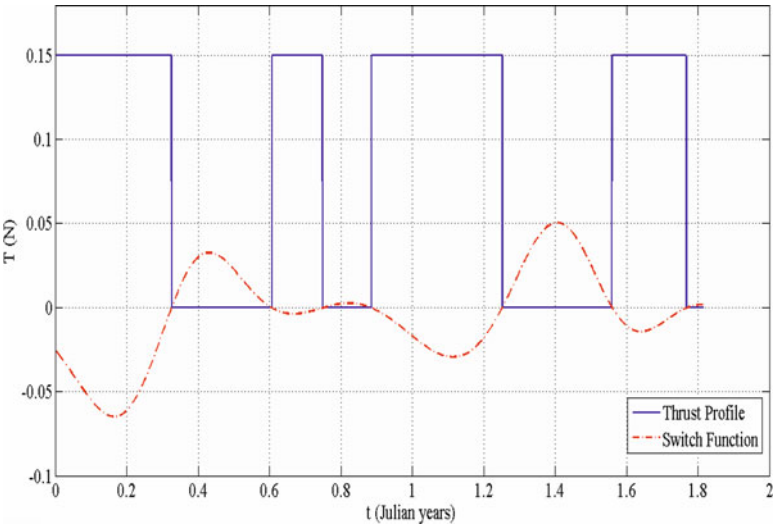


Fig. 9 Thrust profile and switch function of optimal result

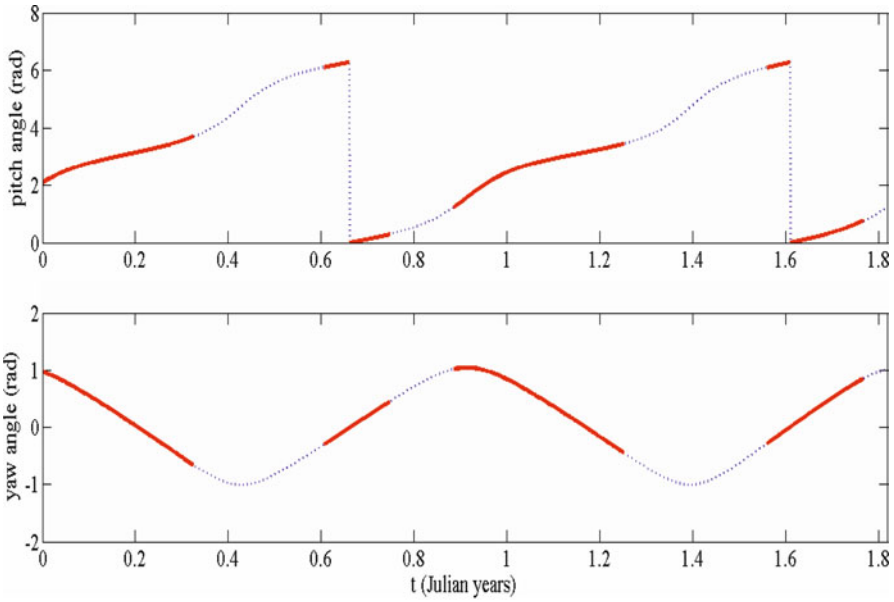


Fig. 10 Steering angles of optimal result

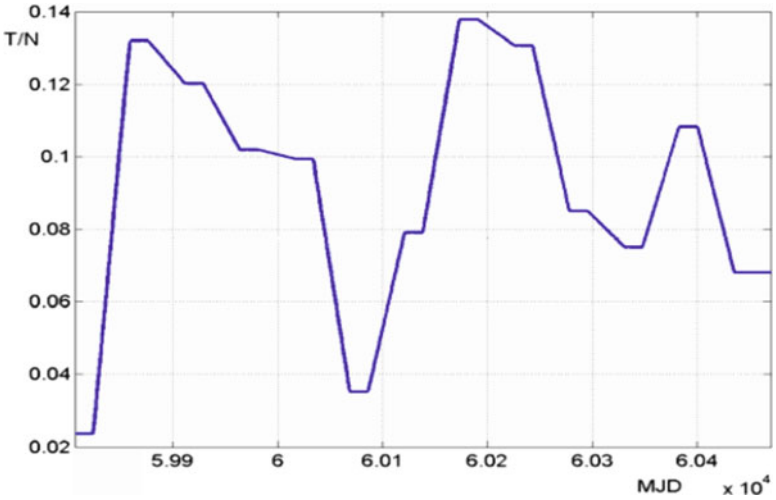


Fig. 11 Thrust profile obtained through direct method

References

1. Hargraves, C., and Paris, S.. Direct Trajectory Optimization Using Nonlinear Programming and Collocation. Journal of Guidance, Control, and Dynamics, 1987, 10(4): 338–342.
2. Enright, P. J., and Conway, B. A.. Discrete Approximations to Optimal Trajectories Using Direct Transcription and Nonlinear Programming. Journal of Guidance, Control, and Dynamics, 1992, 15(4): 994–1002.
3. Pontryagin, L. S., Boltyanskii, V. G., Gamkrelize, R. V., and Mishchenko, E. F.: The Mathematical Theory of Optimal Processes. In: Wiley, New York, 1962, Chap. 2.
4. Kechichian, J. A.. Optimal low thrust orbit geostationary Earth-orbit intermediate acceleration orbit transfer. Journal of Guidance, Control, and Dynamics, 1997, 20(4): 803–811.
5. Ranieri, C. L., and Ocampo, C. A.. Indirect optimization of three-dimensional finite-burning interplanetary transfers including spiral dynamics. Journal of Guidance, Control, and Dynamics, 2009, 32(2): 444–454.
6. Ilgen, M. R.. Hybrid Method for Computing Optimal Low Thrust OTV Trajectories. Advances in the Astronautical Sciences, 1992, 87(2): 941–958.
7. Bertrand, R., and Epenoy, R.. New smoothing techniques for solving bang-bang optimal control problems-numerical results and statistical interpretation. Optimal Control Applications and Methods, 2002, 23: 171–197.

A Collection of Test Multiextremal Optimal Control Problems

Alexander Yu. Gornov, Tatiana S. Zarodnyuk, Taras I. Madzhara, Anna V. Daneeva, and Irina A. Veyalko

Abstract This chapter considers a collection of test optimal control problems that have been applied to test the efficiency of algorithms for many years. The techniques of comparative testing, statistical testing, and stress testing are used for creating problems of this set. The tests are designed in the same format: there is information about the known local extrema, optimal control and trajectory, attainable set approximation, and the number of Cauchy problems required to obtain the optimal value of an objective functional in each test. Currently the implemented collection includes about 100 test cases.

Key words Test collection • Optimal control problem • Attainability set

1 Introduction

Testing is the basic method to experimentally estimate the efficiency of optimization algorithms and programs. There is a great number of publications on the mathematical programming problems that are devoted to this issue; we cannot but mention [3, 11, 12, 19, 21, 23, 28]. The Mathematical Programming Society has developed recommendations that represent a methodological approach to preparation of software testing results [6]. Considerable part of these recommendations can be modified for optimal control problems (OCP).

A.Yu. Gornov (✉) • T.S. Zarodnyuk • T.I. Madzhara • A.V. Daneeva • I.A. Veyalko
Institute for System Dynamics and Control Theory SB RAS, 134 Lermontov St., 664033 Irkutsk, Russia
e-mail: gornov@icc.ru; tz@icc.ru; taras@icc.ru; rozen@icc.ru; veyalko@icc.ru

A. Chinchuluun et al. (eds.), *Optimization, Simulation, and Control*, Springer Optimization and Its Applications 76, DOI 10.1007/978-1-4614-5131-0_16,

All testing methods are based on the collections (libraries) of test problems. A series of test collections have been designed and used for different optimization problems. The present approach has received recognition among many experts working in this area. The quality of individual problems or the entire set only can be disputable but not the approach itself [5, 8, 12, 19, 24, 30, 35]. In order to achieve the goals of testing, to obtain objective information on method behavior, to find classes of problem types for which the method is most efficient, and to create the method versions that are most independent of computers, test problems should meet the following requirements [25]:

1. Tests should be standardized and universally recognized.
2. Tests should model typical difficulties for a specified class of problems.
3. A solution to test problem should be known.
4. Problems should be sufficiently compact.
5. The problems making one method more advantageous than the others cannot be used for tests.

In most cases the requirement of known solution to OCP is unconstructive. Normally analytical methods of the problem analysis turn out to be inapplicable to nonlinear OCP with terminal or phase constraints. Even the auxiliary problems of integrating the system of differential equations or the problems of one-dimensional search of function extremum in the descent direction can be impossible to solve without numerical approaches. The principle of "the best of known solutions" [12] is suggested for solving this problem. This principle is popular in global optimization, where common methods of analytical research for practically significant problems have not been found yet either. The reference solution to an OCP which is used to compare all the other solutions to is obtained by solving the problem with highly overestimated accuracy of each algorithm, which may require long computations but is done once when a test problem is designed.

There are many works devoted to creation of test OCP. First of all it is necessary to mention the collection developed by the group of scientists under the guidance of professor K. Schittkowski. The big collection (A Collection of 1,300 Dynamical Systems for Testing Data Fitting, Optimal Control, Experimental Design, Identification, Simulation or Similar Software [26, 27]) includes about 40 OCP of dynamic systems. Unfortunately, the examples considered in this source were studied only on the basis of local algorithms that have been developed by the mentioned group of specialists. There are test collections designed by the group of specialists headed by professor Betts [4] and collection of professor Teo [31]. However, according to our information, for the time being, there are no generally recognized and widely known test collections of multiextremal problems similar to collections of tests for mathematical programming problems and for OCP.

In this chapter consideration is given to the collection of test problems that have been applied by the authors for many years to test the efficiency of algorithms. The collection contains both the OCP statements known from literature and new problems either constructed on the basis of special techniques or resulting from

implementation of applied projects. The number of Cauchy problems required for operation of all algorithm components is taken as a criterion of algorithm labor intensity. This characteristic is the least dependent on the technical features of the applied computer systems. Besides, in the majority of cases, the Cauchy problem solving takes most of the algorithm operation time.

2 Testing Methods

The methods of testing the optimization algorithms should take into account specific features of the software products intended for solving multiextremal problems. At the same time the following factors should be considered [25]: (1) we do not compare algorithms or methods but their software implementations, (2) testing results can very much depend on how the criteria of problem solving laboriousness are chosen, and (3) behavior of algorithm characteristics is not monotonic at different stages of the problem solving process. The methods applied to create the test problem collection are comparative testing, statistical testing, and stress testing [15].

The comparative testing suggests solving test problems by various algorithms and comparing the results in order to choose the best software implementations of the algorithms. The comparison of calculation results for one problem by different methods is obviously the simplest and, nevertheless, a very efficient way to obtain empirical information about the studied algorithm. Specialists have always believed that the number of successful applications of a method in comparison with other methods is a reliable criterion of the method quality.

The technique of statistical testing is based on specialized parameterized tests with a random selection of their parameters and makes it possible to judge that the algorithm behavior is nonlocal. The method of statistical testing ("stochastic testing") is widely used for other optimization problems, but it is very seldom applied to OCP so far. For OCP the methods of generating test cases for statistical testing should take into consideration the following traditional factors: the possible test sensitivity to variations in its parameters, the possible existence of "abnormal end zones" in generated test cases, and the possible deterioration of differential properties of the test case which leads to troubles when auxiliary problems are solved.

The testing methods can give good information about the speed of the algorithm convergence, but they do not help much if its limiting properties should be studied. It is obvious that with the rarest exception, the problem solution can be obtained only approximately, since it is inevitably affected by rounding and discretization errors. In addition, the establishment of precise boundaries for reliable results of the algorithm is still a complex task which requires special approaches to be used. The stress testing aims to obtain information about the limiting properties of a software system and represents a kind of "proving ground" for the algorithms. The technique

is based on a special set of test cases, which are oriented to typical features of optimization problems.

The stage of operation testing can be considered to be the most important step of algorithm completion, since at this stage the most subtle errors are detected. Duration of this stage depends on the algorithm life cycle and can vary for its different versions. However, only long-term use of the algorithm software implementation and the number of solved problems can make the developer be sure of its reliability.

3 Algorithms

A set of algorithms were used to find a local extremum. The algorithms are based on the optimal control theory and the theory of finite-dimensional optimization. The basic set of methods includes conventional and reduced gradient methods based on the Pontryagin maximum principle, the conjugate gradient method, a one-parameter combination of quasi-Newton methods BFGS and DFP, spectral projected gradient of Birgin-Yevtushenko, Nesterov's ravine methods, and Powell-Brent's search methods [19]. The algorithms from the basic set were applied to design multi-method numerical schemes that allow one to take advantages of different methods at different stages of problem solving process. Search of global extremum is provided by the multi-start method, the curvilinear search method, the net-point method, and the tunneling method [14, 16–18, 33, 34]. To construct the attainability set we employed the stochastic approximation method and the method based on the necessary optimization conditions [13].

4 Uni-extremum Optimal Control Problem

A great amount of the studied nonlinear problems, to our surprise, had only one local extremum. Probably this is related to the fact that in the process of modeling the model constraints were found under which the problem developer could interpret the solutions to be obtained. The "effect of one extremum" as a rule disappeared with significant expansion of the studied area of problem parameters, for example, with increase in time interval.

Problem 1. In the first test the linear functional should be minimized on the nonconvex set described by a bilinear system of differential equations. In the initial formulation the objective functional to be minimized is of integral type [29] and is brought to the terminal form by standard transformations (Table 1).

Table 1 Numerical solutions to test problem 1

N	Functional value	Extreme points
1	−1.33330	(0.66900,−1.33330)
The number of Cauchy problems:		395

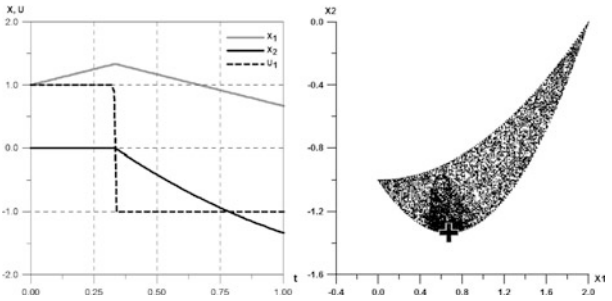
Table 2 Numerical solutions to test problem 2

N	Functional value	Extreme points
1	2.64816	(−0.21602, −0.99348)
The number of Cauchy problems:		65

Statement of test problem 1

$$\dot{x}_1 = u$$
$$\dot{x}_2 = x_1(u-1)$$
$$t \in [0,1]$$
$$x_0 = (1,0)$$
$$u \in [-1,1]$$
$$I(u) = x_2(t_1) \to \min$$

Optimal trajectories and control; attainability set and extreme points



Problem 2. The process of control by Duffing oscillator is described by the nonlinear differential equation $\ddot{x} + \omega^2 x + \epsilon x^3 = u$ [9]. For this test case the following values of parameters used in the initial statement were selected: $\omega = \epsilon = 1$ (Table 2).

Statement of test problem 2

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -\omega^2 x_1 - \epsilon x_1^3 + u$$
$$t \in [0,1]$$
$$x_0 = (1.5,-1.5)$$
$$u \in [-10,10]$$
$$I(u) = 0.5\int_0^{1.5} u^2(t)\mathrm{d}t + x_1^2(t_1) + x_2^2(t_1) \to \min$$

Optimal trajectories and control; attainability set and extreme points

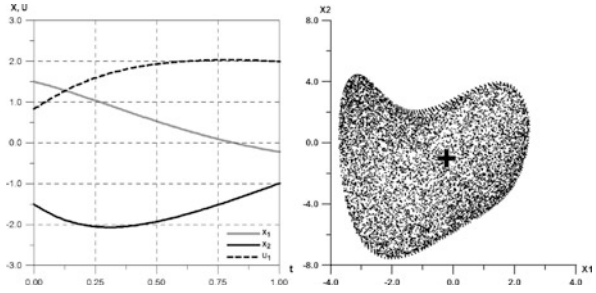


Table 3 Numerical solutions to test problem 3

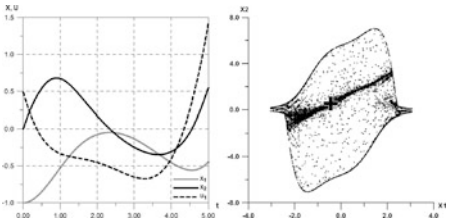
N	Functional value	Extreme points
1	1.56317	(−0.44590, 0.54705)
The number of Cauchy problems:		427

Problem 3. A well-known Van der Pol problem in different statements is used for testing software implementation of algorithms. We propose a modification of the problem presented in [10] (Table 3).

Statement of test problem 3

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -(x_1^2 - 1)^2 x_2 - x_1 + u$$
$$t \in [0, 5]$$
$$x_0 = (-1, 0)$$
$$u \in [-10, 10]$$
$$I(u) = 0.5 \int_0^5 (x_1^2(t) + x_2^2(t) + u^2(t))\mathrm{d}t + 100(x_1^2(t_1) - x_2(t_1) + 1)^2 \to \min$$

Optimal trajectories and control; attainability set and extreme points



5 Multiextremal Optimal Control Problems

The sets of problems of different computational complexity are easily identified among multiextremal problems. Usually this is due to the size of the domain of attraction of the global extremum. In some cases the problem complexity is determined by substantially different properties of the dynamic system in different regions of the time interval. As a rule the problems with long time intervals are more complicated than the problems with smaller time intervals with the same dynamic equations.

Problem 4. The classical example of test dynamic problem is the OCP of nonlinear pendulum (e.g., [1, 32]). The oscillation process is described by the system of nonlinear differential equations (Table 4).

Statement of test problem 4

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = u - \sin x_1$$
$$t \in [0, 5]$$
$$x_0 = (5, 0)$$
$$u \in [-1, 1]$$
$$I(u) = x_1^2(t_1) + x_2^2(t_1) \to \min$$

Optimal trajectories and control; attainability set and extreme points

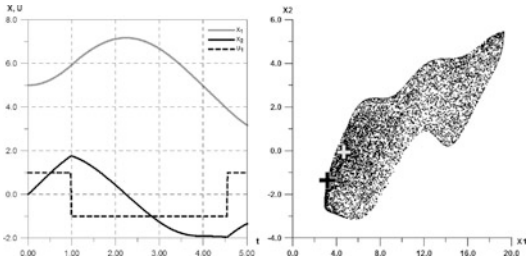


Table 4 Numerical solutions to test problem 4

N	Functional value	Extreme points
1	11.90876	(3.17863,−1.34354)
2	21.82921	(4.67180,−0.05900)
The number of Cauchy problems:		228

Table 5 Numerical solutions to test problem 5

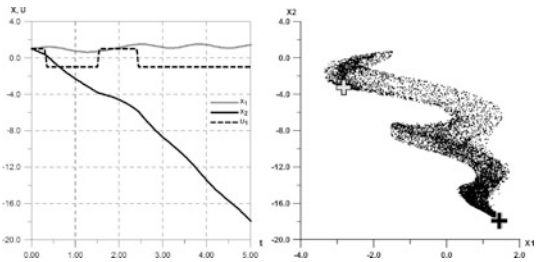
N	Functional value	Extreme points
1	−16.45670	(1.44528,−17.90198)
2	−6.06870	(−2.83105,− 3.23765)
The number of Cauchy problems:		433

Problem 5. The following test problem was generated on the basis of a developed method of test construction. It is necessary to find a control, which minimizes the terminal functional (Table 5).

Statement of test problem 5

$$\dot{x}_1 = \sin x_2$$
$$\dot{x}_2 = u - e^{x_1}$$
$$t \in [0, 5]$$
$$x_0 = (1, 1)$$
$$u \in [-1, 1]$$
$$I(u) = x_1(t_1) + x_2(t_1) \to \min$$

Optimal trajectories and control; attainability set and extreme points



Problem 6. In this test the objective functional is nonconvex and the system of differential equations is nonlinear. The solution to problem 6 is four extrema: a global extremum and three local extrema (Table 6).

Statement of test problem 6

$$\dot{x}_1 = 1 - x_2^2 + 0.5x_2$$
$$\dot{x}_2 = x_1 u$$
$$t \in [0, 1.7]$$
$$x_0 = (0.5, -0.2)$$
$$u \in [-2, 2]$$
$$I(u) = x_1(t_1) - x_2(t_1) + x_1(t_1)x_2(t_1) \to \min$$

Optimal trajectories and control; attainability set and extreme points

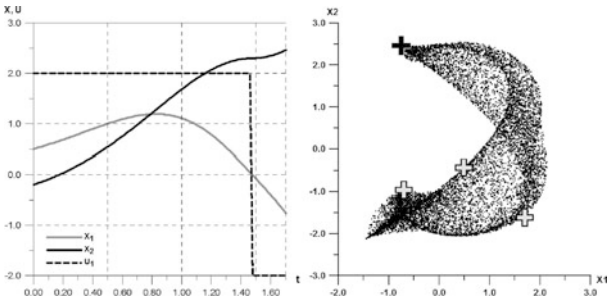


Table 6 Numerical solutions to test problem 6

N	Functional value	Extreme points
1	−5.06255	(−0.75065, 2.46304)
2	0.57058	(1.70119,−1.61242)
3	0.70637	(0.48686,−0.42777)
4	0.92837	(−0.69837,−0.95782)
The number of Cauchy problems:		1,342

Table 7 Numerical solutions to test problem 7

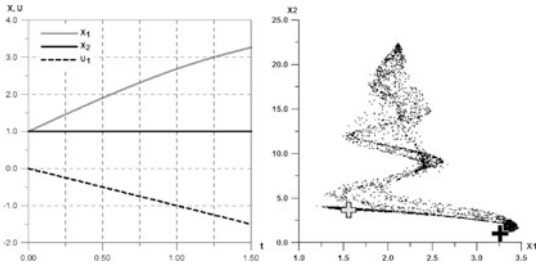
N	Functional value	Extreme points
1	4.25969	(3.25969, 1.00000)
2	5.24298	(1.55558, 3.68740)
The number of Cauchy problems:		91

Problem 7. Optimal control in this test problem is monotonically decreasing over the entire time interval (Table 7).

Statement of test problem 7

$$\dot{x}_1 = \sin x_2 + \cos t$$
$$\dot{x}_2 = (u+t)^2$$
$$t \in [0,\ 1.5]$$
$$x_0 = (1,1)$$
$$u \in [-3,3]$$
$$I(u) = x_1(t_1) + x_2(t_1) \to \min$$

Optimal trajectories and control; attainability set and extreme points



Problem 8. OCP of stirred-tank reactor [7,22] is included in the Handbook of Test Problems in Local and Global Optimization (Floudas and Pardalos, 1999) (Table 8).

Statement of test problem 8

$$\dot{x}_1 = -(2+u)(x_1+0.25) + $$
$$+(x_2+0.5)\mathrm{e}^{\frac{25x_1}{x_1+2}}$$
$$\dot{x}_2 = 0.5 - x_2 - (x_2+0.5)\mathrm{e}^{\frac{25x_1}{x_1+2}}$$
$$t \in [0,\ 0.78],\ x_0 = (0.09, 0.09)$$
$$u \in [0,5]$$
$$I(u) = \int_0^{0.78}(x_1^2(t) + $$
$$+x_2^2(t) + 0.1u^2(t))\mathrm{d}t \to \min$$

Optimal trajectories and control; attainability set and extreme points

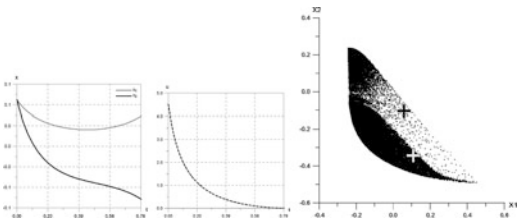


Table 8 Numerical solutions to test problem 8

N	Functional value	Extreme points
1	0.13313	(0.05803, −0.10263)
2	0.24445	(0.10844, −0.34219)
The number of Cauchy problems:		1,356

Table 9 Numerical solutions to test problem 9

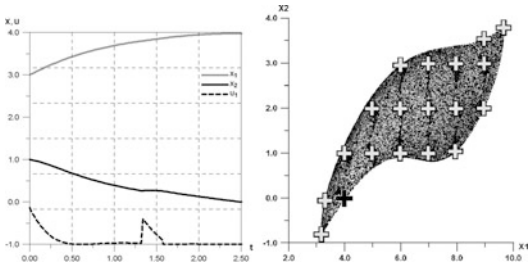
N	Functional value	Extreme points	N	Functional value	Extreme points
1	15.91924	(3.97978, 0.00000)	11	52.73182	(6.96442, 1.98991)
2	16.91420	(3.97978, 0.99496)	12	57.70659	(6.96442, 2.98486)
3	22.16332	(3.16550,−0.80595)	13	65.19579	(7.93279, 1.03957)
4	24.70471	(3.30062,−0.05904)	14	67.65549	(7.95923, 1.98991)
5	25.86868	(4.97469, 0.99496)	15	72.63026	(7.95923, 2.98486)
6	28.85355	(4.97469, 1.98991)	16	84.57997	(8.94777, 1.99435)
7	36.81295	(5.96957, 0.99496)	17	89.54350	(8.95400, 2.98486)
8	39.79782	(5.96957, 1.98991)	18	130.76930	(9.65684, 3.78789)
9	45.03334	(5.98533, 2.95192)	19	112.81680	(8.95986, 3.53954)
10	49.74695	(6.96442, 0.99496)			
The number of Cauchy problems:					5,584

Problem 9. The Rastrigin function [2] and a well-known system of differential equations with nonconvex attainable set were used for construction of the following test problem (Table 9):

Statement of test problem 9

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = u - \sin x_1$$
$$t \in [0, 2.5],$$
$$x_0 = (3, 1)$$
$$u \in [-1, 1]$$
$$I(u) = 20 + x_1^2(t_1) + x_2^2(t_1) - -10\sum_{i=1}^{2} \cos 2\pi x_i(t_1) \to \min$$

Optimal trajectories and control; attainability set and extreme points



Problem 10. Test problem 10 was generated on the basis of the developed method of test design (Table 10).

Table 10 Numerical solutions to test problem 10

N	Functional value	Extreme points
1	−2.83027	(2.70933, 2.21407)
2	0.95589	(2.44380, 2.63072)
3	1.06054	(0.61202, 2.09169)
4	1.11165	(0.56613,−0.81035)
5	1.18847	(0.16385,−0.57431)
The number of Cauchy problems: 2,049		

Table 11 Numerical solutions to test problem 11

N	Functional value	Extreme points
1	−4.81081	(−4.81081,−3.15829)
2	−1.56441	(−1.56441, 3.12855)
3	0.21573	(0.21573,−9.44594)
4	0.83683	(0.83683, 2.97624)
5	6.17400	(6.17400,−0.69621)
The number of Cauchy problems: 2,450		

Statement of test problem 10

$\dot{x}_1 = x_2 u$

$\dot{x}_2 = x_1 + \frac{u}{x_1^2 + x_2^2}$

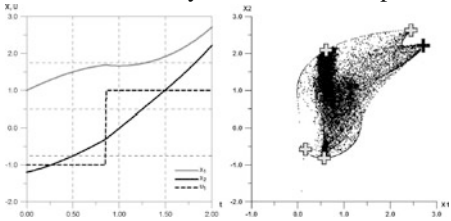
$t \in [0, 2]$

$x_0 = (1, -1.2)$

$u \in [-1, 1]$

$I(u) = (\sqrt{2.72 - x_1(t_1)} - \frac{2.72 - x_1(t_1)}{7} \cdot e^{0.926 x_1(t_1)}) - 1.481 x_1(t_1) - 0.014 x_2^2(t_1) \to \min$

Optimal trajectories and control; attainability set and extreme points



Problem 11. This OCP is formulated in the following way (Table 11):

Statement of test problem 11

$\dot{x}_1 = \cos x_2$

$\dot{x}_2 = u - \sin x_1$

$t \in [0, 7]$

$x_0 = (0, 0)$

$u \in [-1, 1]$

$I(u) = x_1(t_1) \to \min$

Optimal trajectories and control; attainability set and extremum points

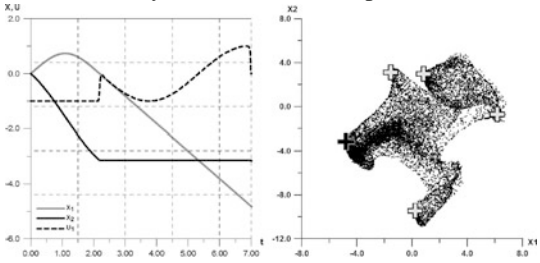


Table 12 Numerical solutions to test problem 12

N	Functional value	Extreme points
1	0.18870	(−1.43361, 9.52600)
2	28.39093	(0.24631, 4.31949)
3	149.90120	(−1.51735,−2.73248)
The number of Cauchy problems: 32,694		

Problem 12. This test problem is one of the most difficult among the presented problems. The numerical search of the global extremum is quite complicated in this case (Table 12).

Statement of test problem 12

$\dot{x}_1 = x_2$

$\dot{x}_2 = u - x_1 + \frac{x_1^3}{6} - \frac{x_1^5}{120}$

$t \in [0, 7]$

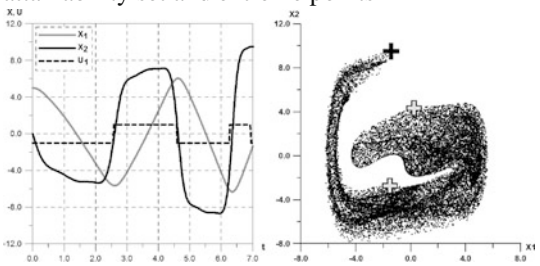
$x_0 = (5, 0)$

$u \in [-1, 1]$

$I(u) = (x_1(t_1) + 1)^2 +$

$+(x_2(t_1) - 9.5)^2 \to \min$

Optimal trajectories and control; attainability set and extreme points



6 Optimal Control Problems with Specific Features of Computation

The problems with "abnormal end zones" in the model, the so-called abend problems, can be very difficult to study numerically. It can be hard for this class of problems to find an initial approximation for the algorithms to start improving control. In addition, abends can occur already during the optimization process if too large variations are generated and disturb the regular states of the process.

The parameter continuation method can be considered a general approach to successfully solving the OCP with specific features of computation [20]. In our case this method means adding parameter $p \in [0, 1]$ to the system of differential equations $\dot{x} = p \cdot f(x, u, t)$. The parameter makes it possible to construct a set of OCP, in which the last problem coincides with the initial statement.

Problem 13. The numerical solution to this test problem is presented in Table 13. Parameter p defines an auxiliary problem on a parametric set. The result of the software operation is presented in the third column of this table (FAIL—program crash, OK—regular work). $I_p(u)$ is a best value of aim functional in auxiliary problem.

Table 13 Numerical solutions to test problem 13

p	Iterations	Status	$I_p(u)$
1.00000	1	FAIL	–
0.90000	1	FAIL	–
0.50000	1	FAIL	–
0.20000	32	OK	1.00787
0.40000	9	OK	0.78160
0.60000	19	OK	0.55372
0.80000	43	OK	0.32616
1.00000	206	OK	0.10627

Statement of test problem 13

$\dot{x}_1 = u + \log_{10} x_2$

$\dot{x}_2 = (x_1 + u)\log_{10} x_1 + 0.01u^2$

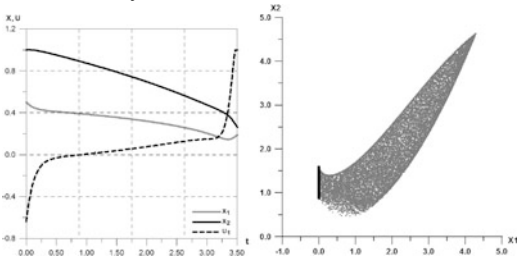
$t \in [0, 3.7]$

$x_0 = (0.5, 1)$

$|u| \le 1, u^0(t) = 1$

$I(u) = x_1^2(t_1) + x_2^2(t_1) \to \min$

Optimal trajectories and control; attainability set



Problem 14. The figure below presents an approximation of the attainable set which contains an "abnormal end zone" indicated in black and the projection of trajectory values which were caused by abends (Table 14).

Statement of test problem 14

$\dot{x}_1 = x_2 + u$

$\dot{x}_2 = x_1 - u - \sqrt{x_1^2 - 0.5x_2^2}$

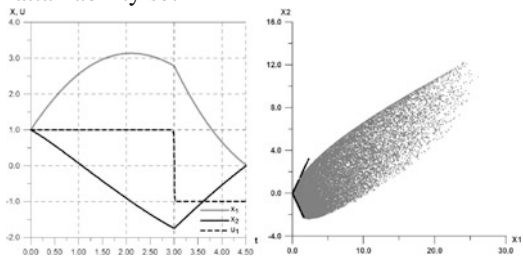
$t \in [0, 4.5]$

$x_0 = (1, 1)$

$|u| \le 1, u^0(t) = 1$

$I(u) = x_1^2(t_1) + x_2^2(t_1) \to \min$

Optimal trajectories and control; attainability set



Problem 15. In this test problem after two abends the value of parameter was selected to find the minimum value of the objective functional by four steps (Table 15).

Table 14 Numerical solutions to test problem 14

p	Iterations	Status	$I_p(u)$	p	Iterations	Status	$I_p(u)$
1.00000	0	FAIL	–	1.00000	0	FAIL	–
0.90000	0	FAIL	–	0.85000	10	OK	0.77887
0.50000	0	FAIL	–	0.90000	15	OK	0.36430
0.20000	0	FAIL	–	0.95000	25	OK	0.10023
0.10000	0	FAIL	–	1.00000	0	FAIL	–
0.05000	288	OK	2.58698	0.82000	57	OK	1.08869
0.06000	190	OK	2.70532	0.84000	22	OK	0.87758
0.08000	170	OK	2.94186	0.86000	0	OK	0.68509
0.10000	56	OK	3.17689	0.88000	3	OK	0.51338
0.20000	99	OK	4.26331	0.90000	11	OK	0.36430
0.30000	25	OK	5.01861	0.92000	7	OK	0.23949
0.40000	24	OK	5.23734	0.94000	26	OK	0.14009
0.60000	25	OK	3.88349	0.96000	8	OK	0.06699
0.80000	20	OK	1.31607	0.98000	20	OK	0.02059
1.00000	0	FAIL	–	1.00000	36	OK	0.00083
0.90000	39	OK	0.36430				

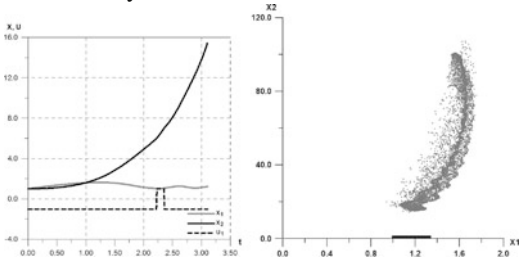
Table 15 Numerical solutions to test problem 15

p	Iterations	Status	$I_p(u)$
1.00000	1	FAIL	–
0.90000	1	FAIL	–
0.50000	1	OK	11.38532
0.60000	0	OK	19.87570
0.80000	0	OK	60.04049
1.00000	25	OK	237.82170

Statement of test problem 15

$$\dot{x}_1 = \sin\sqrt{x_1^2 + x_2^2 - 3u}$$
$$\dot{x}_2 = u + x_1 x_2$$
$$t \in [0, 3.1], x_0 = (1, 1)$$
$$|u| \le 1, u^0(t) = 0$$
$$I(u) = x_1^2(t_1) + x_2^2(t_1) \to \min$$

Optimal trajectories and control; attainability set



Problem 16. There are two "abnormal end zones" on the attainable set in test problem 16 (Table 16).

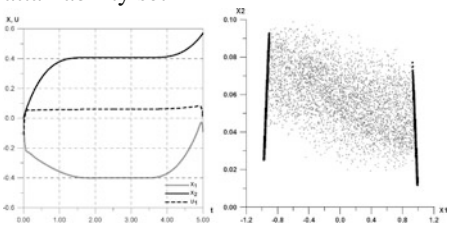
Table 16 Numerical solutions to test problem 16.

p	Iterations	Status	$I_p(u)$	p	Iterations	Status	$I_p(u)$
1.00000	0	FAIL	–	0.08000	15	OK	0.01499
0.90000	0	FAIL	–	0.10000	23	OK	0.02598
0.50000	0	FAIL	–	0.20000	60	OK	0.12548
0.20000	0	FAIL	–	0.30000	71	OK	0.27094
0.10000	0	FAIL	–	0.40000	52	OK	0.43209
0.05000	1	FAIL	–	0.60000	54	OK	0.76138
0.02000	11	OK	0.00057	0.80000	55	OK	1.09105
0.04000	7	OK	0.00269	1.00000	48	OK	1.42072
0.06000	12	OK	0.00730				

Statement of
test problem 16

$$\dot{x}_1 = -3 - \arcsin(x_1 + x_2) + 50u$$
$$\dot{x}_2 = 1 + \arcsin(x_1 - x_2) - u$$
$$t \in [0, 5]$$
$$x_0 = (0, 0)$$
$$|u| \le 1, u^0(t) = 0$$
$$I(u) = \int_0^5 (x_1^2(t) + x_2^2(t) + u^2(t))\mathrm{d}t \to \min$$

Optimal trajectories and control;
attainability set

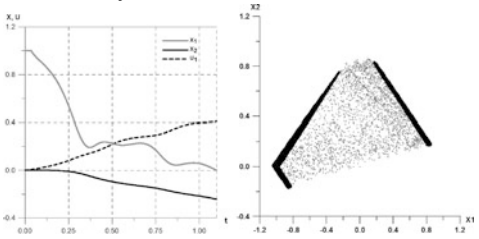


Problem 17. The numerical results for problem 17 are presented in Table 17 in a reduced form. When the value of parameter p increases from 0.620 to 0.800 with a step of 0.02 the functional value varies from 0.17718 to 0.30019.

Statement of
test problem 17

$$\dot{x}_1 = -1 - \arcsin(x_1 + x_2) + u + 5\sin(x_1 + x_2)$$
$$\dot{x}_2 = 1 + \arcsin(x_1 - x_2) - u + 0.2\cos 70x_1$$
$$t \in [0, 1.1], x_0 = (0, 0)$$
$$|u| \le 1, u^0(t) = 0$$
$$I(u) = \int_0^{1.1} (x_1^2(t) + x_2^2(t) + u^2(t))\mathrm{d}t \to \min$$

Optimal trajectories and control;
attainability set



7 Bang–Bang OCP

OCP with bang-bang control (Bang–Bang Control Problems) often arise in many scientific and applied fields. Solutions to nonsingular optimization problems of

Table 17 Numerical solutions to problem 17

p	Iterations	Status	$I_p(u)$	p	Iterations	Status	$I_p(u)$
1.00000	0	FAIL	–	0.54000	9	OK	0.12429
0.90000	0	FAIL	–	0.56000	0	FAIL	–
0.50000	0	FAIL	–	0.55000	9	OK	0.13122
0.20000	4	OK	0.00618	0.56000	10	OK	0.13817
0.40000	6	OK	0.04834	0.58000	0	FAIL	–
0.60000	0	FAIL	–	0.57000	12	OK	0.14508
0.50000	0	FAIL	–	0.58000	12	OK	0.15187
0.45000	6	OK	0.07004	0.60000	0	FAIL	–
0.50000	9	OK	0.09792	0.59000	13	OK	0.15851
0.55000	0	FAIL	–	0.60000	14	OK	0.16495
0.42000	5	OK	0.05632	0.80000	0	FAIL	–
0.44000	6	OK	0.06522	0.70000	0	FAIL	–
0.46000	6	OK	0.07511	0.65000	0	FAIL	–
0.48000	6	OK	0.08602	0.62000	25	OK	0.17718
0.50000	9	OK	0.09792	$\ldots\Delta p = 0.02000\ldots$			
0.52000	9	OK	0.11075	0.80000	23	OK	0.25181
0.54000	0	FAIL	–	1.00000	523	OK	0.30019
0.53000	9	OK	0.11745				

dynamic systems with linear control and without phase constraints are bang-bang solutions. The relay characteristic of optimal control can considerably simplify the problem of search for the functional optimum, since the size of a set of varied controls is much smaller in this case (Tables 18–20).

Problem 18.

$\dot{x}_1 = u - 3\cos x_2^2$

$\dot{x}_2 = e^{x_1} - tu$

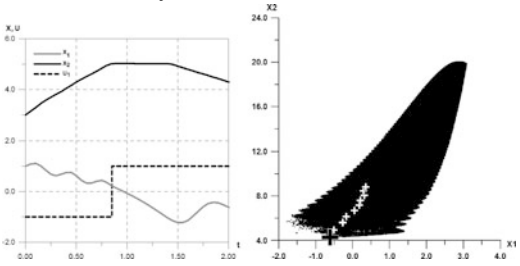
$t \in [0, 2]$

$x_0 = (1, 3)$

$u = \{-1, 1\}$

$I(u) = x_2(t_1) \to \min$

Optimal trajectories and control; attainability set



Problem 19.

$\dot{x}_1 = x_2 + \sin t^2$

$\dot{x}_2 = u + \cos(x_1 x_2)$

$t \in [0, 3]$

$x_0 = (0.3, 0)$

$u = \{-1, 1\}$

$I(u) = -x_1^2(t_1) + x_2^2(t_1) \to \min$

Optimal trajectories and controls; attainability set

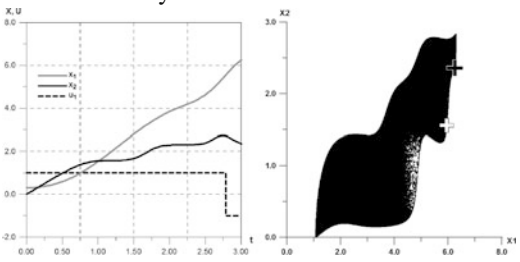


Table 18 Numerical solutions to test problem 18

N	Functional value	Extreme points
1	4.29272	(0.61911, 4.29272)
2	4.97189	(−0.51485, 4.97189)
3	5.62896	(−0.27598, 5.62896)
4	6.17173	(−0.17237, 6.17173)
5	6.72009	(0.04066, 6.72009)
6	7.19763	(0.17311, 7.19763)
7	7.63813	(0.20221, 7.63813)
8	8.06943	(0.31370, 8.06943)
9	8.46982	(0.29069, 8.46982)
10	8.84933	(0.34935, 8.84933)
The number of Cauchy problems: 76,835		

Table 19 Numerical solutions to test problem 19

N	Functional value	Extreme points
1	−33.47855	(6.24783, 2.35729)
2	−32.95061	(5.94775, 1.55729)
The number of Cauchy problems: 155		

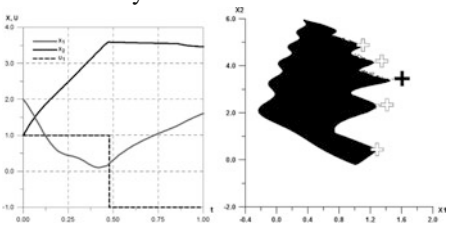
Table 20 Numerical solutions to test problem 20

N	Functional value	Extreme points
1	−1.60657	(1.60657, 3.46381)
2	−1.41458	(1.41458, 2.33631)
3	−1.34505	(1.34505, 4.20458)
4	−1.28311	(1.28311, 0.44614)
5	−1.10476	(1.10476, 4.87465)
The number of Cauchy problems: 11,055		

Problem 20.

$\dot{x}_1 = -6\sin x_1 + x_2 - 3u + 2\cos x_2^2$
$\dot{x}_2 = \cos^2 x_2 + 0.3x_1 + 4u + x_1^2$
$t \in [0,\ 1]$
$x_0 = (2, 1)$
$u = \{-1, 1\}$
$I(u) = -x_1(t_1) \to \min$

Optimal trajectories and controls; attainability set



8 Conclusion

Currently the collection of test problems includes about 100 test cases. The principle of "the best of known solutions" was applied to all tests. The best of the currently known solutions is presented. The collection of problems was described in the

same manner: for each problem its statement was given along with information about the known local extrema, optimal control and trajectory, and the number of Cauchy problems required to obtain the optimal functional value. The attainable set approximations presented can be used for a deeper analysis of the computation results. The authors hope that the interested specialists can carry out their studies of these problems and find better solutions.

Acknowledgements This work is partly supported by Grants N 12-01-00193 and N 10-01-00595 of Russian Foundation for Basic Research.

References

1. Afanasev, V.N., Kolmanovskii, V.B., Nosov, V.R.: Mathematical theory of control system design. Moscow, Visshaya shkola (2003) (in Russian)
2. Ali, M.M., Khompatraporn, C., Zabinsky, Z.B.: A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. J. Glob. Optim. **31**, 635–672 (2005)
3. Batishev, D.I.: The search methods of optimal engineering. Moscow, Soviet radio (1975) (in Russian)
4. Betts, J.T.: Experience with a sparse nonlinear programming algorithm. In: Biegler, L.T., Coleman, T.F., Conn, A.R., Santos, F.N. (eds.) Large Scale Optim. with Appl.: Optim. Des. and Control, **2**, Berlin, Springer (1997)
5. Buckley, A.: A portable package for testing minimization algorithms. Proc. of COAL Conf. on Math. Progr., 5–6 (1982)
6. Crowder, H.P., Dembo, R.S., Mulvey, J.M.: Reporting computational experiments in mathematical programming. Math. Progr. **15**, 316–329 (1978)
7. Dadebo, S., Luus, R.: Optimal control of time-delay systems by dynamics programming. Optim. Control Appl. and Meth. **13**, 29–41 (1992)
8. Dixon, L.C.W, Szego, G.P. (Eds.): Towards global optimization. Amsterdam, North Holland (1978)
9. El-Gindy, T.M., El-Hawary, H.M., Salim, M.S., El-Kady, M.: A Chebyshev approximation for solving optimal control problems. Comput. Math. Applic. **29**(6), 35–45 (1995)
10. El-Kady, M.M., Salim, M.S., El-Sagheer, A.M.: Numerical treatment of multiobjective optimal control problems. Autom. **39**, 47–55 (2003).
11. Evtushenko, Yu.G.: The methods of extremal problems solving and their application for optimization systems. Moscow, Nauka (1982) (in Russian)
12. Floudas, C.A., Pardalos, P.M.: A collection of test problems for constrained global optimization algorithms. Berlin, Springer-Verlag (1990)
13. Gornov, A.Yu.: On a class of algorithms for constructing internal estimates of reachable set. Proc. of Int. Workshop, Pereslavl-Zalessky, Russia (1998) (in Russian)
14. Gornov, A.Yu.: Realization of the random multi-start method for optimal control problems. Proc. of Lyapunov's Symp., Irkutsk, Russia (2003) (in Russian)
15. Gornov, A.Yu.: Computational technologies for solving optimal control problems. Novosibirsk, Nauka (2009) (in Russian)
16. Gornov, A.Yu.: Optimal control problem: computing technologies for finding a global extremum. Proc. of Int. Conf. on Optim., Simul. and Control, Ulaanbaatar, Mongolia (2010)
17. Gornov, A.Yu., Zarodnyuk, T.S.: Method of curvilinear search for global extremum in optimal control problems. Contemp. Technol. Syst. Anal. Simul. **3**, 19–27 (2009) (in Russian)

18. Gornov, A.Yu., Zarodnyuk, T.S.: Method of stochastic coverings for the optimal control problem. Comput. Technol. **2**, 31–42 (2012) (in Russian)
19. Hock, W., Shittkowski, K.: Test examples for nonlinear programming codes. Berlin, Springer-Verlag (1981)
20. Holodniok, N., Klich, A., Kubichek, M., Marek, M.: The analysis methods of nonlinear dynamical model. Moscow, Mir (1991) (in Russian)
21. Jacson, R., Mulvey, J.: A critical review of comparisons of mathematical programming algorithms and software (1953–1977). J. Res. Natl. Bur. Stand. **83**(6), 563–584 (1978)
22. Meyer, C.A., Floudas, C.A., Neumaier, A.: A global optimization with non-factorable constraints. Ind. Eng. Chem. Res. **41**, 6413–6424 (2002)
23. Moiseev, N.N., Ivanilov, Yu.P., Stolyarova, E.M.: The optimization methods. Moscow, Nauka, (1975) (in Russian)
24. More, J.J., Garbow, B.S., Hillstrom, K.E.: Testing unconstrained optimization software. ACM Trans. Math. Soft. **7**, 17–41 (1981)
25. Polyak, B.T.: Introduction to optimization. Moscow, Nauka (1983) (in Russian)
26. Schittkowski, K.: More test examples for nonlinear programming, lecture notes in economics and mathematical systems. Berlin, Springer (1987)
27. Schittkowski, K.: Data fitting and experimental design in dynamical systems with EASY-FIT ModelDesign – user's guide. University of Bayreuth, Germany (2009)
28. Skokov, V.A.: The certain computational experience of solving nonlinear programming problems. Math. Meth. of Solving Econ. Probl. **7**, 51–69 (1997) (in Russian)
29. Srochko, V.A.: Iterative methods of optimal control problem solving. Moscow, Fizmatlit (2000) (in Russian)
30. Strongin, R.G.: The numerical methods of multiextremal optimization. Moscow, Nauka (1978) (in Russian)
31. Teo, K.L.,Wong, K.H.: Nonlinearly constrained optimal control of nonlinear dynamic systems. J. Australian Math. Soc. **33**, 507–530 (1992)
32. Tyatyushkin, A.I.: Numerical methods and software for optimization of controlled systems. Novosibirsk, Nauka (1992) (in Russian)
33. Zarodnyuk, T.S.: The algorithm of numerical solving of multiextremal optimal control problems with box constraints. Comput. technol. (2013) (in Russian)
34. Zarodnuk, T.S., Gornov, A.Yu.: A technology for finding global extremum in a problem of optimal control. Contemp. Technol. Syst. Anal. Simul. Irkutsk. **3**(19), 70–76 (2008) (in Russian)
35. Zhigljavsky, A.A., Zilinskas, A.G.: The methods of global extremum searching. Moscow, Nauka (1991) (in Russian)

A Multimethod Technique for Solving Optimal Control Problem

Alexander I. Tyatyushkin

Abstract A multimethod algorithm for solving optimal control problems is implemented in the form of parallel optimization processes with the choice of the best approximation. The multimethod algorithm based on a sequence of different methods is to provide fast convergence to an optimal solution. Such a technology allows one to take into account some particularities of the problem at all stages of its solving and improve the efficiency of optimal control search.

Key words Optimal control • Multimethod algorithms • Parallel computations • Software packages • Numerical methods

1 Introduction

The technology of finding the numerical solution to the applied optimal control problems is based on universal software which has a well-developed interface and a rich arsenal of optimization methods. Such software allows one to take into account specific features of the problem under consideration by making the use of diverse algorithms of improvement at different stages of iteration process. Application of several numerical methods for solving a single optimization problem was suggested in many publications oriented to the software development [1–3, 6, 7, 11].

Principal difficulty in applying the multimethod algorithm lies in the fact that at each stage of the problem-solving process one has information about efficiency of the method applied at the present moment. To determine the efficiency of any optimization method at some stage of searching for solution to the given problems

A.I. Tyatyushkin (✉)
Institute for System Dynamics and Control Theory SB RAS, 134 Lermontov St., 664033 Irkutsk, Russia
e-mail: tjat@icc.ru

A. Chinchuluun et al. (eds.), *Optimization, Simulation, and Control*, Springer Optimization and Its Applications 76, DOI 10.1007/978-1-4614-5131-0_17,

it is necessary to perform one or several iterations. Therefore, to choose the method which is more appropriate for the given stage of problem solving, the operation of switching from one method to another is usually repeated.

Also, it is necessary to know about switching times. But this information can be easily obtained by tracking the current method measuring parameters characterizing its convergence.

Thus, the principal problem of the multimethod technology is the choice of method which allows one to continue effectively the optimization process from the moment when convergence of current method was impaired.

Modern operational systems provide a solution to the given problem by organizing parallel computational flows for simultaneous computation by several methods. In each flow, one can realize iterative process of one method from a collection of methods. Thus, a single problem can be solved by several methods. With the multiprocessor technology on hand, of course, it is convenient to use individual processor for accomplishing iteration of each method.

After finding the next approximation, each method is considered, for instance, evaluating an increment of the functional. More effective method is taken to continue the optimization. Next, the approximation obtained by this method is transferred to other methods as initial data to perform next iteration. Starting from this approximation, one or several iterations are again performed by all methods. Out of the obtained approximations again we take the one in which the functional has a smaller value.

Continue iterative process until the optimum criterion is met for the obtained approximation. After that we find an approximation solution to the problem under consideration. In this case the solution is found by the multimethod algorithm consisting of a sequence of steps of different methods attached to optimization process to accelerate its convergence. The advantage of the multimethod algorithm in comparison with each method separately lies in its greater adequacy in application to concrete problem. At each stage of searching for solution, the multimethod algorithm makes the use of optimization method which is more suitable in terms of specific features of a given problem (e.g., the ravines of function, specific character, and the structure of constraints).

In the graphic, the decrease in the functional $I(u)$, in iterations of the multimethod algorithm, is shown by the broken line which consists of the graphics of separate methods. Figure 1 shows the multimethod algorithm operation in the case, when two methods $M1$ and $M2$ are used. The graphics given show the decrease in the functional in the iterations of the methods $M1$ and $M2$. The graphic of decrease in the functional in the iterations of the multimethod algorithm is the curve ABC. It is constructed by using two graphics corresponding to the methods $M1$ and $M2$. Namely, the region BC is obtained by parallel translation of the region EL. According to this figure, the zero value of functional is achieved in k_1 iterations of the method $M1$, while the use of the methods $M1$ and $M2$ requires k_2 iterations.

The multimethod algorithm works, up to the $\bar{k}$th iteration, by the method $M1$ (the curve AB) and then by the method $M2$ (the curve BC). The reason is that

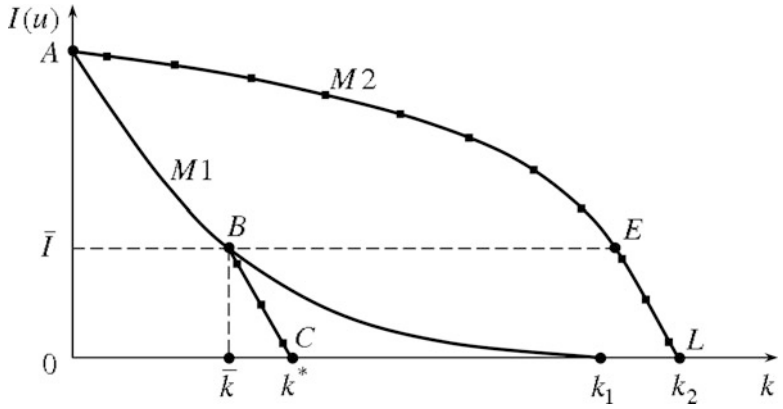


Fig. 1 Graphics of decrease of functional on the iterations of the methods $M1$, $M2$ and multimethod algorithm

beginning with the $\bar{k}$th iteration, the velocity of decrease in the functional during the use of the method $M2$ is higher. As a result, zero value of functional is achieved by the multimethod algorithm using k^* iterations. This is considerably less than in the case where the methods $M1$ and $M2$ are used individually.

2 Parallel Computations in the First-Order Methods

By different criteria for choosing the closest optimization method and also organizing in different ways parallel computations on the method's iterations, several different combinations of algorithms can be obtained to solve a single problem. Moreover, it is possible to construct the multimethod algorithms that do not contain repeated computations in the iterations of different optimization methods. For example, in the methods of gradient type [4, 13], laborious computations of the gradient, requiring an integration of the adjoint system, should be performed only once; then, the obtained gradient should be used in the iterative formulas of all methods. In this case, computational expenditures at one step of multi-flow algorithm are considerably reduced. Moreover, realization of the step by any of the methods is accomplished by using the same approximately obtained values. Then all optimization algorithms are applied as if to the one and the same approximate model. Thus, the criterion of new approximation is defined only by the optimization methods. Otherwise, because of computational errors, the same parameters used to estimate convergence of the methods may have different values which can lead to improper choice of the best method.

Let us consider an optimal control problem provided by the conditions in the form of equalities at the right trajectory end. The controlled process is described by the system

$$\dot{x} = f(x,u,t), \quad x(t_0) = x_0, \quad t \in T = [t_0,t_1], \quad x(t) \in \mathbb{R}^n, \quad u(t) \in \mathbb{R}^r \tag{1}$$

with terminal conditions

$$I_j(u) = \varphi^j(x(t_1)) = 0, \quad j = \overline{1,m} \tag{2}$$

and with phase constraints

$$J_i(u,t) = g^i(x,t) = 0, \quad i = \overline{1,s}, \quad t \in T. \tag{3}$$

The control is constrained through

$$u(t) \in U, \tag{4}$$

where U is a bounded closed set in $\mathbb{R}^r$. Vector functions $f(x,u,t)$ are assumed to be differentiable w.r.t. to x and u and continuous w.r.t. t; $\varphi^j(x)$, $j = \overline{1,m}$ are assumed to be continuously differentiable w.r.t. x functions.

It is required to find the control among controls fulfilled by (3), such that provides the validity of conditions (2), for controlled process (1) and, on the other hand, provides the minimum of the functional

$$I_0(u) = \varphi^0(x(t_1)), \tag{5}$$

where $\varphi^0(x)$ is continuously differentiable function.

The gradients of functionals $I_j(u)$, $j = \overline{0,m}$ in terms of the functions $H^j(\psi_j,x,u,t) = \psi_j'(t)f(x,u,t)$ and adjoint system

$$\dot{\psi}_j = -f_x(x,u,t)'\psi_j(t), \quad \psi_j(t_1) = -\psi_x^j(x(t_1))$$

are given by the formula

$$\nabla I_j(u) = -H_u^j(\psi_j,x,u,t), \quad j = \overline{1,m}. \tag{6}$$

For each $t \in T$ one can calculate in much the same way the gradients $J_j(u,t)$, $j = \overline{1,s}$:

$$\nabla J_j(u,t) = -\overline{H}_u^j(\Phi_j,x,u,t,\tau), \quad t_0 \le \tau \le t \le t_1, \tag{7}$$

where $\overline{H}_u^j(\Phi_j,x,u,t,\tau) = \Phi_j'(t,\tau)f(x,u,\tau)$, $\Phi_j(t,\tau)$, $j = \overline{1,s}$ are the solutions of the conjugate system

$$\frac{\partial \Phi_j(t,\tau)}{\partial \tau} = -\frac{\partial f(x,u,\tau)}{\partial x}\Phi_j(t,\tau), \quad \tau \in T,$$

with the boundary conditions $\Phi_j(t,t) = -\dfrac{\partial g^j(x(t))}{\partial x}$, $j = \overline{1,s}$.

2.1 Application of Gradient Methods

Gradient procedure of the minimization of the functional (4) without taking into account the constraints (2) and (3) is given through relation

$$u^{k+1} = u^k - \alpha_k \nabla I_0(u^k),$$

where α_k are chosen, for example, from condition of fastest decrease of the functional $I_0(u)$. The solution of the problem with terminal conditions (2) without constraints (3) can be obtained by parallel application of the linearization and penalization methods given, for example, in [7]. When using penalization method, penalty functional which consists of the functions (2) and (4) is minimized with the help of gradient procedure. Making use of the multimethod technology, one can also simplify each iteration of the linearization method in such a way that its working time will be close to the one of the penalization method. Then the algorithm will be as follows:

1. For given $u^k(t), t \in T$, system (1) is integrated; in the integration points, the phase coordinates of trajectory $x^k(t)$ are memorized.
2. $m+1$ flows are organized, for parallel integration of adjoint system provided by different initial conditions $\psi_j(t_1) = -\psi_x^j(x(t_1))$, $j = \overline{0,m}$. In the process of integration, the solutions $\psi_j(t)$ are used to construct a linear system of algebraic equations

$$\sum_{i=1}^{m} \left(\int_{t_0}^{t_1} H_u^{j\prime} H_u^i \, \mathrm{d}t \right) \lambda_i = I_j(u^k) - \int_{t_0}^{t_1} H_u^{j\prime} H_u^0 \, \mathrm{d}t, \quad j = \overline{1,m}.$$

3. After solving this system, the values of variables λ_i, $i = \overline{1,m}$ are found.
4. A new approximation of the control

$$u^{k+1} = u^k + \alpha_k \delta u, \qquad \delta u = H_u^0 + \sum_{i=1}^{m} \lambda_i H_u^i,$$

is constructed, where the parameters α_k satisfy inequality

$$I_0(u^k + \alpha_k \delta u) + \beta I_{j_0}(u^k + \alpha_k \delta u) \le I_0(u^k) + \beta I_{j_0}(u^k) - \varepsilon \int_{t_0}^{t_1} \delta u' \delta u \, \mathrm{d}t,$$

$$0 < \varepsilon < 1, \quad j_0 = \arg\max_{1 \le j \le m} \left| I_j(u^k) \right|, \quad \beta = \sum_{i=1}^{m} |\lambda_i| .$$

From this algorithm, as particular case (for $m = 0$), we obtain the ordinary gradient method.

2.2 The Methods for Solving Problem with Constraints on Control

Let us focus on the algorithms intended for solving the problems provided by constraints on control, but with free right end. Suppose that for some $u^k(t) \in U$, $t \in T$, one finds any solution to the system (1) $x^k(t)$, $t \in T$. Setting in (5) $j = 0$, integrate adjoint system from $t = t_1$ to $t = t_0$ when $u = u^k(t)$, $x = x^k(t)$. Calculate on its solution $\psi^k = \psi^0(t)$ the control using the maximum principle:

$$\bar{u}^k(t) = \arg\max_{u \in U} H(\psi^k, x^k, u, t), \quad t \in T,$$

and find the value of scalar function

$$w_k(\bar{u}(t), t) = H(\psi^k, x^k, \bar{u}, t) - H(\psi^k, x^k, u^k, t), \quad t \in T.$$

Let $t = \tau_k$ the maximum point of this function being on T. Then, the necessary optimum condition of the control u^k will be

$$w_k(\bar{u}^k(\tau_k), \tau_k) = 0.$$

In the case when for given u^k and obtained x^k, ψ^k, $\bar{u}^k$, the maximum principle

$$w_k(\bar{u}^k(\tau_k), \tau_k) > 0$$

does not hold; iteration of the method [12] can be made to improve u^k.

Denote the point set, where maximum principle is broken by

$$T_\varepsilon = \left\{ t \in T : \; w_k(\bar{u}^k(t), t) \geq \varepsilon w_k(\bar{u}^k(\tau_k), \tau_k) \right\}, \quad \varepsilon \in [0, 1].$$

Observe that at $\varepsilon = 0$, we have $T_\varepsilon^k = T$, while at $\varepsilon = 1$, the set T_ε^k consists of the maximum points of the function $w_k(u(t), t)$.

By varying ε, we can find such value for which the control

$$u_\varepsilon^k = \begin{cases} \bar{u}^k(t), & t \in T_\varepsilon, \\ u^k(t), \, t \in T \setminus T_\varepsilon \end{cases} \tag{8}$$

provides the least value of the objective functional $I_0(u)$, i.e.,

$$\varepsilon_k = \arg\min_{\varepsilon \in [0,1]} I_0(u_\varepsilon^k).$$

When searching for ε_k, several flows can be used for simultaneous integration of system (1) with controls (8), corresponding to different values of $\varepsilon \in [0, 1]$. In addition, at $t = t_1$, we have different phase space points $x_\varepsilon^k(t_1)$ and pertinent values $I_0(u_\varepsilon^k) = \varphi(x_\varepsilon^k(t_1))$. After the smallest value of the functional $I_0(u_\varepsilon^k)$ is chosen,

we verify the inequality $I_0(u_\varepsilon^k) < I_0(u^k)$, and if it holds, we assume $u^{k+1} = u_\varepsilon^k$. Otherwise, the subdivision of ε can be continued, and the values of functional for the following values can be found.

By virtue of structure of the controls generated by the iteration formula (8), the relaxation of an algorithm can be impaired even before the control satisfying the maximum principle is obtained. Therefore, to continue optimization process, it is necessary to apply another algorithm, in iteration of which the controls are constructed not only with boundary points but also with interior ones w.r.t. the set U as well. For example, the convergence can be restored by constructing a convex combination of two controls

$$u^{k+1}(t) = u^k(t) + \alpha\left[\bar{u}^k(t) - u^k(t)\right], \quad \alpha \in [0,1]. \tag{9}$$

The calculations by the formulas (8) and (9) can be made simultaneously by choosing from the obtained approximations such u^{k+1} to which the smallest value of the functional corresponds. In the case, where the functional values are compared within several iterations, the values of increase in the functional, obtained in the neighboring iterations of each method, should be used as a criterion to compare the efficiency of the methods (8) and (9).

In practice, it is established that the application of the variations of two types, namely, "horizontal" (8) and "vertical" (9), allows us to avoid the effect of "control sticking to the boundaries" which is inherent in the algorithms based on the maximum principle [5, 12].

In the case in the iteration equation (9), the control $\bar{u}^k(t)$ is derived from the linearized maximum principle

$$\bar{u}^k(t) = \arg\max_{u \in U} H_u(\psi^k, x^k, u, t)' u(t), \quad t \in T; \tag{10}$$

we obtain the iterations of the conditional gradient method. It is evident that for the systems which are linear in control, the control function (10) coincides with that deduced from the maximum principle. Another algorithm of control improvement can be obtained by substituting, in the iteration formula (8), the interval T_ε for the following one:

$$T_\varepsilon^k = \left[\tau_k - \varepsilon(\tau_k - t_0^k),\ \tau_k + \varepsilon(t_1^k - \tau_k)\right], \quad \varepsilon \in [0,1], \tag{11}$$

where t_0^k, t_1^k are the nearest left and right discontinuity points of the function $w(\bar{u}^k(t), t)$.

In the process of finding the value of parameter ε_k providing convergence of the algorithm, we can apply the above procedure along with parallel computations. This way of constructing the interval provides blowing down of its ends towards the point τ_k, in the case, if the function $w(\bar{u}^k(t), t)$ is constant within some neighborhood of the point τ_k, thus maintaining the convergence of the algorithm [12].

2.3 Linearization Method for Solving Problems with Phase Constraints

Let $u_k(t)$ be a current approximation of the control, and let $x_k(t)$ be the phase trajectory, corresponding to $u_k(t)$, $t \in T$. Using the gradients (6) and (7) we linearize the conditions (2) and (3) in the neighborhood of u_k:

$$I_i^L(u^k,u) = I_i(u^k) + \int_{T_0}^{T_1} \nabla I_i(u^k,t)'\big(u(t) - u^k(t)\big)\mathrm{d}t = 0, \quad i = \overline{1,m}, \tag{12}$$

$$J_j^L(u^k,u,\tau) = J_j(u^k,\tau) + \int_{t_0}^{\tau} \nabla J_j(u^k,t)'\big(u(t) - u^k(t)\big)\mathrm{d}t = 0, \tag{13}$$

$$j = \overline{1,s}, \quad \tau \in T.$$

Construct a modified Lagrange function for the problem (1)–(5) in the form:

$$\begin{aligned} L(u,u^k,\lambda^k,\mu^k) = {} & I_0(u) - \lambda^{k\prime}\big(I(u) - I^L(u^k,u)\big) \\ & - \int_{t_0}^{t_1} \mu^k(t)'\big(J(u,t) - J^L(u^k,u,t)\big)\mathrm{d}t \\ & + \frac{\rho}{2}\big(I(u) - I^L(u^k,u)\big)'\big(I(u) - I^L(u^k,u)\big) \\ & + \frac{\rho}{2}\int_{t_0}^{t_1} \big(J(u,t) - J^L(u^k,u,t)\big)'\big(J(u,t) - J^L(u^k,u,t)\big)\mathrm{d}t, \end{aligned} \tag{14}$$

where I, I^L — m-vectors; J, J^L — s-vectors; λ^k, μ^k are m- and s-dimensional Lagrange factors; $\rho \geq 0$ is a penalty coefficient.

In the $k+1-st$ iteration of the considered method we solve the minimization problem of functional (14) on the solutions of the system (1) with the linear constraints (12), (13), and (4). By solving the problem, we determine new values of the Lagrange factors λ_i^{k+1}, $i = \overline{1,m}$, $\mu_i^{k+1}(t)$, $i = \overline{1,s}$, $t \in T$.

After determining u^{k+1}, λ^{k+1}, and μ^{k+1}, we again linearize constraints (2), and (3) in the neighborhood of u^{k+1}, construct the functional $L(u,u^{k+1},\lambda^{k+1},\mu^{k+1},\rho)$, and repeat the iteration.

Thus, the algorithm for solving the formulated problem consists of the following main operations:

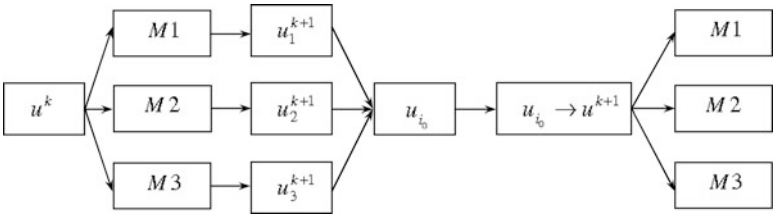


Fig. 2 The scheme of realization of $(k+1)$-th iteration by multimethod algorithm using three methods $M1$, $M2$, and $M3$

1. Linearizing the constraints and solving the auxiliary problem (1), (4), and (12)–(14). Calculating the Jacobian of the linear system (13) is computationally very expensive.
2. Verifying the optimal conditions for the solution obtained on the kth iteration.

When solving the problems with linear constraints the application of this algorithm is greatly simplified because it is not necessary to calculate the constrained Jacobian.

2.4 The Block Scheme of Multimethod Algorithm Operation

Summarizing the above said we see that by applying various iteration procedures and making use of different rules to construct the sets of varying controls, we obtain the collection of algorithm, each working effectively enough only in a certain situation. Thus, in the process of finding the optimal control, it is necessary to include several algorithms.

By organizing parallel computations to realize some collection of algorithms and applying the selection procedure to take the best approximation after simultaneous iterations by all methods, we are able to find effectively optimal control by the multimethod algorithm.

Figure 2 demonstrates how the multimethod algorithm works in the case when three methods are used. The block of selection of the best approximation finds u_{i_0} from a largest value of increment of the functional obtained in the $(k+1)$-th iteration

$$u_{i_0} = \underset{i\in(1,2,3)}{\arg\max} \left(I(u^k) - I(u_i^{k+1})\right).$$

This approximation is passed to all methods $u_i^{k+1} = u_{i_0}$, $i = 1, 2, 3$, to perform the next iteration.

It should be noted that another multimethod algorithm can be generated from the collection of methods for solving another problem. The algorithm can be more adequate because of taking into account the specific features of this problem.

3 Implementation of Multimethod Algorithm

3.1 Solving the Adjoint Problem

The most labor-consuming operation performed at each step of all first-order algorithms is numerical integration of original and adjoint system of differential equations provided by some control. The solving of adjoint system in the iterations is used to find either the value of Hamiltonian or both to perform the calculation of gradients of functionals. Thus, the numerical integration of this system is accomplished at each step of any of the first-order methods. Since the multimethod technology enables the steps to be made simultaneously by all methods, the solution of adjoint system, which is obtained by single integration, is used in all iteration formulas simultaneously.

3.2 Computation of the Method Step

Realization of each step of the first-order method needs to find the value of method's step α or ε, for which new obtained approximation provides the smallest value of objective functional. The search for such a value of α requires multiple integration of initial system (1). By constructing control u^α by given iteration formula at different values of α, and integrating initial system, at $u = u^\alpha$, we obtain various trajectories $x^\alpha(t)$, $t \in T$ and find pertinent values of the functional $I(u^\alpha)$. By applying the one-dimensional search method it is possible by several such recalculations to find the approximate value of $\alpha = \alpha_k$such that the minimum of $I(u^\alpha)$ w.r.t. parameter α is provided. If, in the process, the inequality $I(u^{\alpha_k}) < I(u^k)$ holds, the control u^{α_k} is taken as new approximation u^{k+1}. Otherwise, iteration process, using this method, comes to an end.

To recover the method convergence it is necessary to correct calculations, namely, diminish the integration step, increase the accuracy of one-dimensional search, and so on. The multimethod algorithm can be used to continue the process of control improvement by switching to another method, while the condition for termination of its work is impossibility to guarantee relaxation by none of the methods entering in the given collection.

Another scheme of search can also be used to find the value of α_k while using the multimethod algorithm. Its essence consists in the fact that in the process of subdivision of α, for example, by cutting in two, with each its value, it accomplishes test step by all methods. Since, in many algorithms, for finding α_k, it accomplishes subsequent subdivision in two until the inequality $I(u^{\alpha_k}) < I(u^k)$ holds, then, for any fixed α, the validity of this inequality is checked simultaneously for all methods. In addition, with given value of α, in correspondence with iteration formula of each method, it is constructed the control u^α integrated the system (1). Its solution is used to calculate the functional $I(u^\alpha)$. If, in the process, for some value of parameter α,

for some of the methods, the required inequality holds, then u^{α_k}, obtained by this method, should be taken as a new approximation to be passed to all methods for continuation of iteration process.

3.3 Choice of Optimization Methods

For parameters to be used to estimate the efficiency of iteration process, one can take, for instance, the velocity of reduction of the residuals in optimization conditions, the value of increment of the objective functional, or the extent of violation of some important, for example, in the physical sense, constraints. The accuracy of calculations of these parameters provides the proper choice of the method and opportune transition to another optimization method.

3.4 The Methods of Approximation of Control

The principal error in modeling the control problem arising in the process of numerical solving is due to discrete approximation of controlled dynamic system and tabular representation of control function.

The methods of numerical integration allow for discretization of the system with given accuracy but provided that control functions are continuous and their values can be determined for arbitrary $t \in T$. However, in practice, piecewise constant approximation of control functions is often applied, and its values are defined only in given sites of the temporal network. These values are changed in iteration process, depending on optimization method to decrease minimized functional or to reduce the residuals for given terminal conditions. The values of control functions between the points, which are necessary, for example, to apply the numerical methods of Runge–Kutta type, as usual are taken to be equal either the value in the nearest left point or both are defined by linear interpolation by the values in left and right points. In this process, the error of numerical integration can be considerably extended. Then, on the obtained numerical solution of the system, the values of parameters, which are used to choose the method, could be incorrect.

To diminish the errors of calculation of trajectory, one can condense the temporal network in order to increase the number of desired values of control which implies to solve optimization problem of a big size. In the case if required control is smooth enough, then, with some number of points and with the help of interpolation formulas, the admissible accuracy of its approximation can be provided. However, in general case, if control functions are discontinuous, for example, of relay type, this approach may turn out to be inefficient, since, in this case, the condensation of network affects the approximation accuracy only in the neighborhood of the switching points. Therefore, in the process of solving the bang–bang optimal control problems, it is necessary to use the procedure of switching times correction to

provide the approximation for control functions with prescribed accuracy. In the absence of this procedure, the control is found in the form of the array of numbers defined on temporal network. Such a control may differ in significantly from the optimal one both by the value of objective functional and by the accuracy of optimal conditions fulfillment.

3.5 The Estimation of Accuracy of a Chosen Method

The application of multimethod algorithm is correct only if all methods use the same approximation of the control and provide the same accuracy of integration of the systems. In this case, all optimization methods are used to solve the same finite dimensional problem obtained by discretization, while the values of parameters used to choose the method give a more correct estimate of the algorithm efficiency, providing thus a correct choice of the method to continue iteration process. As a result, the approximate solution is obtained in a lesser number of iterations, in comparison to individual use of each method from a given collection. This is because at each stage of solving the considered problem, a more effective algorithm is applied to be more adequate in this situation.

4 The Numerical Experiments

Let us present the results of the numerical experiments to demonstrate application of the multimethod technology. Two examples are considered: a test problem of the Rosenbrock function minimization and the problem of optimal control of the rocket flight.

4.1 Example 1

It is required to find the minimum of Rosenbrock function $f(x_1,x_2) = 100(x_2 - x_1^2)^2 + (1-x_1)^2$ with constraint $(x_1-2)^2 + (x_2-2)^2 \leq 1$ and initial approximation $x^0 = (2,1)$.

It is known that the absolute minimum of this function is achieved at the point $(1,1)$. It is equal to 0. In the problem under consideration, the point $(1,1)$ is not feasible. Therefore the zero value of objective functional is not attained.

This problem was solved numerically by the method of conditional gradient, by the method of gradient projection, and also by the multimethod algorithm which contains these two methods. The smallest value of the function equal to 0.0358, for prescribed accuracy $\varepsilon = 10^{-4}$, was attained using 240, 341, and 170 iterations, respectively. Minimum point is $x^* = (1.189, 1.415)$.

4.2 Example 2

In this case, controlled process is described by the system

$$\dot{x}_1 = -cpx_1^2[1.174 - 0.9\cos u_1] - g\sin x_2/(1+x_3)^2,$$
$$\dot{x}_2 = 0.6cpx_1^2\sin u_1 + x_1\cos x_2/[R(1+x_3)] - g\cos x_2/[x_1(1+x_3)^2],$$
$$\dot{x}_3 = x_1\sin x_2/R,$$
$$\dot{x}_4 = x_1\cos x_2/[1+x_3],$$

where $\rho = 0.002704\exp(-4.26Rx_3)$, $R = 209$, $g = 0.00032172$, and $c = 26600$.

The initial data $x(0) = (0.36, -0.141372, 0.019139, 0.0)$ and terminal conditions $x(t_k) = (0.27, 0.0, 0.011962,$ *unbounded*)are given, while the parameter t_k is not fixed. It is required to find such control $u(t)$, $t \in [0, t_k]$ and such smallest value of parameter t_k that the fulfillment of terminal conditions is guaranteed, while the functional

$$I(u) = \int_0^{t_k} x_1[\exp(-4.26Rx_3)]^{1/3}\,dt$$

should attain the smallest value.

This problem was solved by the above linearization method and also by the projected Lagrangian method, in iteration of which the nonlinear Lagrangian with linearizable constraints is minimized [11]. Approximate solution (the accuracy w.r.t. boundary conditions is 10^{-3}) was found in 282 and 215 iterations, respectively. The solution with the same accuracy, that was obtained with the multimethod algorithm including these two methods, was found in 164 iterations. The value of parameter t_k equals 72.412, while the functional $I(u)$ attains the value equal to 24.59.

5 Conclusion

Thus, we can conclude that for each problem under consideration there exists an appropriate sequence of steps based on different methods which provides more effective search for the optimal control. In the multimethod algorithms the construction of such a sequence is accomplished automatically according to some given criterion estimating the efficiency of optimization process at each stage of the problem solving. The use of the technology described above is based on the application software, for example, [5–11, 14], which includes the methods of first and second order for solving the optimal control problems with constraints of different types.

References

1. Evtushenko, Yu.G.: Methods of solving of extremal problems and its application. Nauka, Moscow (1982).
2. Gurman, V.I., Dmitri'ev, M.G., Osipov, G.S.: Intellectual multimethod's technology for solving and analysis of control problems: Preprint of Institute of programmed systems of RAS. Pereslavl–Zallesskii (1996).
3. Ling, L., Xue G.: Optimization of Molecular Similarity Index with Applications to Biomolecules. J. Glob. Optim. **14**(3), 299–312 (1999)
4. Lyubushin, A.A., Chernous'ko, F.L.: The method of successive approximations for computation of optimal control. Izv. AN SSSR. Tehn. kibernetika. 2, 141–159 (1983).
5. Morzhin, O.V., Tyatyushkin, A.I.: An algorithm of the section method and program tools for reachable sets approximating. J. of Computer and Systems Sciences International. **47**(1), 1–7 (2008).
6. Tyatyushkin, A.I.: Package KONUS for optimization of continuous controlled systems. The packages of applied softwares: The experience of using. Nauka, Moscow (1989).
7. Tyatyushkin, A.I.: Numerical methods and software for optimization of controlled systems. Nauka, Novosibirsk (1992).
8. Tyatyushkin, A.I.: Numerical methods for optimization of controlled systems. J. Stability and control: Theory and Appl. **3**(2), 150–174 (2000)
9. Tyatyushkin, A.I.: Parallel computations in Optimal control problems. Siberian J. of Number.Mathematics (Sib.Branch of Russ. Acad. of Sci). **3**(2), 181–190, Novosibirsk (2000).
10. Tyatyushkin, A.I.: Many-Method Technique of Optimization of Control Systems. Nauka, Novosibirsk (2006).
11. Tyatyushkin, A.I., Zholudev, A.I., Erinchek, N.M.: The program system for solving optimal control problems with phase constraints. Intern. J. of Software Engineering and Knowledge Engineering. **3**(4), 487–497 (1993).
12. Vasil'ev, O.V., Tyatyushkin, A.I.: On some method of solving of optimal control problems based on maximum principle. J. vychisl. matem. i mat. fiziki. **21**(6), 1376–1384 (1981).
13. Yuan, G.: Modified nonlinear conjugate gradient methods with sufficient descent property for large-scale optimization problems. Optim. Lett. **3**(1), 11–21 (2009).
14. Zholudev, A.I., Tyatyushkin, A.I., Erinchek, N.M.: Numerical optimization methods of controlled systems. Izv. AN SSSR. Tehn. kibernetika. 4, 14–31 (1989).

Tunneling Algorithm for Solving Nonconvex Optimal Control Problems

Alexander Yurievich Gornov and Tatiana Sergeevna Zarodnyuk

Abstract This chapter considers a new method of search for the global extremum in a nonlinear nonconvex optimal control problem. The method employs a curvilinear search technique to implement the tunneling phase of the algorithm. Local search in the minimization phase is carried out with the standard algorithm that combines the methods of conjugate and reduced gradients.

The software implementation of the suggested tunneling algorithm was tested on a collection of nonconvex optimal control problems and demonstrated efficiency of the this approach.

Key words Optimal control • Global optimization • Tunneling methods • Curvilinear search

1 Introduction

Development of approaches to the study of optimal control problems (OCP) was considered by many researchers [2, 6, 12, 15, 16, 20, 25]. Previously genetic algorithms were suggested to solve optimization problems of controllable dynamic systems [1, 3, 9, 14, 18, 19, 22, 23, 26]. However, in recent years, many specialists have tried to construct more capable algorithms [4, 11, 13, 17, 21]. Most publications devoted to this problem concern the transfer of ideas from finite-dimensional optimization to the area of optimal control. However, in many cases, this leads to the algorithms that require large computations to achieve the final results. The approach suggested by us is based on the methods proposed in the theory ofoptimal control

A.Y. Gornov • T.S. Zarodnyuk (✉)
Institute for System Dynamics and Control Theory SB RAS, 134 Lermontov St., 664033 Irkutsk, Russia
e-mail: gornov@icc.ru; tz@icc.ru

A. Chinchuluun et al. (eds.), *Optimization, Simulation, and Control*, Springer Optimization and Its Applications 76, DOI 10.1007/978-1-4614-5131-0_18,

and differential inclusions. From our viewpoint it provides much faster solution of multiextremal problems of optimal control with nonlinear systems of differential equations and nonconvex functionals.

The idea of tunneling algorithms is well known in the global optimization theory [29]. The tunneling algorithms were first suggested in the finite-dimensional optimization in the works by A. V. Vilkov, N. P. Zhidkov and B. M. Shchedrin, and A. V. Levy and A. Montalvo were based on the use of the tunneling function $T(x)$ for finding the point from the neighborhood of the next local extremum of function $f(x)$ [10, 24]: $T(x) = (f(x) - f(x)^*)/\|x - x^*\|^\alpha$, here x^*—the best iteration value, $\alpha > 0$—a parameter. The above studies show that in the case of one-dimensional continuously differentiable function $f(x)$ with a finite number of local extrema, the algorithm with tunneling function $T(x)$ converges to the point of global minimum. The methods based on the use of other techniques of escaping from the found extrema were unveiled in the studies by J. Barhen, J. W. Burdick, B. C. Certin, R. P. Ge, Y. F. Qin, Y. Yao, and others.

Tunneling algorithms rest on various methods of escaping from the local extrema. Once the next local extremum (the local phase of the algorithm) is found, we search for a point at which the quality criterion value is lower than the known best iteration value (the tunneling phase of the algorithm). The found point is chosen to be an initial one for the next local descent. To solve nonconvex optimization problems of dynamic systems we previously suggested and studied the method of curvilinear search [8]. It is based on the idea of Chentsov [5] about the possibility of using the property of connection of the controllable system attainability set for construction of nonlocal numerical methods of functional minimization. The paper addresses a new method of search for the global extremum in the nonlinear nonconvex OCP.

2 Formulation of a Nonconvex Optimal Control Problem

The main subject of our study is OCPs with parallelepiped constraints on control. The system describing the controllable dynamic process may be nonlinear, and, hence, the objective functional may turn out to be nonconvex, which results in appearance of local extrema in OCP. Control functions in the considered problems are smooth and piecewise continuous functions.

The standard statement of the problem is as follows:

$$\dot{x} = f(x,u,t),\ x(t_0) = x^0,\ t \in T = [t_0,\ t_1], \tag{1}$$

$$U = \{u(t) \in R^r : \underline{u} \le u(t) \le \overline{u}\}, \tag{2}$$

$$I(u) = \phi(x(t_1)) \to \min. \tag{3}$$

The OCP consists in the search of control $u^*(t)$ that meets the parallelepiped constraints (2) and allows one to obtain the minimum value of the terminal

functional (3) which depends on the system trajectory (1) at the finite moment of time t_1. The vector function $f(x,u,t)$ and the scalar function $\phi(x)$ are assumed to be continuously differentiable for all arguments except t.

By the multiextremal OCP we understand the problems with feasible controls $u^1 = u^1(t)$ and $u^2 = u^2(t)$ from the set U (2), which can upset the convexity condition of the aim functional $I(u)$: $I(u^1 + \beta(u^2 - u^1)) > I(u^1) + \beta(I(u^2) - I(u^1))$, $\beta \in [0,1]$ [7].

3 Description of Tunneling Algorithm

Tunneling algorithms can be subdivided into two phases: the minimization phase and the tunneling phase [29]. In the proposed approach the first phase implies the application of a standard combination of the method of conjugate gradients and the method of reduced gradient. The second phase is based on the curvilinear search method, which represents a simplest scheme of sequential variations in the control space. This scheme presupposes a combination of the best control in the iteration and the auxiliary control with projection of the control variations onto the feasible parallelepiped. The variations of control are projected onto the terminal phase space in the form of curved lines, along which the smallest value of functional is sought in each iteration. The methods of generating stochastic auxiliary controls are described in Sect. 4.

The first phase deals with the search of local extremum I_{loc}, where $I_{\text{loc}} = \underset{\{u(t):\, u(t)\in U\}}{locmin}\ I(u)$. The initial control $u^0 \in U$ is constructed using the algorithms intended for generation of controls in the form of random relay, piecewise-linear,

The tunneling algorithm

Initialization: Choose $u^0 \in U$

Iteration k ($k \geq 1$)**:**

1. The minimization phase
Find $I_{\text{loc}} = \underset{\{u(t):\, u(t)\in U\}}{locmin}\ I(u)$

2. The tunneling phase
If find $I(u):\ I(u) - I_{\text{loc}} < \varepsilon$
then go to the minimization phase
else $I^* = I_{\text{loc}}$

and spline functions. The second phase suggests finding $I(u)$ such that $I(u) - I_{\text{loc}} < \varepsilon$, where ε determines accuracy of search for the minimum functional value. The found value of $I(u)$ is used in the subsequent minimization phase as initial value for starting the search for the local extremum. If a better value is not found in the tunneling phase we suppose that $I^* = I(u^*) = I_{\text{loc}}$, where I^* is the minimum value of terminal functional in the nonconvex OCP which is obtained with control $u^* = u^*(t)$.

4 The Curvilinear Search Method (The Tunneling Phase)

The modifications of the curvilinear search method are made on the basis of three different methods designed to generate auxiliary controls and variants of constructing control variations. One auxiliary control of the kind $\overline{u}^1 = \overline{u}^1(t)$ makes it possible to obtain linear variation of control $\tilde{u}^1(\alpha) = \alpha(\overline{u}^1 - u_{\text{best}}) + u_{\text{best}}$, $\alpha \in [0, 1]$; two auxiliary controls $\overline{u}^1 = \overline{u}^1(t)$ and $\overline{u}^2 = \overline{u}^2(t)$ make it possible to obtain quadratic variation $\tilde{u}^2(\alpha) = \alpha^2((\overline{u}^1 + \overline{u}^2)/2 - u_{\text{best}}) + u_{\text{best}} + \alpha(\overline{u}^2 - \overline{u}^1)/2 + u_{\text{best}}$, $\alpha \in [-1, 1]$. Three controls provide cubic variation of control:

$$\tilde{u}^3(\alpha) = \alpha^3 \left(\frac{-\overline{u}^1 - 3\overline{u}^2 + \overline{u}^3}{6} + 0.5u_{\text{best}} \right) + \alpha^2 \left(\frac{\overline{u}^1 + \overline{u}^2}{2} - u_{\text{best}} \right) + \alpha \left(\frac{-2\overline{u}^1 + 6\overline{u}^2 - \overline{u}^3}{6} - 0.5u_{\text{best}} \right) + u_{\text{best}}, \ \alpha \in [-1, 2]. \tag{4}$$

Variations of control $\tilde{u}^i(\alpha, t)$, $i = \overline{1,3}$ coincide with auxiliary control actions $\overline{u}^j$, $j = \overline{1,3}$ for certain values of parameter α (e.g., $\tilde{u}^2(-1) = \overline{u}^1$, $\tilde{u}^2(1) = \overline{u}^2$); at the same time $\tilde{u}^1(0) = \tilde{u}^2(0) = \tilde{u}^3(0) = u_{\text{best}}$. The best control u_{best} is the control providing the minimum functional value among all known controls in the current iteration of the algorithm. Different variants of the curvilinear search algorithm are implemented for each case of constructing control variation. The first variant is based on linear combination of the best and auxiliary controls (variant 1). The second variant uses quadratic variations of control (variant 2). The cubic combination of the best and auxiliary controls is applied in the third variant of the curvilinear search algorithm (variant 3). We will present the algorithm variant based on the use of three auxiliary control actions for construction of cubic variations of control.

The Curvilinear Search Algorithm (Variant 3)

1. Choose the initial control $u^0(t) \in U$, $t \in T = [t_0, t_1]$.
2. Specify algorithmic parameters:
   N_C is the number of iterations in the curvilinear search algorithm.
   N_P is a starting number of points for one-dimensional search.
3. Calculate the functional value $I(u^0)$ chosen at the current step as the best value $I_{\text{best}} = I(u^0)$, $u_{\text{best}}(t) = u^0(t)$. In the kth iteration ($k \geq 0$).

4. For all $i = \overline{1,3}$,
   a. Stochastic auxiliary controls $\overline{u}^i = \overline{u}^i(t) \in U$, $t \in T = [t_0, t_1]$ are generated.
   b. If $I(\overline{u}^i) < I_{\text{best}}$ than $I_{\text{best}} = I(\overline{u}^i)$ and $u_{\text{best}}(t) = \overline{u}^i(t)$, go to step 9.
5. The control $\tilde{u}^3(\alpha, t)$ is formed (4), such that $\tilde{u}^3(0) = u_{\text{best}}$, $\tilde{u}^3(-1) = \overline{u}^1$, $\tilde{u}^3(1) = \overline{u}^2$, $\tilde{u}^3(2) = \overline{u}^3$.
6. $\tilde{u}^3(\alpha)$ is projected onto the feasible region:
   a. If $\tilde{u}^3(\alpha) < \underline{u}$ than $\tilde{u}^3(\alpha) = \underline{u}$, $t \in T = [t_0, t_1]$.
   b. If $\tilde{u}^3(\alpha) > \overline{u}$ than $\tilde{u}^3(\alpha) = \overline{u}$, $t \in T = [t_0, t_1]$.
7. Find $\min\limits_{\alpha \in [-1,2]} I(\tilde{u}^3(\alpha))$.
8. If $I(\tilde{u}^3(\alpha^*)) < I_{\text{best}}$ than $I_{\text{best}} = I(\tilde{u}^3(\alpha^*))$ and $u_{\text{best}} = \tilde{u}^k(\alpha^*)$.
9. Store I_{best} and $u_{\text{best}}(t)$.

   The iteration ends.

5 Methods for Generation of Starting Points

The efficiency of algorithms designed to find the minimum in the nonconvex OCP depends on the extent to which the starting controls cover the feasible set. The algorithms intended for generation of starting points for the methods of solving multiextremal OCP should meet the following criteria: the generated controls should be random and feasible (lie within parallelepiped constraints) and belong to the class of piecewise continuous functions and form a dense set in the limit [7].

The algorithms that make it possible to generate controls in the form of relay functions with a fixed or random number of switching points, spline functions, and piecewise-linear and tabulated functions have been investigated and implemented (Fig. 1).

The curvilinear search algorithm was constructed on the basis of the first three methods of control generation. Generation of control actions in the tabulated form (Fig. 1d), when at each node of the discretization grid a random value is chosen from a feasible interval, is considered to be inefficient and, therefore, is not included in the selected set of auxiliary algorithms.

6 Numerical Examples

Several problems from the test collection of nonconvex OCP [27] were used to test the proposed algorithmic scheme. The values of global and local extrema, as well as the points at which they are attained, are shown in Tables 1–3 for test problems 1–3. The optimal controls, trajectories, and attainability sets of controllable dynamic systems are shown in Figs. 2–4.

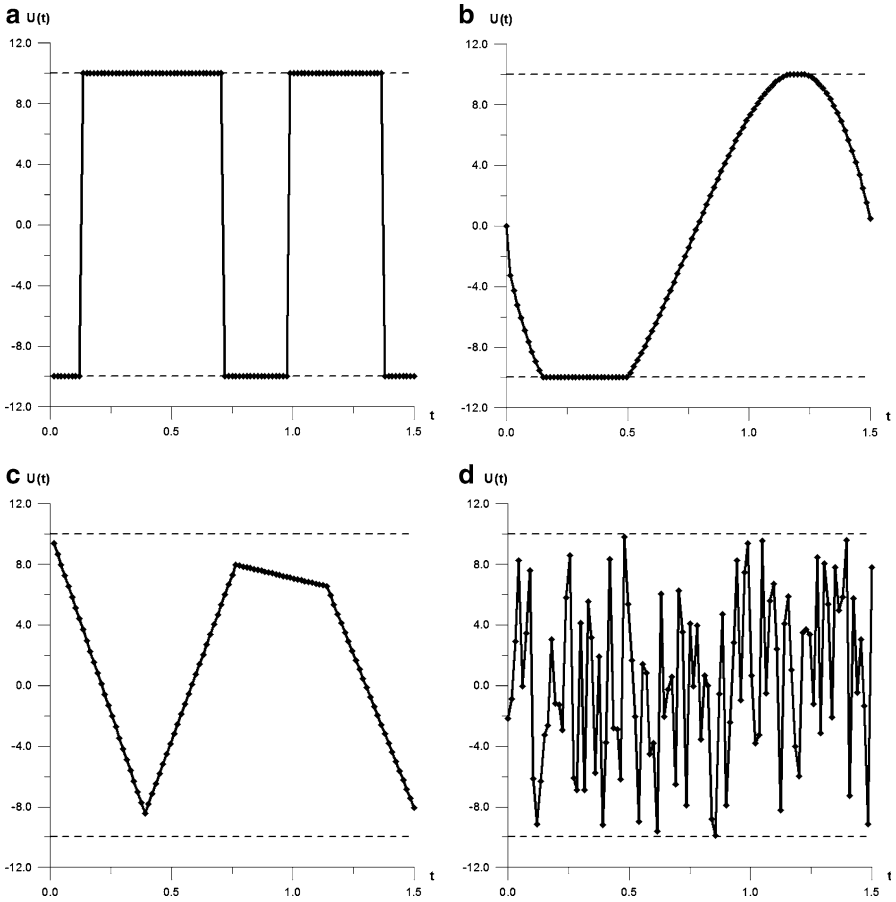


Fig. 1 Examples of generated controls in the form of: (**a**) relay functions, (**b**) spline functions, (**c**) piecewise-linear functions, and (**d**) tabulated functions

6.1 Test Problem 1

The controllable dynamic process is described by the system of differential equations $\dot{x}_1 = x_2 + x_1 \sin x_1 + u_1$, $\dot{x}_2 = \sqrt{2.1 - u_1 \cos x_2}$, $t \in [0, 4]$. The value of phase vector at initial time $x_1(t_0) = 3$, $x_2(t_0) = 0$ and the set of feasible controls $U = [-1, 1]$ are given. It is necessary to minimize the nonconvex terminal functional $I(u) = -(x_1(t_1) - 5)^2 - (x_2(t_1 - 6)^2 \to \min$.

Table 1 The results of solving test problem 1

N	Value of objective functional	Extreme points x_1	x_2
1	−2.30262	4.1059	4.7740
2	−25.36658	10.036	6.0914
3	−0.40256	4.7926	6.5996

Table 2 The results of solving test problem 2

N	Value of objective functional	Extreme points x_1	x_2
1	1.06054	0.6120	2.0917
2	0.95589	2.4438	2.6307
3	−2.83027	2.7093	2.2141
4	1.11165	0.5661	−0.8103
5	1.188471	0.1638	−0.5743

Table 3 The results of solving test problem 3

N	Value of objective functional	Extreme points x_1	x_2
1	28.39093	0.2463	4.3195
2	0.18870	−1.4336	9.5260
3	149.9012	−1.5173	−2.7325

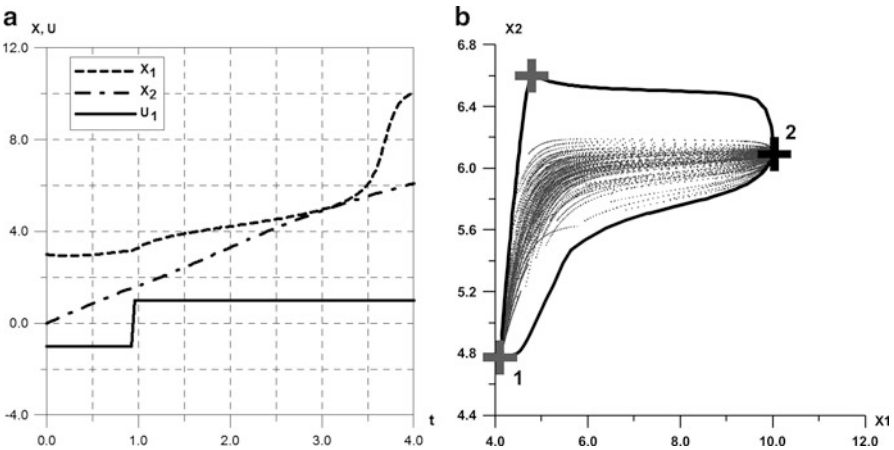


Fig. 2 (**a**) Optimal control and respective trajectories of the system, (**b**) Attainable set with extreme points in test problem 1

Computation for test problem 1	
Minimization phase	$I(u) = -2.30260$
Tunneling phase	17 iterations of the curvilinear search algorithm $I(u) = -2.34511$
Minimization phase	$I(u) = -25.36658$
Tunneling phase	50 iterations of the curvilinear search algorithm
Optimal solution: $I^*(u) = -25.36658$	
The number of Cauchy problems is 20684. The computing time is 6 s.	

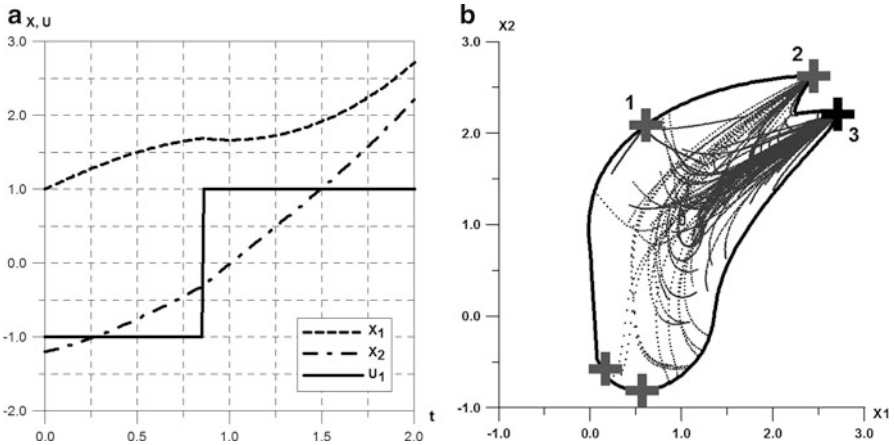


Fig. 3 (**a**) Optimal control and respective trajectories of the system, (**b**) attainable set with extreme points in test problem 2

Computation for test problem 2	
Minimization phase	$I(u) = 1.06066$
Tunneling phase	3 iterations of the curvilinear search algorithm $I(u) = 0.95504$
Minimization phase	$I(u) = 0.95504$
Tunneling phase	13 iterations of the curvilinear search algorithm $I(u) = 0.94117$
Minimization phase	$I(u) = -2.82955$
Tunneling phase	50 iterations of the curvilinear search algorithm
Optimal solution: $I^*(u) = -2.82955$	
The number of Cauchy problems is 43379. The computing time is 9 s	

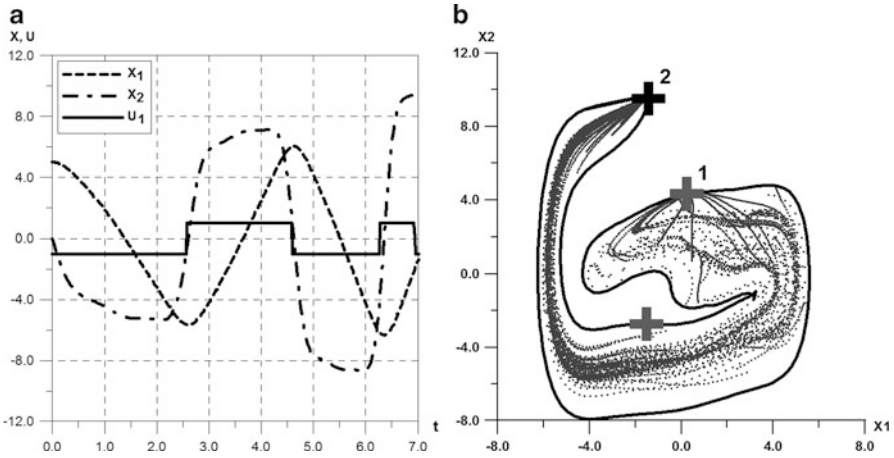


Fig. 4 (**a**) Optimal control and respective trajectories of the system, (**b**) attainable set with extreme points in test problem 3

Computation for test problem 3	
Minimization phase	$I(u) = 28.44292$
Tunneling phase	Nine iterations of the curvilinear search algorithm $I(u) = 22.66894$
Minimization phase	$I(u) = 0.34703$
Tunneling phase	Fifty iterations of the curvilinear search algorithm
Optimal solution: $I^*(u) = 0.34703$	
The number of Cauchy problems is 110,885. The computing time is 30 s	

6.2 Test Problem 2

Test problem 2 is formulated in the following way: $\dot{x}_1 = x_2 \cdot u_1$, $\dot{x}_2 = x_1 + u_1/(x_1^2 + x_2^2)$, $x_1(t_0) = 1$, $x_2(t_0) = -1.2$, $u \in [-1, 1]$, $t \in [0, 2]$. It is necessary to minimize the functional $I(u) = \left(\sqrt{2.72 - x_1} - \frac{2.72 - x_1}{7}\right) \cdot \mathrm{e}^{0.926 \cdot x_1} - 1.481 x_1 - 0.014 x_2^2 \to \min$.

6.3 The Test Problem 3

In test problem 3 the controllable dynamic process is described by the following nonlinear system of differential equations: $\dot{x}_1 = x_2$, $\dot{x}_2 = u_1 - x_1 + \frac{x_1^3}{6} - \frac{x_1^5}{120}$, $t \in [0, 7]$. The initial value of phase vector $x_1(t_0) = 5$, $x_2(t_0) = 0$ and the set of feasible controls $U = [-1, 1]$ are specified. The terminal functional is formulated as follows: $I(u) = (x_1 + 1)^2 + (x_2 - 9.5)^2 \to \min$.

7 Conclusion

The software implementation of the suggested tunneling algorithm has been tested on a collection of nonconvex OCP. The computational experiments have demonstrated the efficiency of the proposed approach. The global extremum known from the source was found in all solved nonconvex OCP.

The tunneling algorithm suggested in this chapter makes it possible to solve multiextremal OCP faster as compared to the existing approaches. The developed approach was implemented and included in the software OPTCON-III [28] which is intended to solve a wide class of OCP.

Acknowledgements This work is partly supported by Grants N 12-01-00193 and N 10-01-00595 of the Russian Foundation for Basic Research.

References

1. Banga, J.R., Seider, W.D.: Global optimization of chemical processes using stochastic algorithms. In: Floudas, C.A., Pardalos, P.M. (eds.) State of the Art in Glob. Optim., 563–583 (1996)
2. Banga, J.R., Versyck, K.J., Van Impe, J.F.: Computation of optimal identification experiments for nonlinear dynamic process models: a stochastic global optimization approach. Ind. Eng. Chem. Res. **41**, 2425–2430 (2002)
3. Bobbin, J., Yao, X.: Solving optimal control problems with a cost on changing control by evolutionary algorithms. Proc. IEEE Int. Conf. Evol. Comput., Indianapolis, USA, 331–336 (1997)
4. Chachuat, B., Latifi, M.A.: A new approach in deterministic global optimization of problems with ordinary differential equations. In: Floudas, C.A., Pardalos, P.M. (eds.) Front. in Glob. Optim., 83–108 (2003)
5. Chentsov, A.G.: Asymptotic attainability. Kluwer, Dordrecht (1997)
6. Esposito, W.R., Floudas, C.A.: Determistic global optimization in nonlinear optimal control problems. J. Glob. Optim. **17**, 97–126 (2000b)
7. Gornov, A.Yu.: The computational technologies for solving optimal control problems. Nauka, Novosibirsk (2009) (in Russian)
8. Gornov, A.Yu., Zarodnyuk, T.S.: Method of curvilinear search for global extremum in optimal control problems. Contemp. Technol. Syst. Anal. Simul. **3**, 19–27 (2009) (in Russian)
9. Hashem, M.M.A., Watanabe, K., Izumi, K., A new evolution strategy and its application to solving optimal control problems. JSME Int. J., **41**(3), 406–412 (1998)
10. Levy, A.V., Montalvo, A.: The tunneling algorithm for the global minimization of functions. SIAM J. Sci. Stat. Comput. **6**, 15–29 (1985)
11. Lin, Y.D., Stadtherr, M.A.: Deterministic global optimization of nonlinear dynamic systems. AIChE J. **53**(4), 866–875 (2007)
12. Liu, Y., Teo, K.L.: An adaptive dual parametrization algorithm for quadratic semi-infinite programming problems. J. Glob. Optim. **24**(2), 205–217 (2002)
13. Long, C.E., Polisetty, P.K., Gatzke, E.P.: Deterministic global optimization for nonlinear model predictive control of hybrid dynamic systems. Int. J. Robust Nonlinear Control **17**(13), 1232–1250 (2007)
14. Lopez-Cruz, I.L.: Efficient Evolutionary Algorithms for Optimal Control. PhD Thesis, Wageningen University, The Netherlands (2002)

15. Luus, R.: Piecewise linear continuous optimal control by using iterative dynamic programming. Ind. Eng. Chem. Res. **32**, 859–865 (1993)
16. Papamichail, I., Adjiman, C.S.: A rigorous global optimization algorithm for problems with ordinary differential equations. J. Glob. Optim. **24**, 1–33 (2002)
17. Pardalos, P., Yatsenko, V.: Optimization approach to the estimation and control of lyapunov exponents journal of optimization theory and its applications **128**(1), 29–48 (2006)
18. Roubos, J.A., Van Straten, G., Van Boxtel, A.J.B.: An evolutionary strategy for fed-batch bioreactor optimization: concepts and performance. J. Biotechnol. **67**, 173–187 (1999)
19. Sim, Y.C., Leng, S.B., Subramaniam, V.: A combined genetic algorithms-shooting method approach to solving optimal control problems. Int. J. Syst. Sci. **31**(1), 83–89 (2000)
20. Singer, A.B., Barton, P.I.: Global solution of optimization problems with dynamic systems embedded. In: Floudas, C.A., Pardalos, P.M. (eds.) Front. in Glob. Optim., 477–498 (2003)
21. Singer, A.B., Barton, P.I.: Global optimization with nonlinear ordinary differential equations. J. Glob. Optim. **34**(2), 159–190 (2006)
22. Smith, S., Stonier, R.: Applying evolution program techniques to constrained continuous optimal control problems. Proc. IEEE Conf. Evol. Comput., Piscataway, USA, 285–290 (1996)
23. Storn, R., Price, K., Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. **11**, 341–359 (1997)
24. Vilkov, A.V., Zhidkov, N.P., Shchedrin, B.M.: A method of search for the global minimum of the single-variable function. J. Comput. Math. and Math. Phys. **4**, 1040–1042 (1975) (in Russian)
25. Wang, F.S., Chiou, J.P.: Optimal control and optimal time location problems of differential-algebraic systems by differential evolution. Ind. Eng. Chem. Res. **36**, 5348–5357 (1997)
26. Yamashita, Y., Shima, M.: Numerical computational method using genetic algorithm for the optimal control problem with terminal constraints and free parameters. Nonlin. Anal., Theory, Meth. and Appl. **30**(4), 2285–2290 (1997)
27. Zarodnyuk, T.S., Gornov, A.Yu.: Test collection of nonconvex optimal control problems. Proc. Conf. “Lyapunov Read. and Present. of Inf. Technol.”, Irkutsk, Russia (2008) (in Russian)
28. Zarodnyuk, T.S., Gornov, A.Yu.: The basic components of program software OPTCON-III for solving nonconvex optimal control problem. Proc. of XVI Baikal All-Russian Conf. “Inf. and Math. Technol. in Sci. and Control”, Irkutsk, Russia (2010) (in Russian)
29. Zhigljavsky, A.A., Zhilinskas, A.G.: The methods for global extremum search. Nauka, Moscow (1991) (in Russian)

Solving Linear Systems with Polynomial Parameter Dependency with Application to the Verified Solution of Problems in Structural Mechanics

Jürgen Garloff, Evgenija D. Popova, and Andrew P. Smith

Abstract We give a short survey on methods for the enclosure of the solution set of a system of linear equations where the coefficients of the matrix and the right hand side depend on parameters varying within given intervals. Then we present a hybrid method for finding such an enclosure in the case that the dependency is polynomial or rational. A general-purpose parametric fixed-point iteration is combined with efficient tools for range enclosure based on the Bernstein expansion of multivariate polynomials. We discuss applications of the general-purpose parametric method to linear systems obtained by standard finite element analysis of mechanical structures and illustrate the efficiency of the new parametric solver.

1 Introduction

In this chapter we consider linear systems

$$A(x) \cdot s = d(x), \tag{1a}$$

where the coefficients of the $m \times m$ matrix $A(x)$ and the vector $d(x)$ are functions of n parameters $x_1, \ldots, x_n$ varying within given intervals $[x_1], \ldots, [x_n]$

$$a_{ij}(x) = a_{ij}(x_1, \ldots, x_n), \quad d_i(x) = d_i(x_1, \ldots, x_n), \quad i, j = 1, \ldots, m, \tag{1b}$$

$$x \in [x] = ([x_1], \ldots, [x_n])^\top. \tag{1c}$$

J. Garloff (✉) • A.P. Smith
University of Applied Sciences / HTWG Konstanz, Postfach 100543,
D-78405 Konstanz, Germany
e-mail: garloff@htwg-konstanz.de; smith@htwg-konstanz.de

E.D. Popova
Institute of Mathematics & Informatics, Bulgarian Academy of Sciences,
Acad. G. Bonchev str., Bldg. 8, BG-1113 Sofia, Bulgaria
e-mail: epopova@bio.bas.bg

A. Chinchuluun et al. (eds.), *Optimization, Simulation, and Control*, Springer Optimization and Its Applications 76, DOI 10.1007/978-1-4614-5131-0_19,

The set of solutions to (1a)–(1c), called the *parametric solution set*, is

$$\Sigma = \Sigma(A(x), d(x), [x]) := \{s \in \mathbb{R}^m \mid A(x) \cdot s = d(x) \text{ for some } x \in [x]\}. \quad (2)$$

Engineering problems that involve such parametric linear systems may stem from structural mechanics, e.g., [3, 4, 21, 26, 29, 38, 42], the design of electrical circuits [5,6], resistive networks [10], and robust Monte Carlo simulation [17], to name but a few examples. The source of parametric uncertainty is often the lack of precise data which may result from a lack of knowledge due to, e.g., measurement imprecision or manufacturing imperfections, or an inherent variability in the parameters, e.g., physical constants are only known to within certain bounds.

The parametric solution set can be described explicitly only in very simple cases. Therefore, one attempts to find the smallest axis-aligned box in $\mathbb{R}^m$ containing Σ. Since even this set can only be found easily in some special cases, it is more practical to attempt to compute a tight outer approximation to this box.

The chapter is organised as follows. In Sect. 2 we introduce the basic definitions and rules of interval arithmetic. Which is a fundamental tool of our approach. In this section we also compare the interval solution set with the parametric solution set and give a short overview of methods for its enclosure. In Sect. 3.1 we present a method for the enclosure of the parametric solution set, called the *parametric residual iteration method.* This method needs tight bounds on the range of multivariate functions. In the applications we will present later in this chapter the coefficient functions (1b) are polynomials or rational functions. To find the range of a multivariate polynomial, we recall in Sect. 3.2 a method which is based on the expansion of a polynomial into Bernstein polynomials, termed the *Bernstein form*. Implementation issues concerning the combination of the parametric residual iteration method with the Bernstein form are discussed in Sect. 3.3. We apply the combined approach in Sect. 4 to some problems of structural mechanics and draw some conclusions in Sect. 5.[1]

2 The Parametric Solution Set

2.1 Interval Arithmetic

Let $\mathbb{IR}$ denote the set of the compact, nonempty real intervals. The arithmetic operation $\circ \in \{+, -, \cdot, /\}$ on $\mathbb{IR}$ is defined in the following way:

If $a = [\underline{a}, \overline{a}], b = [\underline{b}, \overline{b}] \in \mathbb{IR}$, then

$$a + b = [\underline{a} + \underline{b}, \overline{a} + \overline{b}],$$
$$a - b = [\underline{a} - \overline{b}, \overline{a} - \underline{b}],$$

[1] Preliminary results were presented at the 2nd International Conference on Uncertainty in Structural Dynamics, Sheffield, UK, June 15–17, 2009.

$$a \cdot b = \left[\min\{\underline{a}\underline{b}, \underline{a}\overline{b}, \overline{a}\underline{b}, \overline{a}\overline{b}\}, \max\{\underline{a}\underline{b}, \underline{a}\overline{b}, \overline{a}\underline{b}, \overline{a}\overline{b}\}\right],$$
$$a / b = [\min\{\underline{a}/\underline{b}, \underline{a}/\overline{b}, \overline{a}/\underline{b}, \overline{a}/\overline{b}\},$$
$$\max\{\underline{a}/\underline{b}, \underline{a}/\overline{b}, \overline{a}/\underline{b}, \overline{a}/\overline{b}\}], \text{ if } 0 \notin b.$$

As a consequence of these definitions we obtain the inclusion isotonicity of the interval arithmetic operations: If $a_1, b_1 \in \mathbb{IR}$ with $a_1 \subseteq a$ and $b_1 \subseteq b$ then it holds that

$$a_1 \circ b_1 \subseteq a \circ b.$$

Note that some relations known to be true in the set $\mathbb{R}$, e.g., the distributive law, are not valid in $\mathbb{IR}$. Here we have the weaker subdistributive law

$$a \cdot (b + c) \subseteq a \cdot b + a \cdot c \text{ for } a, b, c \in \mathbb{IR}.$$

The width of an interval $a = [\underline{a}, \overline{a}]$ is defined as

$$\omega(a) = \overline{a} - \underline{a}.$$

By $\mathbb{IR}^n$ and $\mathbb{IR}^{n \times n}$ we denote the set of n-vectors and n-by-n matrices with entries in $\mathbb{IR}$, respectively. For a nonempty bounded set $\mathcal{S} \subseteq \mathbb{R}^n$, define its interval hull by $\square\mathcal{S} := [\inf \mathcal{S}, \sup \mathcal{S}] = \cap\{[s] \in \mathbb{IR}^n \mid \mathcal{S} \subseteq [s]\}$.

Where the end-points of an interval are stored as floating-point numbers, it is necessary to use *outward rounding* in all operations, viz. the infimum is rounded down and the supremum is rounded up. In this way, interval operations deliver guaranteed results even in the presence of rounding errors with floating-point arithmetic.

Further details on arithmetic with intervals may be found in [1, 22].

2.2 The Interval Solution Set Versus the Parametric Solution Set

A system of linear interval equations is a collection of systems

$$A \cdot s = d,\ A \in [A],\ d \in [d],\ \text{where } [A] \in \mathbb{IR}^{m \times m},\ [d] \in \mathbb{IR}^m; \tag{3}$$

its solution set

$$\{s \in \mathbb{R}^m \mid \exists A \in [A],\ \exists d \in [d] : A \cdot s = d\} \tag{4}$$

is called here the *interval solution set*. There are many methods for the enclosure of the interval solution set (cf. [1, 22]). With the parametric linear system (1a) a system (3) is associated which is obtained when each entry in (1b) is replaced by an enclosure for the range of the functions a_{ij} and d_i over $[x]$. In general, the resulting

interval system can be more easily solved than the parametric system. However, the dependencies between the parameters are lost, and so the interval solution set is in general much larger than the parametric solution set.

2.3 Prior Work on the Parametric Solution Set

One of the earliest papers on the solution of linear systems with nonlinear parameter dependencies is [8], cf. [9]. Later works focus on the solution of systems of linear equations whose coefficient matrices enjoy a special structure. Here the interval solution set (4) is restricted in such a way that only matrices which have this special structure are considered. The restricted solution set can also often be represented as a parametric solution set (2), cf. [12] for examples and references. In the sequel we survey some methods for the enclosure of the parametric solution set which have a wider range of applicability.

A method which is applicable to parameter dependencies which can be represented as

$$A(x) \;=\; \sum_{k=1}^{n} x_k A^{(k)},\; d(x) \;=\; \sum_{k=1}^{n} x_k d^{(k)},\; A^{(k)} \in \mathbb{R}^{m\times m},\; d^{(k)} \in \mathbb{R}^m,\; k = 1,\ldots,n,$$

was recently given in [14]. This parameter dependency covers the (skew-) symmetric, Toeplitz, and Hankel matrices and was also considered in [4].

In [10] parametric linear systems are considered where the uncertain parameters x_i enter the system (1a) in a rank-one manner. As an example, any planar resistive network has the property that with resistances associated with the parameters x_i the resulting system of linear equations, corresponding to application of Kirchhoff's laws, has a rank-one structure. Such systems are solved in [5,6] by application of the Sherman–Morrison formula. For systems with a rank-one structure, results are obtained in [10] which allow one to decide which parameters influence components of the solution

$$s(x) \;=\; A(x)^{-1} d(x)$$

in a monotone, convex, or concave manner. Such information greatly facilitates the computation of an enclosure of the solution set (2).

Another direct method is presented in [15]. Here the coefficient functions of (1a) are assumed only to be continuous. They are approximated by linear functions in such a way that one obtains a superset of (2). An interval enclosure for this superset is determined as an interval vector whose midpoint is obtained as the solution of a certain system of linear equations. The vector which contains the (half-) widths of the component intervals is computed as the solution of another system and therefore must be positive, which is a restriction of the method.

In [36] a direct method is proposed for the case of linear parameter dependency based on inclusion theorems of Neumaier [22]. However, a prerequisite for this

method is that a matrix of coefficients generated from the inverse of the midpoint of the interval matrix A must be an H-matrix [22], a condition which seems to be rarely satisfied for typical problems.

The method which presently seems to have the widest range of applicability is the parametric linear solver developed by the second author (E. D. P.); see Sect. 3.1 for details.

3 Methodology

3.1 The Residual Iteration Method

In this section we consider a self-verified method for bounding the parametric solution set. This is a general-purpose method since it does not assume any particular structure among the parameter dependencies. The method originates in the inclusion theory for nonparametric problems, which is discussed in many works (cf. [34] and the literature cited therein). The basic idea of combining the Krawczyk-operator [16] and the existence test by Moore [20] is further elaborated by S. Rump [33] who proposes several improvements leading to inclusion theorems for the interval solution (4). In [34, Theorem 4.8] S. Rump gives a straightforward generalisation to (1a) with affine-linear dependencies in the matrix and the right-hand side. With obvious modifications, the corresponding theorems can also be applied directly to linear systems involving nonlinear dependencies between the parameters in $A(x)$ and $d(x)$. This is demonstrated in [26,29]. The following theorem is a general formulation of the enclosure method for linear systems involving arbitrary parametric dependencies.

Theorem 1. *Consider a parametric linear system defined by (1a)–(1c). Let* $R \in \mathbb{R}^{m\times m}$, $[y] \in \mathbb{IR}^m$, $\tilde{s} \in \mathbb{R}^m$ *be given and define* $[z] \in \mathbb{IR}^m$, $[C] \in \mathbb{IR}^{m\times m}$ *by*

$$[z] := \square\{R\,(d(x) - A(x)\tilde{s}) \mid x \in [x]\},$$
$$[C] := \square\{I - R\cdot A(x) \mid x \in [x]\},$$

where I *denotes the identity matrix. Define* $[v] \in \mathbb{IR}^m$ *by means of the following Gauss–Seidel iteration:*

$$1 \le i \le m \;:\; [v_i] := \Big\{[z] + [C]\cdot([v_1],\ldots,[v_{i-1}],[y_i],\ldots,[y_m])^\top\Big\}_i.$$

If $[v] \subseteq [y]$ *and* $[v]_i \ne [y]_i$ *for* $i = 1,\ldots,n$, *then* R *and every matrix* $A(x)$ *with* $x \in [x]$ *are regular, and for every* $x \in [x]$ *the unique solution* $\widehat{s} = A^{-1}(x)d(x)$ *of (1a)–(1c) satisfies* $\widehat{s} \in \tilde{s} + [v]$.

In the examples we present in Sect. 4, we have chosen $R \approx A(\check{x})^{-1}$ and $\tilde{s} \approx R^{-1}d(\check{x})$, where $\check{x}$ is the midpoint of $[x]$.

The above theorem generalises [34, Theorem 4.8] by stipulating a sharp enclosure of $C(x) := I - R \cdot A(x)$ for $x \in [x]$, instead of using the interval extension $C([x])$. A sharp enclosure of the iteration matrix $C(x)$ for $x \in [x]$ is also required by other authors (who do not refer to [34]), e.g., [4], without addressing the issue of rounding errors. Examples demonstrating the extended scope of application of the generalised inclusion theorem can be found in [23, 25, 31]. It should be noted that the above theorem provides strong regularity (cf. [25]), which is a weaker but sufficient condition for regularity of the parametric matrix.

When aiming to compute a self-verified enclosure of the solution to a parametric linear system by the above inclusion method, a fixed-point iteration scheme is proven to be very useful. A detailed presentation of the computational algorithm can be found in [26, 33].

In case of arbitrary nonlinear dependencies between the uncertain parameters, computing $[z]$ and $[C]$ in Theorem 1 requires a sharp range enclosure of nonlinear functions. This is a key problem in interval analysis, and there exists a huge number of methods and techniques devoted to this problem, with no one method being universal. In this work we restrict ourselves to linear systems where the elements of $A(x)$ and $d(x)$ are rational functions of the uncertain parameters. In this case the coefficients of $z(x) = R(d(x) - A(x)\tilde{s})$ and $C(x)$ are also rational functions of x. The quality of the range enclosure of $z(x)$ will determine the sharpness of the parametric solution set enclosure. In [26] the above inclusion theorem is combined with a simple interval arithmetic technique providing inner and outer bounds for the range of monotone rational functions. The arithmetic of generalised (proper and improper) intervals is considered as an intermediate computational tool for eliminating the dependency problem in range computation and for obtaining inner estimations by outwardly rounded interval arithmetic. Since this methodology is not efficient in the general case of non-monotone rational functions, in this work we combine the parametric fixed-point iteration with range enclosing tools based on the Bernstein expansion of multivariate polynomials.

3.2 *Bernstein Enclosure of Polynomial Ranges*

In this section we recall some properties of the Bernstein expansion which are fundamental to our approach, cf. [2, 11, 41] and the references therein.

Firstly, some notation is introduced. We define multi-indices $i = (i_1, \ldots, i_n)^T$ as vectors, where the n components are nonnegative integers. The vector 0 denotes the multi-index with all components equal to 0. Comparisons are used entrywise. Also the arithmetic operators on multi-indices are defined componentwise such that $i \odot l := (i_1 \odot l_1, \ldots, i_n \odot l_n)^T$, for $\odot = +, -, \times$, and $/$ (with $l > 0$). For instance, i/l, $0 \le i \le l$, defines the Greville abscissae. For $x \in \mathbb{R}^n$ its monomials are

$$x^i := \prod_{\mu=1}^{n} x_\mu^{i_\mu}. \tag{5}$$

For the n-fold sum we use the notation

$$\sum_{i=0}^{l} := \sum_{i_1=0}^{l_1} \cdots \sum_{i_n=0}^{l_n} . \tag{6}$$

The generalised binomial coefficient is defined by

$$\binom{l}{i} := \prod_{\mu=1}^{n} \binom{l_\mu}{i_\mu}. \tag{7}$$

For reasons of familiarity, the Bernstein coefficients are denoted by b_i; this should not be confused with components of the right-hand side vector b of (1a). Hereafter, a reference to the latter will be made explicit.

3.2.1 The Bernstein Form

An n-variate polynomial p,

$$p(x) = \sum_{i=0}^{l} a_i x^i, \quad x = (x_1, \ldots, x_n), \tag{8}$$

can be represented over

$$\begin{aligned} [x] &:= [\underline{x}_1, \overline{x}_1] \times \cdots \times [\underline{x}_n, \overline{x}_n], \\ \underline{x} &= (\underline{x}_1, \ldots, \underline{x}_n),\ \overline{x} = (\overline{x}_1, \ldots, \overline{x}_n), \end{aligned} \tag{9}$$

as

$$p(x) = \sum_{i=0}^{l} b_i B_i(x), \tag{10}$$

where B_i is the *i-th Bernstein polynomial of degree* $l = (l_1, \ldots, l_n)$,

$$B_i(x) = \binom{l}{i} \frac{(x - \underline{x})^i (\overline{x} - x)^{l-i}}{(\overline{x} - \underline{x})^l}, \tag{11}$$

and the so-called *Bernstein coefficients* b_i of the same degree are given by

$$b_i = \sum_{j=0}^{i} \frac{\binom{i}{j}}{\binom{l}{j}} (\overline{x} - \underline{x})^j \sum_{\kappa=j}^{l} \binom{\kappa}{j} \underline{x}^{\kappa - j} a_\kappa, \quad 0 \le i \le l. \tag{12}$$

The essential property of the Bernstein expansion is the *range enclosing property*, namely that the range of p over $[x]$ is contained within the interval spanned by the minimum and maximum Bernstein coefficients:

$$\min_i \{b_i\} \le p(x) \le \max_i \{b_i\},\ x \in [x]. \tag{13}$$

It is also worth noting that the values attained by the polynomial at the vertices of $[x]$ are identical to the corresponding vertex Bernstein coefficients, e.g., $b_0 = p(\underline{x})$ and $b_l = p(\overline{x})$. The *sharpness property* states that the lower (resp. upper) bound provided by the minimum (resp. maximum) Bernstein coefficient is sharp, i.e. there is no underestimation (resp. overestimation), if and only if this coefficient corresponds to a vertex of $[x]$.

The traditional approach (see, e.g., [11, 41]) requires that all of the Bernstein coefficients are computed, and their minimum and maximum are determined. By use of an algorithm (cf. [11, 41]) which is similar to de Casteljau's algorithm (see, e.g., [32]), this computation can be made efficient, with time complexity $O(n\hat{l}^{n+1})$ and space complexity (equal to the number of Bernstein coefficients) $O((\hat{l}+1)^n)$, where $\hat{l} = \max_{i=1}^{n} l_i$. This exponential complexity is a drawback of the traditional approach, rendering it infeasible for polynomials with moderately many (typically, ten or more) variables.

In [37] a new method for the representation and computation of the Bernstein coefficients is presented, which is especially well suited to sparse polynomials. With this method the computational complexity typically becomes nearly linear with respect to the number of the terms in the polynomial, instead of exponential with respect to the number of variables. This improvement is obtained from the results surveyed in the following sections. For details and examples the reader is referred to [37].

3.2.2 Bernstein Coefficients of Monomials

Let $q(x) = x^r$, $x = (x_1, \ldots, x_n)$, for some $0 \leq r \leq l$. Then the Bernstein coefficients of q (of degree l) over $[x]$ (9) are given by

$$b_i = \prod_{m=1}^{n} b_{i_m}^{(m)}, \tag{14}$$

where $b_{i_m}^{(m)}$ is the i_mth Bernstein coefficient (of degree l_m) of the univariate monomial x^{r_m} over $[\underline{x}_m, \overline{x}_m]$. If the box $[x]$ is restricted to a single orthant of $\mathbb{R}^n$ then the Bernstein coefficients of q over $[x]$ are monotone with respect to each variable x_j, $j = 1, \ldots, n$.

With this property, for a single-orthant box, the minimum and maximum Bernstein coefficients must occur at a vertex of the array of Bernstein coefficients. This also implies that the bounds provided by these coefficients are sharp; see the aforementioned sharpness property. Finding the minimum and maximum Bernstein coefficients is therefore straightforward; it is not necessary to explicitly compute the whole set of Bernstein coefficients. Computing the component univariate Bernstein coefficients for a multivariate monomial has time complexity $O(n(\hat{l}+1)^2)$. Given the exponent r and the orthant in question, one can determine whether the monomial (and its Bernstein coefficients) is increasing or decreasing with respect to each coordinate direction, and one then merely needs to evaluate the monomial at these two vertices.

Without the single-orthant assumption, monotonicity does not necessarily hold, and the problem of determining the minimum and maximum Bernstein coefficients is more complicated. For boxes which intersect two or more orthants of $\mathbb{R}^n$, the box can be bisected, and the Bernstein coefficients of each single-orthant sub-box can be computed separately.

3.2.3 The Implicit Bernstein Form

Firstly, we can observe that since the Bernstein form is linear, if a polynomial p consists of t terms, as follows,

$$p(x) = \sum_{j=1}^{t} a_{i_j} x^{i_j}, \quad 0 \le i_j \le l, \quad x = (x_1, \ldots, x_n), \tag{15}$$

then each Bernstein coefficient is equal to the sum of the corresponding Bernstein coefficients of each term, as follows:

$$b_i = \sum_{j=1}^{t} b_i^{(j)}, \quad 0 \le i \le l, \tag{16}$$

where $b_i^{(j)}$ are the Bernstein coefficients of the jth term of p. (Hereafter, a superscript in brackets specifies a particular term of the polynomial. The use of this notation to indicate a particular coordinate direction, as in the previous section, is no longer required.)

Therefore one may implicitly store the Bernstein coefficients of each term and compute the Bernstein coefficients as a sum of t products, only as needed. The implicit Bernstein form thus consists of computing and storing the n sets of univariate Bernstein coefficients (one set for each component univariate monomial) for each of t terms. Computing this form has time complexity $\mathrm{O}(nt(\hat{l}+1)^2)$ and space complexity $\mathrm{O}(nt(\hat{l}+1))$, as opposed to $\mathrm{O}((\hat{l}+1)^n)$ for the explicit form. Computing a ingle Bernstein coefficient from the implicit form requires $(n+1)t-1$ arithmetic operations.

3.2.4 Determination of the Bernstein Enclosure for Polynomials

We consider the determination of the minimum Bernstein coefficient; the determination of the maximum Bernstein coefficient is analogous. For simplicity we assume that $[x]$ is restricted to a single orthant.

We wish to determine the value of the multi-index of the minimum Bernstein coefficient in each direction. In order to reduce the search space (among the $(\hat{l}+1)^n$ Bernstein coefficients) we can exploit the monotonicity of the Bernstein coefficients of monomials and employ uniqueness, monotonicity, and dominance

tests cf. [37] for details. As the examples in [37] show, it is often possible in practice to dramatically reduce the number of Bernstein coefficients that have to be computed.

3.3 Software Tools

In our implementation we have combined software for the parametric residual iteration method with software developed for the enclosure of the range of a multivariate polynomial using the implicit Bernstein form. In the case of a rational, non-polynomial parameter dependency, the ranges of the numerator and the denominator have to be bounded independently at the expense of some overestimation. In both packages interval arithmetic is used throughout, such that the resulting enclosure for the parametric solution set can be *guaranteed* also in the presence of rounding errors. The software tools for the residual iteration are implemented in a *Mathematica* [40] environment by the second author (E. D. P); this software is publically available [24, 26]. The software for the Bernstein form is written by the last author (A. P. S.) and utilises the C++ interval library `filib++` [18, 19]. Since this is a specialised software exhibiting good performance there is no reason for its re-implementation in *Mathematica*. In order to shorten the development time and to preserve the beneficial properties of both implementation environments, we have connected both software packages into a new parametric solver via the *MathLink* [40] communication protocol, for details see [12]. However, this connection leads to longer computing times compared to an implementation in a single environment. For details of the implementation and the accessibility of the combined software see [12].

4 Application to Structural Mechanics

A standard method for solving problems in structural mechanics, such as linear static problems, is the finite element method (FEM). In the case of linearised geometric displacement equations and linear elastic material behaviour, the method leads to a system of linear equations which in the presence of uncertain parameters becomes a parametric system. Treating the parametric system as an interval system and using a typical interval method for the enclosure of (4) in general result in intervals for the quantities sought which are too wide for practical purposes.

In [21, 42] the authors combine an element-by-element (EBE) formulation, where the elements are kept disassembled, with a penalty method for imposing the necessary constrains for compatibility and equilibrium, in order to reduce the overestimation in the solution intervals. This approach should be applied simultaneously with FEM and affects the construction of the global stiffness matrix and the right-hand side vector, making them larger. A nonparametric fixed-point iteration is

then used to solve the parametric interval linear system. While special construction methods are applied in [21], the parametric system obtained by standard FEM applied to a structural steel frame with partially constrained connections is solved by a sequence of interval-based (but not parametric) methods [3].

In the sequel we illustrate the usage of the new parametric solver based on bounding polynomial ranges by the implicit Bernstein form as described in Sect. 3.2. The improved efficiency is demonstrated by comparing both the computing time and the quality of the enclosure of the parametric solution set for the new solver and a previous solver which is based on the combination of the parametric residual iteration with the method for bounding the range of a rational function presented in [26], cf. Sect. 3.1. To compare the quality of two enclosures $[a]$ and $[b]$ with $[a] \subseteq [b]$ we employ a measure $\mathscr{O}_\omega$ for the overestimation of $[a]$ by $[b]$ which is defined by

$$\mathscr{O}_\omega([a],[b]) := 100(1-\omega([a])/\omega([b])), \tag{17}$$

where ω denotes the width of an interval.

The following examples were run on a PC with an AMD Athlon-64 3 GHz processor.

4.1 One-Bay Steel Frame

We consider a simple one-bay structural steel frame, as shown in Fig. 1, which was initially studied by interval methods in [3]. Following standard practice, the authors have assembled a parametric linear system of order eight and involving eight uncertain parameters. The typical nominal parameter values and the corresponding worst-case uncertainties, as proposed in [3] but converted to SI units, are shown in Table 1. The explicit analytic form of the given system involving polynomial parameter dependencies can be found in [3, 29].

As in [3, 29], we solved the system first with parameter uncertainties which are 1 % of the values presented in the last column of Table 1.

The previous parametric solver finds an enclosure for the solution set in about 0.34 s, whereas the new solver needs only 0.05 s. The quality of the enclosures provided by both solvers is comparable. As shown in [26, 29], the solution enclosure obtained by the parametric solver is better by more than one order of magnitude than the solution enclosure obtained in [3].

Based on the runtime efficiency of the new parametric solver, we next attempt to solve the same parametric linear system for the worst-case parameter uncertainties in Table 1 ranging between about 10 % and 46 %. Firstly, we notice that the parametric solution depends linearly on the parameter H, so that we can obtain a better solution enclosure if we solve two parametric systems with the corresponding end-points for H. Secondly, enclosures of the hull of the solution set are obtained by subdivision of the worst-case parameter intervals $(E_b, E_c, I_b, I_c, A_b, A_c, \alpha)^\top$

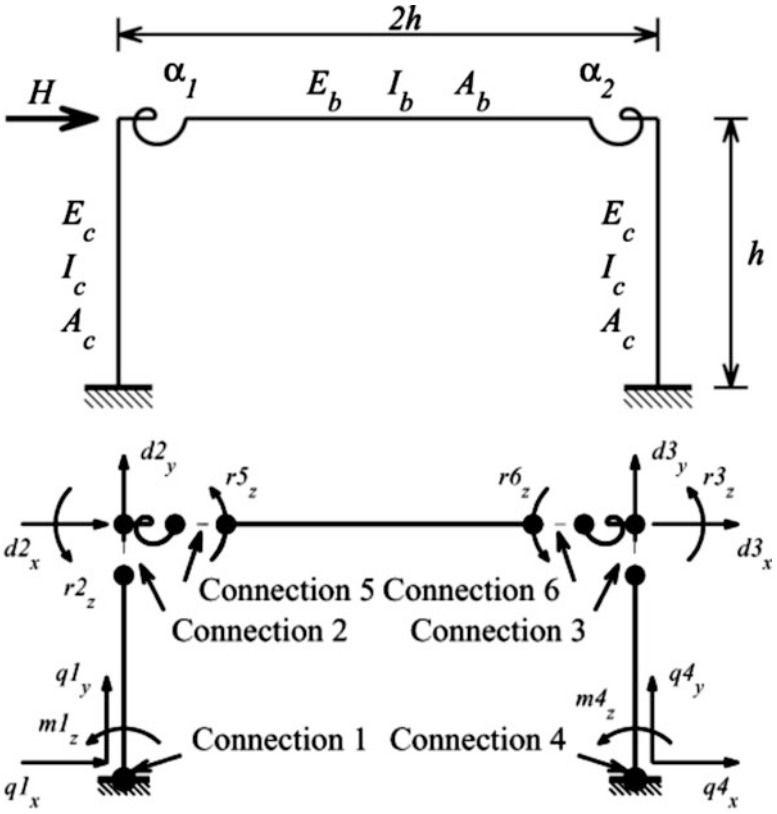


Fig. 1 One-bay structural steel frame [3]

Table 1 Parameters involved in the steel frame example

Parameter		Nominal value	Uncertainty
Young modulus	E_b	$1.999 * 10^8$ kN/m^2	$\pm 2.399 * 10^7$ kN/m^2
	E_c	$1.999 * 10^8$ kN/m^2	$\pm 2.399 * 10^7$ kN/m^2
Second moment	I_b	$2.123 * 10^{-4}$ m^4	$\pm 2.123 * 10^{-5}$ m^4
	I_c	$1.132 * 10^{-4}$ m^4	$\pm 1.132 * 10^{-5}$ m^4
Area	A_b	$6.645 * 10^{-3}$ m^2	$\pm 6.645 * 10^{-4}$ m^2
	A_c	$9.290 * 10^{-3}$ m^2	$\pm 9.290 * 10^{-4}$ m^2
External force	H	23.600 kN	± 9.801 kN
Joint stiffness	α	$3.135 * 10^5$ kNm/rad	$\pm 1.429 * 10^5$ kNm/rad
Length	L_c	3.658 m, L_b 7.316 m	

into $(2,2,2,2,1,1,6)^\top$ subintervals of equal width, respectively. We use more subdivision with respect to α since α is subject to the greatest uncertainty. The solution enclosure, obtained within 11 s, is given in Table 2. Moreover, the quality of the solution enclosure $[u]$ of the respective eight quantities is compared to the combinatorial solution $[\tilde{h}]$, i.e. the convex hull of the solutions to the point linear

Table 2 One-bay steel frame example with worst-case parameter uncertainties (Table 1)

	$10^5 *$ solution enclosure $[u]$	$\mathscr{O}_\omega([\tilde{h}],[u])$
$d2_x$:	[138.54954789, 627.59324779]	12.5
$d2_y$:	[0.29323100807, 2.1529383383]	8.0
$r2_z$:	[−129.02427835, −22.381136355]	23.7
$r5_z$:	[−113.21398401, −17.95789860]	25.6
$r6_z$:	[−105.9680866, −17.64526946]	25.0
$d3_x$:	[135.25570695, 616.85512710]	12.7
$d3_y$:	[−3.7624790816, −0.41629803684]	13.2
$r3_z$:	[−122.3361772, −21.69878778]	23.5

Solution enclosure $[u]$ found by dividing the parameter intervals $(E_b, E_c, I_b, I_c, A_b, A_c, \alpha)^\top$ into $(2,2,2,2,1,1,6)^\top$ subintervals of equal width, respectively
All interval end-points are multiplied by 10^5
The enclosure $[u]$ is compared to the combinatorial solution $[\tilde{h}]$

systems obtained when the parameters take all possible combinations of the interval end-points. The combinatorial solution serves as an *inner* estimation of the solution enclosure.

These results show that by means of a small number of subdivisions, the new parametric solver provides a good solution enclosure very quickly for the difficult problem of worst-case parameter uncertainties. Note that sharper bounds, close to the exact hull, can be obtained by proving the monotonicity properties of the parametric solution [28].

4.2 Two-Bay Two-Story Frame Model with 13 Parameters

We consider a two-bay two-story steel frame with IPE 400 beams and HE 280 B columns, as shown in Fig. 2, after [29]. The frame is subjected to lateral static forces and vertical uniform loads. Beam-to-column connections are considered to be semirigid, and they are modelled by single rotational spring elements. Applying conventional methods for the analysis of frame structures, a system of 18 linear equations is obtained, where the elements of the stiffness matrix and of the right-hand side vector are rational functions of the model parameters. We consider the parametric system resulting from a finite element model involving the following 13 uncertain parameters: A_c, I_c, E_c, A_b, I_b, E_b, c, $w_1, \ldots, w_4$, F_1, F_2. Their nominal values, taken according to the European Standard Eurocode3 [7], are given in Table 3. The explicit analytic form of the given parametric system can be found in [30].

The parametric system is solved for the element material properties $(A_c, \ldots, E_b)$, which are taken to vary within a tolerance of 1 % (i.e. $[x - x/200, x + x/200]$, where x is the corresponding parameter nominal value from Table 3) while the spring stiffness and all applied loadings are taken to vary within 10 % tolerance intervals.

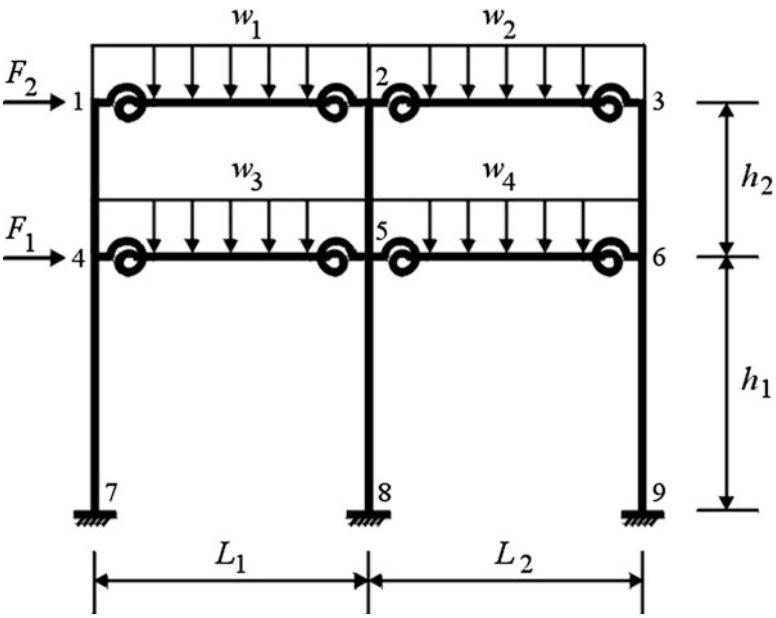


Fig. 2 Two-bay two-story steel frame [29]

Table 3 Parameters involved in the two-bay two-story frame example with their nominal values

Parameter	Columns (HE 280 B)	Beams (IPE 400)
Cross-sectional area	$A_c = 0.01314$ m^2	$A_b = 0.008446$ m^2
Moment of inertia	$I_c = 19270 * 10^{-8}$ m^4	$I_b = 23130 * 10^{-8}$ m^4
Modulus of elasticity	$E_c = 2.1 * 10^8$ kN/m^2	$E_b = 2.1 * 10^8$ kN/m^2
Length	$L_c = 3$ m	$L_b = 6$ m
Rotational spring stiffness	$c = 10^8$ kN	
Uniform vertical load	$w_1 = \cdots = w_4 = 30$ kN/m	
Concentrated lateral forces	$F_1 = F_2 = 100$ kN	

The previous parametric solver finds an enclosure for the solution set in about 7.4 s, whereas the new solver needs only about 1.3 s; here it is about six times faster. The solution enclosure provided by the new solver is also significantly tighter; the overestimation (17) of the components of the enclosure provided by the previous solver relative to the respective components found by the new solver ranges between 53.46 and 92.92.

An algebraic simplification applied to functional expressions in computer algebra environments may reduce the occurrence of interval variables, which could result in a sharper range enclosure. Such an algebraic simplification is expensive and when applied to complicated rational expressions usually does not result in a sharper range enclosure. For the sake of comparison, we have run the previous parametric solver in two ways: applying intermediate simplification during the range computation,

and without any algebraic simplification. The above results were obtained when the range computation does not use any algebraic simplification. When the range computation of the previous solver uses intermediate algebraic simplification, the cost of this improvement is that the computing time is approximately doubled; the results are obtained in 14.4 s. This is much slower, but provided a tighter enclosure of the solution set than the rational solver, based on polynomial ranges, which did not account for all the parameter dependencies. Here the overestimation of the new solver relative to the modified previous solver ranges between 18.62 and 37.07. It should be noted that given the complicated rational expressions such an improvement is not at all typical (in the next example, the improvement is only marginal at a much larger computation time possibly due to the more complicated expressions). Details may be found in [12].

4.3 Two-Bay Two-Story Frame Model with 37 Parameters

As a larger problem of a parametric system involving rational parameter dependencies, we consider the finite element model of the two-bay two-story steel frame from the previous example, where each structural element has properties varying independently within 1 % tolerance intervals. This does not change the order of the system but it now depends on 37 interval parameters. The explicit analytic form of the given parametric system can be found in [30]. Here the right-hand side vector is given to illustrate the dependencies.

$$\left(f_2,\ -\frac{1}{2}w_1Lb_1,\ -\frac{w_1Lb_1^2}{12(1+\frac{2Eb_1Ib_1}{cLb_1})},\ 0,\ -\frac{w_1Lb_1}{2}-\frac{w_2Lb_2}{2},\ \frac{w_1Lb_1^2}{12(1+\frac{2Eb_1Ib_1}{cLb_1})}\right.$$
$$-\frac{w_2Lb_2^2}{12(1+\frac{2Eb_2Ib_2}{cLb_2})},\ 0,\ -\frac{w_2Lb_2}{2},\ \frac{w_2Lb_2^2}{12(1+\frac{2Eb_2Ib_2}{cLb_2})},\ f_1,\ -\frac{1}{2w_3Lb_3},$$
$$-\frac{w_3Lb_3^2}{12(1+\frac{2Eb_3Ib_3}{cLb_3})},\ 0,\ -\frac{w_3Lb_3}{2}-\frac{w_4Lb_4}{2},\ \frac{w_3Lb_3^2}{12(1+\frac{2Eb_3Ib_3}{cLb_3})}$$
$$\left.-\frac{w_4Lb_4^2}{12(1+\frac{2Eb_4Ib_4}{cLb_4})},\ 0,\ -\frac{w_4Lb_4}{2},\ \frac{w_4Lb_4^2}{12(1+\frac{2Eb_4Ib_4}{cLb_4})}\right)^\top.$$

The previous solver finds an enclosure for the solution set in about 755 s and thereby exhibits performance approximately three times slower than the new solver (about 245 s). Also, the quality of the solution enclosure provided by the new solver is much better than the solution enclosure provided by the previous solver; here, the relative overestimation ranges between 28.4 and 95.46.

5 Conclusions

In this chapter, we demonstrated the advanced application of a general-purpose parametric method, combined with the Bernstein enclosure of polynomial ranges, to linear systems obtained by standard FEM analysis of mechanical structures, and illustrated the efficiency of the new parametric solver. Further applications, viz. to truss structures with uncertain node locations, can be found in [38].

It is shown that powerful techniques for range enclosure are necessary to provide tight bounds on the solution set, in particular when the parameters of the system are subject to large uncertainties and the dependencies are complicated.

The new self-verified parametric solvers can be incorporated into a general framework for the computer-assisted proof of global and local monotonicity properties of the parametric solution. Based on these properties, a guaranteed and highly accurate enclosure of the interval hull of the solution set can be computed [13, 28, 39]. The parametric solver for square systems also facilitates the guaranteed enclosures of the solution sets to over- and underdetermined parametric linear systems [27].

Being presently the only general-purpose parametric linear solver, the presented methodology and software tools are applicable in the context of any problem (stemming, e.g., from fuzzy set theory [35] or the other fields listed in the Introduction) that requires the solution of linear systems whose input data depend on uncertain (interval) parameters.

Acknowledgements This work has been supported by the State of Baden-Württemberg, Germany.

References

1. Alefeld, G., Herzberger, J.: Introduction to Interval Computations. Academic, New York (1983)
2. Cargo, G.T., Shisha, O.: The Bernstein form of a polynomial. J. Res. Nat. Bur. Standards **70B**, 79–81 (1966)
3. Corliss, G., Foley, C., Kearfott, R.B.: Formulation for reliable analysis of structural frames. Reliab. Comput. **13**, 125–147 (2007)
4. Dessombz, O., Thouverez, F., Laîné, J.-P., Jézéquel, L.: Analysis of mechanical systems using interval computations applied to finite element methods. J. Sound Vibrat. **239**(5), 949–968 (2001)
5. Dreyer, A.: Interval Analysis of Analog Circuits with Component Tolerances. Shaker-Verlag, Aachen, Germany (2005)
6. Dreyer, A.: Interval methods for analog circuits. Report No. 97, Fraunhofer Institut für Techno- und Wirtschaftsmathematik, Kaiserslautern, Germany (2006)
7. European Standard: Eurocode 3: Design of Steel Structures. European Committee for Standardization, Ref.No. prEN 1993-1-1:2003 E, Brussels (2003)
8. Franzen, R.: Die intervallanalytische Behandlung parameterabhängiger Gleichungssysteme. Berichte der GMD, vol. 47, Bonn (1971)
9. Franzen, R.: Die Konstruktion eines Approximationspolynoms für die Lösungen parameterabhängiger Gleichungssysteme. Z. Angew. Math. Mech. **52**, T202–T204 (1972)

10. Ganesan, A., Ross, S.R., Barmish, B.R.: An extreme point result for convexity, concavity and monotonicity of parameterized linear equation solutions. Linear Algebra Appl. **390**, 61–73 (2004)
11. Garloff, J.: Convergent bounds for the range of multivariate polynomials. In: Nickel, K. (ed.) Interval Mathematics 1985, Lect. Notes in Comp. Sci., vol. 212, pp. 37–56. Springer, Berlin (1986)
12. Garloff, J., Popova, E.D., Smith, A.P.: Solving linear systems with polynomial parameter dependency. Preprint No.1/2009, Bulgarian Academy of Sciences, Institute of Mathematics and Informatics, Department of Biomathematics, Sofia (2009). http://www.math.bas.bg/~epopova/papers/09Preprint-GPS.pdf
13. Garloff, J., Smith, A.P., Werkle, H.: A verified monotonicity-based solution of a simple finite element model with uncertain node locations. Proc. Appl. Math. Mech. (PAMM) **10**, 157–158 (2010)
14. Hladík, M.: Enclosures for the solution set of parametric interval linear systems. KAM-DIAMATIA Series No. 2010–983, Department of Applied Mathematics, Charles University, Prague, Czech Republic (2010)
15. Kolev, L.V.: Improvement of a direct method for outer solution of linear parametric systems. Reliab. Comput. **12**, 193–202 (2006)
16. Krawczyk, R.: Newton-Algorithmen zur Bestimmung von Nullstellen mit Fehlerschranken. Computing **4**, 187–201 (1969)
17. Lagoa, C.M., Barmish, B.R.: Distributionally robust Monte Carlo simulation: A tutorial survey. In: Proceedings of the 15th IFAC World Congress, pp. 1327–1338 (2002). http://www.ece.lsu.edu/mcu/lawss/add_materials/BRossBarmishTutorial.pdf
18. Lerch, M., Tischler, G., Wolff von Gudenberg, J.: `filib++` – Interval library specification and reference manual. Technical Report 279, University of Würzburg (2001)
19. Lerch, M., Tischler, G., Wolff von Gudenberg, J., Hofschuster, W., Krämer, W.: `filib++`, a fast interval library supporting containment computations. ACM Trans. Math. Software **32**(2), 299–324 (2006). http://www2.math.uni-wuppertal.de/org/WRST/software/filib.html
20. Moore, R.E.: A test for existence of solutions to nonlinear systems. SIAM J. Numer. Anal. **14**, 611–615 (1977)
21. Muhanna, R.L., Zhang, H., Mullen, R.L.: Interval finite elements as a basis for generalized models of uncertainty in engineering mechanics. Reliab. Comput. **13**, 173–194 (2007)
22. Neumaier, A.: Interval Methods for Systems of Equations. Cambridge University Press, Cambridge (1990)
23. Popova, E.D.: Generalizing the parametric fixed-point iteration. Proc. Appl. Math. Mech. (PAMM) **4**(1), 680–681 (2004)
24. Popova, E.D.: Parametric interval linear solver. Numer. Algorithms **37**(1–4), 345–356 (2004)
25. Popova, E.D.: Strong regularity of parametric interval matrices. In: Dimovski, I., et al. (eds.) Mathematics & Education in Mathematics, pp. 446–451. IMI-BAS, Sofia (2004)
26. Popova, E.D.: Solving linear systems whose input data are rational functions of interval parameters. In: Boyanov, T., et al. (eds.) NMA 2006, Lect. Notes in Comp. Sci., vol. 4310, pp. 345–352. Springer, Berlin (2007). Extended version in: Preprint 3/2005, Institute of Mathematics and Informatics, BAS, Sofia (2005). http://www.math.bas.bg/~epopova/papers/05PreprintEP.pdf
27. Popova, E.D.: Improved solution enclosures for over- and underdetermined interval linear systems. In: Lirkov, I., Margenov, S., Waśniewski, J. (eds.) Large-Scale Scientific Computing 2005, Lect. Notes in Comp. Sci., vol. 3743, pp. 305–312. Springer, Berlin (2006)
28. Popova, E.D.: Computer-assisted proofs in solving linear parametric problems. In: Luther, W., Otten, W. (eds.) IEEE–Proceedings of SCAN 2006, 12th GAMM–IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics, Duisburg, Germany, pp. 35–43. IEEE Computer Society Press, Library of Congress Number 2007929345 (2007)

29. Popova, E.D., Iankov, R., Bonev, Z.: Bounding the response of mechanical structures with uncertainties in all the parameters. In: Muhanna, R.L., Mullen, R.L. (eds.) Proceedings of the NSF Workshop on Reliable Engineering Computing, Savannah, Georgia, pp. 245–265 (2006)
30. Popova, E.D., Iankov, R., Bonev, Z.: FEM model of a two-bay two-story steel frame – 2 benchmark examples (2009). http://www.math.bas.bg/~epopova/papers/2bay2storyProblems.pdf
31. Popova, E.D., Krämer, W.: Inner and outer bounds for the solution set of parametric linear systems. J. Comput. Appl. Math. **199**(2), 310–316 (2007)
32. Prautzsch, H., Boehm, W., Paluszny, M.: Bezier and B-Spline Techniques. Springer, Berlin (2002)
33. Rump, S.: New results on verified inclusions. In: Miranker, W.L., Toupin, R. (eds.) Accurate Scientific Computations, Lect. Notes in Comp. Sci., vol. 235, p. 31–69. Springer, Berlin (1986)
34. Rump, S.: Verification methods for dense and sparse systems of equations. In: Herzberger, J. (ed.) Topics in Validated Computations, pp. 63–135. North-Holland, Amsterdam (1994)
35. Śkalna, I.: Parametric fuzzy linear systems. In: Castillo, O., et al. (eds.) Theoretical Advances and Applications of Fuzzy Logic and Soft Computing, Advances in Soft Computing, vol. 42, pp. 556–564. Springer, Berlin (2007)
36. Śkalna, I.: Direct method for solving parametric interval linear systems with non-affine dependencies. In: Parallel Processing and Applied Mathematics, 8th International Conference (PPAM 2009), Wroclaw, Poland, Sept 13–16 2009. Revised Selected Papers, Part II, In: Wyrzykowski, R., Dongarra, J., Karczewski, K., Waśniewski, J.W. (eds.) Lect. Notes in Comp. Sci., vol. 6068, pp. 485–494 (2010)
37. Smith, A.P.: Fast construction of constant bound functions for sparse polynomials. J. Global Optim. **43**(2–3), 445–458 (2009)
38. Smith, A.P., Garloff, J., Werkle, H.: Verified solution for a simple truss structure with uncertain node locations. In: Gürlebeck, K., Könke, C. (eds.) Proceedings of the 18th International Conference on the Application of Computer Science and Mathematics in Architecture and Civil Engineering, Weimar, Germany (2009)
39. Smith, A.P., Garloff, J., Werkle, H.: A method for the verified solution of finite element models with uncertain node locations. To appear in 11th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP11), Zürich, Switzerland (2011)
40. Wolfram Research Inc.: Mathematica, Version 5.2. Champaign, Illinois (2005)
41. Zettler, M., Garloff, J.: Robustness analysis of polynomials with polynomial parameter dependency using Bernstein expansion. IEEE Trans. Automat. Contr. **43**, 425–431 (1998)
42. Zhang, H.: Nondeterministic Linear Static Finite Element Analysis: An Interval Approach. Ph.D. thesis, School of Civil and Environment Engineering, Georgia Institute of Technology (2005)

A Fast Block Krylov Implicit Runge–Kutta Method for Solving Large-Scale Ordinary Differential Equations

A. Bouhamidi and K. Jbilou

Abstract In this chapter, we describe a new based block Krylov–Runge–Kutta method for solving stiff ordinary differential equations. We transform the linear system arising in the application of Newton's method to a nonsymmetric matrix Stein equation that will be solved by a block Krylov iterative method. Numerical examples are given to illustrate the performance of our proposed method.

Key words Block Krylov • Newton method • ODE • Optimization • Runge–Kutta

1 Introduction

This chapter is concerned with the numerical solution of the following ODE problem:

$$\begin{cases} y'(t) = f(t, y(t)), t \in [t_0, T], \\ y(t_0) = y_0 \in \mathbb{R}^m, \end{cases} \tag{1}$$

where $y : [t_0, T] \to \mathbb{R}^m$. We assume that the function

$$f : [t_0, T] \times \mathbb{R}^m \longrightarrow \mathbb{R}^m$$

is continuous in $t \in [t_0, T]$ and Lipschitz continuous in $y \in \mathbb{R}^m$, i.e.,

$$||f(t, y) - f(t, z)|| \leq M||y - z||,$$

A. Bouhamidi • K. Jbilou (✉)
LMPA Universite du Littoral Cote d'Opale, 50 rue F. Buisson BP. 699,
F-62228 Calais Cedex, France
e-mail: bouhamidi@lmpa.univ-littoral.fr; jbilou@lmpa.univ-littoral.fr

A. Chinchuluun et al. (eds.), *Optimization, Simulation, and Control*, Springer Optimization and Its Applications 76, DOI 10.1007/978-1-4614-5131-0_20,

for some positive M. These conditions guarantee the existence and uniqueness of a solution y of (1). The numerical approximation of ODEs is still a very attractive problem. The most popular numerical methods for solving (1) are the well-known s-stage implicit Runge–Kutta (IRK) methods defined by the following relations; for more details (see [4, 12, 13]):

$$\mathbf{y}_i = y_n + h \sum_{j=1}^{s} a_{i,j} f(t_n + c_j h,\ \mathbf{y}_j), \quad i = 1, \ldots, s \tag{2}$$

and

$$y_{n+1} = y_n + h \sum_{j=1}^{s} b_j f(t_n + c_j h,\ \mathbf{y}_j), \tag{3}$$

where $t_n = t_0 + nh$, for $n = 0, \ldots, N$, is a discretization by the $N+1$ points $t_0, \ldots, t_N$ of the interval $[t_0, T]$ and $h = (T - t_0)/N$ is the stepsize. Here y_n is an approximation of $y(t_n)$ and the m-dimensional vectors $\mathbf{y}_i$ approximate $y(t_n + c_i h)$.
Let $b = (b_1, \ldots, b_s)^T$ $(s \geq 2)$ be the weight vector, $c = (c_1, \ldots, c_s)^T$ the node vector, and let $\widetilde{A} = (a_{ij})_{i,j=1,\ldots,s}$ be the IRK coefficient matrix. The vector c is such that $\widetilde{A}\, e = c$ with $e = (1, \ldots, 1)^T \in \mathbb{R}^s$.

The Runge–Kutta coefficients are usually given by the Butcher tableau as follows:

$$\begin{array}{c|c} c & \widetilde{A} \\ \hline & b^T \end{array} = \begin{array}{c|ccc} c_1 & a_{11} & \ldots & a_{1s} \\ \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & \ldots & a_{ss} \\ \hline & b_1 & \ldots & b_s \end{array}.$$

The main difficulty in the implementation of IRK methods (for stiff problems) is to efficiently solve nonlinear system of equations. These nonlinear systems could be solved by using Newton-type methods, and this requires high computation and cpu time when m is large. For medium problems, some IRK schemes have been developed to reduce these costs; see [7, 10, 14, 20, 21]. For large problems, Krylov subspace methods such as the GMRES algorithm [17] could also be used; see [3].

In this chapter, we will exploit the special structure of the matrices of the linear system arising at each step of the Newton method. Hence, using some properties of the Kronecker product, the linear systems are transformed into Stein matrix equations that will be solved by an extended block Arnoldi (EBA) method.

Let $A = (a_{ij})$ and $B = (b_{ij})$ be $m \times s$ and $n \times q$ matrices, respectively. The Kronecker product of the matrices A and B is defined as the $mn \times sq$ matrix $A \otimes B = (a_{ij}B)$. The *vec* operator transforms a matrix A of size $m \times s$ to a vector $a = vec(A)$ of size $ms \times 1$ by stacking the columns of A. Some properties of the Kronecker product are given in [16].

This chapter is organized as follows. In Sect. 2, we describe the IRK methods and the connection with Stein matrix equations. To solve these matrix equations, a numerical method based on a block Arnoldi-type method is given in Sect. 3. The last section is devoted to some numerical examples.

2 Implicit Runge–Kutta Methods and Stein Matrix Equations

Using tensor notations, the one-step IRK methods (2)–(3) can be written as

$$\mathbf{y} = (e \otimes y_n) + h(\widetilde{A} \otimes I_m)F(t_n, \mathbf{y}), \tag{4}$$

$$y_{n+1} = y_n + h(b \otimes I_m)F(t_n, \mathbf{y}), \tag{5}$$

where I_m is the $m \times m$ identity matrix, $\mathbf{y} = (\mathbf{y}_1^T, \ldots, \mathbf{y}_s^T)^T \in \mathbb{R}^{ms}$ is the stage vector, and the function $F : [t_0, T] \times \mathbb{R}^{ms} \to \mathbb{R}^{ms}$ is given by

$$F(t, \mathbf{y}) = (f(t_n + c_1 h, \mathbf{y}_1)^T, \ldots, f(t_n + c_s h, \mathbf{y}_s)^T)^T. \tag{6}$$

By solving the nonlinear system given in (4) we obtain $\mathbf{y}$ and we compute y_{n+1} from (5).

Let $R_n : \mathbb{R}^{ms} \to \mathbb{R}^{ms}$ be the function defined by

$$R_n(\mathbf{y}) = -\mathbf{y} + (e \otimes y_n) + h(\widetilde{A} \otimes I_m)F(t_n, \mathbf{y}). \tag{7}$$

Hence, (4) is equivalent the following nonlinear system of equation:

$$R_n(\mathbf{y}) = \mathbf{0}. \tag{8}$$

Therefore, applying Newton's method to the Eq. (8), we get the iterations

$$\begin{cases} \mathbf{y}^{(0)} \text{ initial guess} \\ \mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} - [J_{R_n}(\mathbf{y}^{(k)})]^{-1} R_n(\mathbf{y}^{(k)}), k = 0, 1, \ldots, \end{cases} \tag{9}$$

where $J_{R_n}(\mathbf{y})$ is the Jacobian matrix of size $ms \times ms$ of the function R_n evaluated at $\mathbf{y}$. Hence, an easy computation gives

$$J_{R_n}(\mathbf{y}) = -I_{ms} + h(\widetilde{A} \otimes I_m)J_F(t_n, \mathbf{y}),$$

where $J_F(t_n, \mathbf{y})$ is the Jacobian matrix of size $ms \times ms$ of the function F evaluated at $(t_n, \mathbf{y})$. It is also easy to obtain

$$J_F(t_n, \mathbf{y}) = \text{diag}[J_f(t_n + c_1 h, \mathbf{y}_1), \ldots, J_f(t_n + c_s h, \mathbf{y}_s)],$$

where $J_f(t, y) = \left[\frac{\partial f_i}{\partial y_j}(t, y)\right]_{1 \le i,j \le m}$ with $f = (f_1, \ldots, f_m)^T$ is the Jacobian matrix of size $m \times m$ of the function f.

In practical computations, the numerical value of $J_f(t_n + c_i h, \mathbf{y}_i)$, $(i = 1, \ldots, s)$ changes very slowly during the Newton iterations, and then there is no need to reevaluate it. We will assume that $J_f(t_n + c_i h, \mathbf{y}_i)$ needs to be evaluated once at

each step. Thus the same J_n approximation will be used for every stage and over all the iterations. Therefore, the Jacobian matrix $J_F(t_n,\ \mathbf{y})$ can be approximated by

$$\widehat{J_n} = \text{diag}[\underbrace{J_n, \ldots, J_n}_{s \text{ times}}] = I_s \otimes J_n$$

and the Jacobian matrix $J_{R_n}(\ \mathbf{y})$ by

$$\widehat{J}_{R_n} = -I_{ms} + h(\widetilde{A} \otimes I_m)(I_s \otimes J_n).$$

Using the property $(A \otimes B)(C \otimes D) = AC \otimes BD$ for appropriate sizes of the matrices A, B, C, and D, we obtain

$$\widehat{J}_{R_n} = -I_{ms} + \widetilde{A} \otimes (hJ_n). \tag{10}$$

Now, the Newton scheme (9) can be replaced by the modified one

$$\begin{cases} \mathbf{y}^{(0)} \text{ initial guess} \\ \mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} - \widehat{J}_{R_n}^{-1} R_n(\mathbf{y}^{(k)}), k = 0, 1, \ldots. \end{cases} \tag{11}$$

The vector $\mathbf{x}^{(k)} = \widehat{J}_{R_n}^{-1} R_n(\mathbf{y}^{(k)})$ is obtained as the solution of the following $ms \times ms$ linear system:

$$\left[\widetilde{A} \otimes (hJ_n) - I_{ms}\right] \mathbf{x}^{(k)} = R_n(\mathbf{y}^{(k)}). \tag{12}$$

Let $\mathscr{F}$ be the matrix-mapping $\mathscr{F} : [t_0, T] \times \mathbb{R}^{m \times s} \to \mathbb{R}^{m \times s}$ derived from F_n by

$$\mathscr{F}(t, Y) = [f(t + c_1 h,\ \mathbf{y}_1), \ldots, f(t + c_s h,\ \mathbf{y}_s)] \in \mathbb{R}^{m \times s},$$

and let $\mathscr{R}_n : \mathbb{R}^{m \times s} \to \mathbb{R}^{m \times s}$ be the residual defined by

$$\mathscr{R}_n(Y) = -Y + y_n \mathbf{e}^T + h\mathscr{F}(t_n, Y)\widetilde{A}^T.$$

Then it is easy to see that for any vector $y \in \mathbb{R}^{ms}$ and a matrix $Y \in \mathbb{R}^{m \times s}$ with $y = vec(Y)$, we have $R_n(y) = vec(\mathscr{R}_n(Y))$. Let $X^{(k)}$ and $Y^{(k)}$ be the $m \times s$ matrices such that $\mathbf{x}^{(k)} = vec(X^{(k)})$ and $\mathbf{y}^{(k)} = vec(Y^{(k)})$. Then using the property

$$vec(AXB) = (B^T \otimes A)\, vec(X), \tag{13}$$

the linear system (12) can be transformed to the following Stein matrix equation:

$$(hJ_n)X^{(k)}\widetilde{A}^T - X^{(k)} = \mathscr{R}_n(Y^{(k)}). \tag{14}$$

Therefore from the relations (11), the approximations $Y^{(k)}$ are derived from the iterations

$$\begin{cases} Y^{(0)} \quad \text{initial matrix guess} \\ Y^{(k+1)} = Y^{(k)} - X^{(k)}, k = 0, 1, \ldots, kmax, \end{cases} \tag{15}$$

where $X^{(k)}$ is the solution of the Stein matrix equation (14). Using again (13), the relation (5) can also be given as

$$y_{n+1} = y_n + h\mathscr{F}(t_n, \widetilde{Y})b,$$

where $\widetilde{Y} = Y^{(kmax)}$ is obtained from the iterations (15).

For small problems, the nonsymmetric Stein matrix equation (14) can be solved by direct methods or by transforming this matrix equation to a linear system using properties of the Kronecker product. For large problems, these two approaches are very expensive. In this case, we will propose an iterative projection method for solving these nonsymmetric Stein matrix equations.

3 A Numerical Method for Solving Large Nonsymmetric Stein Matrix Equations

In this section we will propose and study a numerical method for solving the following Stein matrix equation:

$$AXB - X = C, \tag{16}$$

where $A \in \mathbb{R}^{m\times m}$, $B \in \mathbb{R}^{s\times s}$, $C \in \mathbb{R}^{m\times s}$, and $X \in \mathbb{R}^{m\times s}$, where the integer m is large and the integer s is of moderate size. We also assume here that B is nonsingular.

The matrix equation (16) plays an important role in linear control and filtering theory for discrete-time large-scale dynamical systems and other problems; see [5, 6, 11, 15, 19] and the references therein. It also appears in image restoration techniques [2] and in each step of Newton's method for discrete-time algebraic Riccati equations [16].

When the matrices are of small sizes, direct methods based on the Schur decomposition could be used for solving the matrix equation (16); see [1]. Notice also that by using (13), the matrix equation (16) can be formulated as the following $ms \times ms$ linear system of equations:

$$(A \otimes B^T - I_{ms})\, vec(X) = vec(C). \tag{17}$$

Krylov subspace methods such as the GMRES algorithm [17] could be used to solve the linear system (17). It is known that the matrix equation (16) has a unique solution if and only if $\lambda_i(A)\lambda_j(B) \neq 1$ for all $i = 1, \ldots, m$; $j = 1, \ldots, s$ where $\lambda_i(A)$ is the ith eigenvalue of the matrix A. This will be assumed through this chapter.

We present here a Galerkin projection method based on the EBA algorithm [18]. We consider the case where the $m \times s$ matrix C is of full rank and $s \ll m$.

Algorithm 1 The EBA Algorithm

1. Inputs: A an $m \times m$ matrix, V an $m \times s$ matrix and k an integer.
2. Compute the QR decomposition of $[V, A^{-1}V] = V_1 \Lambda$, where V_1 is orthogonal and Λ upper triangular.
3. Set $\mathcal{V}_0 = [\,]$.
4. For $j = 1, \ldots, k$
   - Set: $V_j^{(1)}$ the first s columns of V_j.
   - Set: $V_j^{(2)}$ the second s columns of V_j.
   - $\mathcal{V}_j = [\mathcal{V}_{j-1}, V_j]$; $\widehat{V}_{j+1} = \left[A V_j^{(1)}, A^{-1} V_j^{(2)}\right]$.
   - Orthogonalize $\widehat{V}_{j+1}$ w.r. to $\mathcal{V}_j$ to get V_{j+1}:
   - for $i = 1, 2, \ldots, j$
     $H_{i,j} = V_i^T \widehat{V}_{j+1}$.
     $\widehat{V}_{j+1} = \widehat{V}_{j+1} - V_i H_{i,j}$.
   - endfor;
5. Compute the QR decomposition of $\widehat{V}_{j+1} = V_{j+1} H_{j+1,j}$.
6. EndFor.

3.1 The Extended Block Arnoldi Algorithm

We first recall the EBA process applied to the pair (A, V) where $A \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times s}$. The projection subspace $\mathcal{K}_k(A, V)$ of $\mathbb{R}^m$ that we will consider was introduced in [8, 18]:

$$\mathcal{K}_k(A, V) = \text{Range}([V, A^{-1}V, AV, A^{-2}V, A^2V, \ldots, A^{-(k-1)}V, A^{k-1}V]).$$

Note that the subspace $\mathcal{K}_k(A, V)$ is a sum of two block Krylov subspaces

$$\mathcal{K}_k(A, V) = \mathbb{K}_k(A, V) + \mathbb{K}_k(A^{-1}, A^{-1}V)$$

where $\mathbb{K}_k(A, V) = \text{Range}([V, AV, \ldots, A^{k-1}V])$. The following algorithm allows us to compute an orthonormal basis of the extended Krylov subspace $\mathcal{K}_k(A, V)$. This basis contains information on both A and A^{-1}. The EBA process is described as follows:

Since the above algorithm involves implicitly a Gram–Schmidt process, the obtained block vectors $\mathcal{V}_k = [V_1, V_2, \ldots, V_k]$ ($V_i \in \mathbb{R}^{m \times 2s}$) have their columns mutually orthogonal provided none of the upper triangular matrices $H_{j+1,j}$ are rank deficient. Hence, after m steps, Algorithm 1 builds an orthonormal basis $\mathcal{V}_k$ of the Krylov subspace $\mathcal{K}_k(A, V)$ and a block upper Hessenberg matrix H_k whose nonzero blocks are the $H_{i,j}$. Note that each submatrix $H_{i,j}$ ($1 \leq i \leq j \leq k$) is of order $2s$.

Let $\mathcal{T}_k \in \mathbb{R}^{2ks \times 2ks}$ be the restriction of the matrix A to the extended Krylov subspace $\mathcal{K}_k(A, V)$, i.e., $\mathcal{T}_k = \mathcal{V}_k^T F \mathcal{V}_k$. It is shown in [18] that $\mathcal{T}_k$ is also block upper Hessenberg with $2s \times 2s$ blocks. Moreover, a recursion is derived to compute

$\mathscr{T}_k$ from H_k without requiring matrix-vector products with A. For more details, on how to compute $\mathscr{T}_k$ from H_k, we refer to [18]. We note that for large problems, the inverse of the matrix A is not computed explicitly, and in this case we can use iterative solvers with preconditioners to solve linear systems with A. However, when these linear systems are note solved accurately, the theoretical properties of the EBA process are no longer valid.

Next, we give some properties that will be useful later. Let $\bar{\mathscr{T}}_k = \mathscr{V}_{k+1}^T A \mathscr{V}_k$, and suppose that k steps of Algorithm 1 have been run, then we have

$$A\mathscr{V}_k = \mathscr{V}_{k+1}\bar{\mathscr{T}}_k, \tag{18}$$

$$= \mathscr{V}_k \mathscr{T}_k + V_{k+1} T_{k+1,k} E_k^T, \tag{19}$$

where $T_{i,j}$ is the $2s \times 2s$, (i,j)-block of $\mathscr{T}_k$, and $E_k = [O_{2s\times 2(k-1)s}, I_{2s}]^T$ is the matrix of the last $2s$ columns of the $2ks \times 2ks$ identity matrix I_{2ks}.

3.2 The Extended Block Arnoldi Algorithm for Stein Equations

In this section, we will apply the EBA algorithm to get approximate solutions to the Stein matrix equation (16). We project the Stein equation (16) onto an extended block Krylov subspace and then solve, at each iteration, the obtained low-dimensional equation.

Let $\mathscr{A}$ be the linear operator from $\mathbb{R}^{m\times s}$ onto $\mathbb{R}^{m\times s}$ defined as follows:

$$\mathscr{A}: \; X \longrightarrow \mathscr{A}(X) = AXB - X. \tag{20}$$

Then the Stein equation (16) can be written as

$$\mathscr{A}(X) = C. \tag{21}$$

We will solve the problem (21) which is equivalent to the initial problem (16).

Let X_0 be an initial guess, and set $R_0 = C - AX_0B + X_0$, the extended block Arnoldi Stein method constructs, at step k, the new approximation X_k as follows:

$$X_k^{(i)} - X_0^{(i)} = Z_k^{(i)} \in \mathscr{K}_k(\mathscr{A}, R_0); i = 1, \ldots, s, \tag{22}$$

with the orthogonality relation

$$R_k^{(i)} \perp \mathscr{K}_k(\mathscr{A}, R_0); i = 1, \ldots, s, \tag{23}$$

where $R_k^{(i)}$ is the ith component of the residual $R_k = C - \mathscr{A}(X_k)$ and $X_k^{(i)}$ is the ith column of X_k. We give the following result which is easy to prove [9].

Theorem 1. *Let $\mathscr{A}$ be the operator defined by (20), then*

$$\mathscr{K}_k(\mathscr{A}, R_0) = \mathscr{K}_k(A, R_0).$$

Using this last property, the relations (22) and (23) are written as

$$X_k^{(i)} - X_0^{(i)} = Z_k^{(i)} \in \mathscr{K}_k(A, R_0), \tag{24}$$

and

$$R_k^{(i)} \perp \mathscr{K}_k(A, R_0); i = 1, \ldots, s. \tag{25}$$

Assume that R_0 is of rank s and let $[R_0, A^{-1}R_0] = V_1 U_1$ (the QR decomposition of $[R_0, A^{-1}R_0]$ where the $m \times 2s$ matrix V_1 is orthogonal and U_1 is $2s \times 2s$ upper triangular).
Now as the columns of the matrix $\mathscr{V}_k$ (constructed by the EBA algorithm) form a basis of the extended block Krylov subspace $\mathscr{K}_k(A, R_0)$, the relation (24) implies that $X_k = X_0 + \mathscr{V}_k Y_k$ where Y_k is a $2ks \times s$ matrix. The relation (25) implies that

$$\mathscr{V}_k^T (R_0 - A \mathscr{V}_k Y_k B + \mathscr{V}_k Y_k) = 0.$$

Therefore, using (19), and the fact that $\mathscr{V}_k$ is orthonormal, we obtain the low-dimensional Stein equation

$$\mathscr{T}_k Y_k B - Y_k = \tilde{C} \tag{26}$$

with $\tilde{C} = \tilde{E}_1 U_{1,1}$ where $\tilde{E}_1$ is the $2ks \times s$ matrix whose upper $s \times s$ principal block is the identity matrix I_{2ks} and $U_{1,1}$ is the first $s \times s$ block of U_1.

The matrix equation (26) will be solved by using a direct method such as the Hessenberg–Schur method [6]. We assume that during the iterations $\lambda_i(\mathscr{T}_k)\lambda_j(B) < 1$, and this implies that the Eq. (26) has a unique solution.

The next result allows us to compute the norm of the residual (at each iteration) without computing the residual. This will be used to stop the iterations in the EBA Stein algorithm without having to compute an extra product with the matrix A which reduces the cost and storage for large problems.

Theorem 2. *The norm of the residual R_k is given by*

$$\begin{aligned} \| R_k \|_F &= \| T_{k+1,k} E_k^T Y_k B \|_F \\ &= \| T_{k+1,k} \tilde{Y}_k B \|_F, \end{aligned}$$

where $\tilde{Y}_k$ is the $2s \times s$ matrix corresponding to the last s rows of the matrix Y_k.

The proof is easily obtained by using the relation (26) and the fact that the matrix V_{k+1} is orthogonal. The EBA algorithm for solving (16) is summarized as follows:

Algorithm 2 The EBA Algorithm for Stein Equations

1. Choose a tolerance tol, an initial guess X_0; an integer $kmax$ and set $k = 1$.
2. Compute $R_0 = C + X_0 - AX_0B$.
3. Compute the QR decomposition: $R_0 = V_1U_1$.
4. While $k < kmax$ and $||R_k||_F > tol$ do
   - Apply Algorithm 1 to the pair (A, V_1) to generate the blocks $V_1, \ldots, V_{k+1}$; and the block Hessenberg matrix $\mathscr{T}_k$.
   - Solve by a direct method (the Schur method) the low-order Stein equation $\mathscr{T}_k Z B - Z = \tilde{C}$.
   - Compute $X = X_0 + \mathscr{V}_k Z$.
   - Compute $\| R_k \|_F$, by using Theorem 2.
   - Set $k = k + 1$ and go to step 4.
5. End.

To save CPU time and memory requirements, Algorithm 1 will be used in a restarted mode. This means that we restart the algorithm every k_1 iterations were k_1 is a fixed integer.

4 Numerical Examples

The numerical experiments were performed in Matlab 7.0.4, on Windows XP system running on Intel(R) Core(TM) 2 Duo CPU 3.00 GHz with 3.23 GB RAM. In our experiments, a maximum number of 30 iterations was allowed for both the EBA and for the Newton method.

We used the 3-stage RADAU-IIA method of order $p = 2s - 1 = 5$ which the corresponding Butcher tableau is given by

$$\begin{array}{c|ccc} \frac{4-\sqrt{6}}{10} & \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\ \frac{4+\sqrt{6}}{10} & \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\ 1 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \\ \hline & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \end{array}.$$

As a numerical example, we consider the following heat equation:

$$\begin{cases} \dfrac{\partial u(t,x)}{\partial t} = c^2 \dfrac{\partial^2 u(t,x)}{\partial x^2} + g(t,x), \quad (t,x) \in [t_0, T] \times [\alpha, \beta] \\ u(t,\alpha) = u(t,\beta) = 0,\ t \in [t_0, T], \\ u(t_0,x) = y_0(x),\ x \in [\alpha, \beta]. \end{cases}$$

Let $y_i(t)$ be the approximation of the exact value $u(t,x_i)$, and replace $\dfrac{\partial^2 u(t,x_i)}{\partial x^2}$ with the approximation

$$\frac{\partial^2 u(t,x_i)}{\partial x^2} \simeq [u(t,x_i+k) - 2u(t,x_i) + u(t,x_i-k)]/k^2,$$

where $x_i = x_{i-1} + k$, the parameter k is the stepsize on the x-axis, $k = (\beta-\alpha)/(m+1)$, $m \in \mathbb{N}$, with $x_0 = \alpha$ and $x_{m+1} = \beta$. The vector $y(t) = (y_1(t), \ldots, y_m(t))^T$ is the exact solution of the following problem:

$$\begin{cases} y'(t) = f(t,y(t)),\ t \in [t_0, T] \\ y(t_0) = y_0, \end{cases}$$

with

$$\begin{aligned} f(t,y) &= \widetilde{A}y + \mathbf{g}(t), \\ \widetilde{A} &= c^2 \frac{(m+1)^2}{(\beta-\alpha)^2} \operatorname{tridiag}(1,-2,1), \\ \mathbf{g}(t) &= (g(t,x_1), \ldots, g(t,x_m))^T, \end{aligned}$$

and the vector $\mathbf{y}_0 = (y_0(x_1), \ldots, y_0(x_m))^T$.

The eigenvalues of the matrix $\widetilde{A}$ are $\lambda_i = -4c^2 \frac{(m+1)^2}{(\beta-\alpha)^2} \sin^2(\frac{i\pi}{2(m+1)})$, for $i = 1, \ldots, m$.
When m increases, the stiff ratio also increases. The stiffness of the problem is due to the distribution of the eigenvalues λ_i, and the stiff ratio S_R is given by

$$S_R = \frac{\max_{1 \le i \le m} |\lambda_i|}{\min_{1 \le i \le m} |\lambda_i|}.$$

As an example, we consider the case where $c = \dfrac{\beta-\alpha}{\pi}$,

$$g(t,x) = \mathrm{e}^{-t} \sin\left(\frac{\pi(x-\alpha)}{\beta-\alpha}\right) \sin(t-t_0),$$

and $y_0(x) = 0$. The exact solution is given by

$$u(t,x) = \mathrm{e}^{-t} \sin\left(\frac{\pi(x-\alpha)}{\beta-\alpha}\right) (1 - \cos(t-t_0)).$$

In Table 1, we listed the results for the first experiment. We used different values of the dimension m, and we reported the relative error norms and the required CPU time for each value of m. The relative error norm is given by $||Y - \overline{Y}||_F / ||\overline{Y}||_F$ where $\overline{Y}$ is the matrix whose columns are the exact vector solutions and Y is the matrix whose columns are the computed approximate vector solutions.

Table 1 Results for experiment 1

Dimension	Time (s)	Relative error
1,000	41	80e−007
2,000	246	5.62e−008
3,000	503	2.50e−008

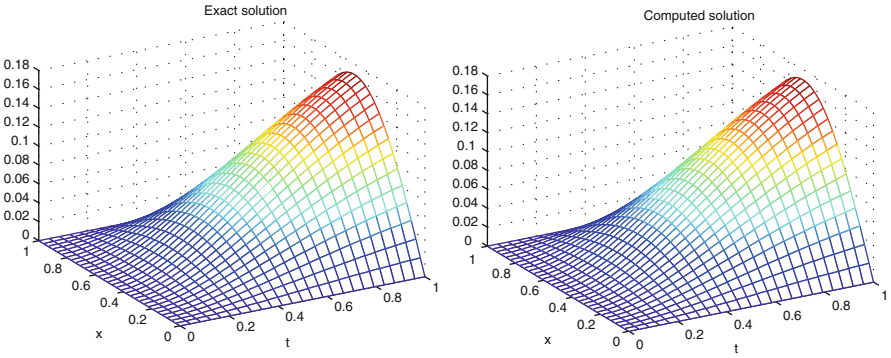


Fig. 1 Exact solution (*left*) and computed solution (*right*)

For the second test, we set $m = 25$ and $N = 30$. The stiff ratio is $S_R \simeq 3.8 \times 10^2$, and the error is $||Y - \overline{Y}||_F / ||\overline{Y}||_F \simeq 4.02 \times 10^{-6}$. The exact solution $u(t,x)$ is given in the left of Fig. 1, and the computed approximation is in the right of Fig. 1.

5 Summary

In this chapter, we proposed a numerical method for solving stiff ordinary differential equations in large dimensional spaces. The Newton method was applied to solve the derived nonlinear systems, and this requires, at each iteration of the Newton method, the computation of the solution of large special linear systems. These linear systems were transformed to nonsymmetric Stein matrix equations. Then, we used an EBA method to obtain approximate solutions to these matrix equations. We finally gave some numerical tests with relatively large problems.

References

1. A. Y. Barraud. *A* numerical algorithm to solve $A^T XA - X = Q$, IEEE Trans. Autom. Contr., AC-22: 883–885, (1977).
2. A. Bouhamidi and K. Jbilou: *Sylvester Tikhonov-regularization methods in image restoration*, J. Comput. Appl. Math., 206(1):86–98, (2007).
3. P. N. Brown, A. C. Hindmarsh, and L. R. Petzold: *Using Krylov methods in the solution of large-scale differential-algebraic systems*, SIAM J. Sci. Comput. 15: 1467–1488, (1994).

4. J.C. Butcher: *The Numerical Analysis of Ordinary Differential Equations*, Wiley, Chichester, 1987.
5. B.N. Datta: *Krylov-subspace methods for large scale matrix problems in control*, Generation of Computer Systems, 19: 125–126, (2003).
6. B.N. Datta: *Numerical Methods for Linear Control Systems*, Elsevier Academic press, 2004.
7. K. Dekker, *Partitioned Krylov subspace iteration in implicit Runge–Kutta methods*, Linear Algebra Appl. 431:488–494, (2009)
8. V. Druskin, L. Knizhnerman, *Extended Krylov subspaces: approximation of the matrix square root and related functions*, SIAM J. Matrix Anal. Appl., 19(3):755–771, (1998).
9. A. El Guennouni, K. Jbilou and A.J. Riquet , *Block Krylov subspace methods for solving large Sylvester equations*, Numer. Alg., 29: 75–96, (2002).
10. C. W. Gear, *Simultaneous numerical solutions of differential-algebraic equations*, IEEE Trans. Circuit Theory, CT-18, 1: 89–95, (1971).
11. K. Glover, D.J.N. Limebeer, J.C. Doyle, E.M. Kasenally and M.G. Safonov, *A characterisation of all solutions to the four block general distance problem*, SIAM J. Control Optim., 29: 283–324, (1991).
12. E. Hairer, S. P. Nørsett and G. Wanner, *Solving ordinary differential equations I. Nonstiff Problems, 2nd Revised Editions*, Springer Series in computational Mathematics, Vol. 8, Springer-Verlag, Berlin, 1993.
13. E. Hairer, and G. Wanner, *Solving ordinary differential equations II. Stiff and differential algebraic problems, 2nd Revised Editions*, Comput. Math., Vol. 14, Springer-Verlag, Berlin, 1996.
14. L. O. Jay, *Inexact simplified Newton iterations for implicit Runge–Kutta methods*, SIAM J. Numer. Anal. 38: 1369–1388, (2000).
15. K. Jbilou A. Messaoudi H. Sadok, *Global FOM and GMRES algorithms for matrix equations*, Appl. Num. Math., Appl. Num. math., 31: 49–63, (1999).
16. P. Lancaster, L. Rodman, *Algebraic Riccati Equations*, Clarendon Press, Oxford, 1995.
17. Y. Saad and M.H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statis. Comput., 7:856–869, (1986).
18. V. Simoncini, *A new iterative method for solving large-scale Lyapunov matrix equations*, SIAM J. Sci. Comp., 29(3):1268–1288, (2007).
19. P. Van Dooren, *Gramian based model reduction of large-scale dynamical systems*, in Numerical Analysis, Chapman and Hall, pp. 231–247, CRC Press London, 2000.
20. D. Voss, S. Abbas: *Block predictorcorrector schemes for the parallel solution of ODEs*, Comp. Math. Appl. 33: 65–72, (1997).
21. D. Voss, P.H. Muir, *Mono-implicit Runge-Kutta schemes for of initial value ODES the parallel solution*, J. Comp. Appl. Math. 102: 235–252, (1999).

Semilocal Convergence with R-Order Three Theorems for the Chebyshev Method and Its Modifications

Zhanlav Tugal and Khongorzul Dorjgotov

Abstract In this chapter we consider some modifications of the Chebyshev method that are free from second derivative and prove semilocal convergence theorems for these modifications as well as for the Chebyshev method. These two modifications can be considered as a generalization of some well-known iterative methods.

Key words Chebyshev method • Convergence • Nonlinear equations

1 Introduction

As is known, the higher order methods, such as Halley and Chebyshev methods play an important role in the solution of nonlinear equations. Especially they can be used in problems, where a quick convergence is required, such as stiff systems [11] and bifurcation problems [13]. However, they are not used often in practice due to their operational cost. For instance, in the iterative third-order methods, the main problem is to evaluate the second derivative of the operator. To overcome this difficulty, in the past years appeared many (multipoint) iterative methods [5–7, 12, 15] free from second derivative but with the same order of convergence. As a result, the operational cost is reduced to that of a second-order iterations, such as Newton's method.

In this chapter we propose some new modifications (multipoint iterations) of the Chebyshev method which are free from second derivative (Sect. 2). In Sects. 3–5 we analyze the convergence of the Chebyshev method and its two modifications,

Z. Tugal (✉) • K. Dorjgotov
School of Mathematics and Computer Science, National University of Mongolia, Ulaanbaatar, Mongolia
e-mail: tzhanlav@yahoo.com

A. Chinchuluun et al. (eds.), *Optimization, Simulation, and Control*, Springer Optimization and Its Applications 76, DOI 10.1007/978-1-4614-5131-0_21,

respectively, by using a technique consisting of a new system of real sequences [2,8]. In Sect. 6, we give mild convergence conditions for these methods. In the last Sect. 7, we present numerical results.

2 Some Modifications of Chebyshev Method

We consider a nonlinear equation

$$F(x) = 0. \tag{1}$$

Here $F : \Omega \subseteq X \to Y$ is a nonlinear Frechet twice differentiable operator defined on a convex, nonempty domain Ω, and X, Y are Banach spaces. The well-known Chebyshev method for solving the nonlinear equation (1) is given by [7]:

$$\begin{aligned} y_n &= x_n - \Gamma_n F(x_n), \qquad \Gamma_n = F'(x_n)^{-1}, \\ x_{n+1} &= y_n - \frac{1}{2}\Gamma_n F''(x_n)(y_n - x_n)^2, \quad n = 0, 1, \ldots. \end{aligned} \tag{2}$$

As in scalar cases [15] we can take next approximations

$$\begin{aligned} \frac{1}{2}F''(x_n)(y_n - x_n)^2 &\approx \frac{1}{2\theta}(F'(z_n) - F'(x_n))(y_n - x_n), \\ z_n &= (1-\theta)x_n + \theta y_n \quad 0 < \theta \leq 1 \end{aligned}$$

and

$$\frac{1}{2}F''(x_n)(y_n - x_n)^2 \approx \left(1 + \frac{b}{2}\right)F(y_n) + bF(x_n) - \frac{b}{2}F(z_n),$$

where

$$z_n = x_n + \Gamma_n F(x_n), \qquad -2 \leq b \leq 0.$$

As a consequence, we define the following new modifications:

$$\begin{aligned} y_n &= x_n - \Gamma_n F(x_n) \\ z_n &= (1-\theta)x_n + \theta y_n, \quad \theta \in (0, 1] \\ x_{n+1} &= y_n - \frac{1}{2\theta}\Gamma_n(F'(z_n) - F'(x_n))(y_n - x_n) \end{aligned} \tag{3}$$

and

$$\begin{aligned} y_n &= x_n - \Gamma_n F(x_n) \\ z_n &= x_n + \Gamma_n F(x_n), \\ x_{n+1} &= y_n - \Gamma_n\left(\left(1 + \frac{b}{2}\right)F(y_n) + bF(x_n) - \frac{b}{2}F(z_n)\right), \\ & -2 \leq b \leq 0. \end{aligned} \tag{4}$$

Thus we have classes of new two-and three-point iterative processes (3) and (4). It should be pointed out that such iterations (3) and (4) were given in [15] for functions of one variable.

In [5, 6] it was suggested a uniparametric Halley-type iterations with free from second derivative of the form

$$\begin{aligned} y_n &= x_n - \Gamma_n F(x_n) \\ z_n &= (1-\theta)x_n + \theta y_n, \quad \theta \in (0,1] \\ H(x_n,y_n) &= \frac{1}{\theta}\Gamma_n(F'(z_n) - F'(x_n)) \\ x_{n+1} &= y_n - \frac{1}{2}H(x_n,y_n)\left[I + \frac{1}{2}H(x_n,y_n)\right]^{-1}(y_n - x_n), \quad n \geq 0 \end{aligned} \tag{5}$$

and proved order three convergence of (5), as Halley method. If we take the approximation

$$\left[I + \frac{1}{2}H(x_n,y_n)\right]^{-1} \approx I$$

in (5), then (5) leads to (3). In this sense our modification (3) is easier than (5). It also should be pointed out that the iteration (3) with $\theta = 1/2$ and $\theta = 1$ was given in [7] and [1], respectively, and proven order three convergence under some restrictions. The iterations (4) can be considered as a generalization of some well-known iterations for function of one variable. For instance, if $b = -2$ the iteration (4) leads to two-point one with third-order convergence, suggested by Kou et al. [10]. If $b = 0$ the iteration (4) leads to also two-point one with third-order convergence that was suggested by Potra and Ptak [9, 12] and CL2 method [1]. From (3) and (4) it is clear that the modification (4) is preferable to (3), especially for the system of nonlinear equations, because in (3) the matrix-vector multiplication is needed in each iteration.

3 Recurrence Relations

In [14] we reduced the two-dimensional cubic decreasing region into one-dimensional region for the Chebyshev method. Now we will study the convergence of Chebyshev method (2) in detail. We assume that $\Gamma_0 \in \mathbf{L}(Y,X)$ exists at some $x_0 \in \Omega$, where $\mathbf{L}(Y,X)$ is a set of bounded linear operators from Y into X. In what follows we assume that

$$\begin{aligned} &(c_1) \quad \|F''(x)\| \leq M, \qquad x \in \Omega, \\ &(c_2) \quad \|y_0 - x_0\| = \|\Gamma_0 F(x_0)\| \leq \eta, \\ &(c_3) \quad \|\Gamma_0\| \leq \beta, \\ &(c_4) \quad \|F''(x) - F''(y)\| \leq K\|x-y\|, \qquad x,y \in \Omega, \qquad K > 0. \end{aligned}$$

Let us suppose that

$$a_0 = M\beta\eta \tag{6}$$

and define the sequence

$$a_{n+1} = f(a_n)^2 g(a_n) a_n, \tag{7}$$

where

$$f(x) = \frac{2}{2-2x-x^2}, \qquad g(x) = \frac{x^2(4+x)}{8} d, \tag{8}$$

and $d = 1+2\omega$, $\omega = \frac{K}{M^2 m}$, $m = \min_n \|\Gamma_n\| > 0$. In Sect. 4, we will show that $m > 0$.

Lemma 1. *Let f, g be two real functions given in (8). Then*
(i) f is increasing and $f(x) > 1$ for $x \in (0, \frac{1}{2})$.
(ii) g is increasing in $(0, \frac{1}{2})$.
(iii) $f(\gamma x) < f(x)$, $g(\gamma x) \le \gamma^2 g(x)$ for $x \in (0, \frac{1}{2})$ and $\gamma \in (0,1)$.

The proof is trivial [8].

Lemma 2. *Let $0 < a_0 < \frac{1}{2}$ and $f(a_0)^2 g(a_0) < 1$. Then the sequence $\{a_n\}$ is decreasing.*

Proof. From the hypothesis we deduce that $0 < a_1 < a_0$. Now we suppose that $0 < a_k < a_{k-1} < \cdots < a_1 < a_0 < 1/2$. Then $0 < a_{k+1} < a_k$ if and only if $f^2(a_k) g(a_k) < 1$. Notice that $f(a_k) < f(a_0)$ and $g(a_k) < g(a_0)$. Consequently, $f^2(a_k) g(a_k) < f^2(a_0) g(a_0) < 1$. □

Lemma 3. *If $0 < a_0 < \frac{1}{2d}$, then $f^2(a_0) g(a_0) < 1$.*

Proof. It is easy to show that the inequality $f^2(a_0) g(a_0) < 1$ is equivalent to

$$\varphi(a_0) = 2a_0^4 + (8-d)a_0^3 - 4da_0^2 - 16a_0 + 8 > 0.$$

Since

$$\varphi(0) = 8 > 0, \quad \varphi(0.5) = \frac{9}{8}(1-d) < 0 \quad (\varphi(0.5) = 0 \text{ when } d = 1),$$

$$\varphi'(a_0) = 8a_0^3 + 24a_0^2 - 3da_0^2 - 8da_0 - 16, \quad \varphi'(a_0) < 0 \quad \text{for } 0 < a_0 < 0.5.$$

Therefore there exists

$$\overline{a_0} < \frac{1}{2},$$

such that

$$\varphi(\overline{a_0}) = 0.$$

We compute

$$\varphi\left(\frac{1}{2d}\right) = \frac{d-1}{8d^4}\left(64d^3 - 8d^2 - 9d - 1\right).$$

It is clear that

$$\varphi\left(\frac{1}{2d}\right) > 0$$

for $d > 1$. Thus $\varphi(a_0) > 0$ for $0 < a_0 < \frac{1}{2d}$. □

Lemma 4. *Let us suppose that the hypothesis of Lemma 3 is satisfied and define* $\gamma = a_1/a_0$. *Then*
(i) $\gamma = f(a_0)^2 g(a_0) \in (0;1)$
(ii_n) $a_n \le \gamma^{3^{n-1}} a_{n-1} \le \gamma^{\frac{3^n-1}{2}} a_0$
(iii_n) $f(a_n)g(a_n) \le \frac{\gamma^{3^n}}{f(a_0)}, \quad n \ge 0$

Proof. Notice that (i) is trivial. Next we prove (ii_n) following an inductive procedure. So

$$a_1 \le \gamma a_0$$

and by Lemma 1 we have

$$f(a_1)g(a_1) < f(\gamma a_0)g(\gamma a_0) < f(a_0)\gamma^2 g(a_0) = \frac{\gamma^2 f^2(a_0)g(a_0)}{f(a_0)} = \frac{\gamma^3}{f(a_0)},$$

i.e., (ii_1), (iii_1) are proved. If we suppose that (ii_n) is true, then

$$\begin{aligned} a_{n+1} &= f^2(a_n)g(a_n)a_n \le f^2(\gamma^{\frac{3^n-1}{2}} a_0)g(\gamma^{\frac{3^n-1}{2}} a_0)a_n \\ &\le f^2(a_0)\gamma^{3^n-1} g(a_0)\gamma^{\frac{3^n-1}{2}} a_0 = \gamma^{1+\frac{3}{2}(3^n-1)} a_0 = \gamma^{\frac{3^{n+1}-1}{2}} a_0, \end{aligned}$$

$$\text{and } f(a_{n+1})g(a_{n+1}) \le \frac{f(a_0)\gamma^{3^{n+1}-1} g(a_0)}{f(a_0)} f(a_0) = \frac{\gamma^{3^{n+1}}}{f(a_0)} = \Delta\gamma^{3^{n+1}}$$

$\Delta = \frac{1}{f(a_0)} < 1$ and the proof is complete. □

4 Convergence Study of Chebyshev Method

In this section, we study the sequence $\{a_n\}$ defined above and prove the convergence of the sequence $\{x_n\}$ given by (2). Notice that

$$\begin{aligned} &M\|\Gamma_0\|\|\Gamma_0 F(x_0)\| \le a_0 \\ &\|x_1 - x_0\| \le \left(1 + \frac{a_0}{2}\right)\|\Gamma_0 F(x_0)\|. \end{aligned}$$

Given this situation we prove following statements for $n \geq 1$:

$$(I_n) \quad \|\Gamma_n\| = \|F'(x_n)^{-1}\| \leq f(a_{n-1})\|\Gamma_{n-1}\|$$

$$(II_n) \quad \|\Gamma_n F(x_n)\| \leq f(a_{n-1})g(a_{n-1})\|\Gamma_{n-1}F(x_{n-1})\|$$

$$(III_n) \quad M\|\Gamma_n\|\|\Gamma_n F(x_n)\| \leq a_n$$

$$(IV_n) \quad \|x_{n+1} - x_n\| \leq \left(1 + \frac{a_n}{2}\right)\|\Gamma_n F(x_n)\|$$

$$(V_n) \quad y_n, x_{n+1} \in B(x_0, R\eta), \text{ where } B(x_0, R\eta) = \left\{x \in \Omega : \|x - x_0\| < \frac{1 + a_0/2}{1 - \gamma\Delta}\eta\right\}$$

Assuming

$$\left(1 + \frac{a_0}{2}\right)a_0 < 1, \qquad x_1 \in \Omega,$$

we have

$$\|I - \Gamma_0 F'(x_1)\| \leq \|\Gamma_0\|\|F'(x_0) - F'(x_1)\| \leq M\|\Gamma_0\|\|x_1 - x_0\| \leq \left(1 + \frac{a_0}{2}\right)a_0 < 1.$$

Then, by the Banach lemma, Γ_1 is defined and

$$\|\Gamma_1\| \leq \frac{\|\Gamma_0\|}{1 - \|\Gamma_0\|\|F'(x_0) - F'(x_1)\|} \leq \frac{1}{1 - \left(1 + \frac{a_0}{2}\right)a_0}\|\Gamma_0\| = f(a_0)\|\Gamma_0\|.$$

On the other hand, if $x_n, x_{n-1} \in \Omega$, we will use Taylor's formula

$$F(x_n) = F(x_{n-1}) + F'(x_{n-1})(x_n - x_{n-1}) + \frac{F''(\xi_n)}{2}(x_n - x_{n-1})^2, \tag{9}$$

$$\xi_n = \theta x_n + (1 - \theta)x_{n-1}, \qquad \theta \in (0, 1). \tag{10}$$

Taking into account (2) we obtain

$$x_n - x_{n-1} = \left[I - \frac{1}{2}\Gamma_{n-1}F''(x_{n-1})(y_{n-1} - x_{n-1})\right](y_{n-1} - x_{n-1}). \tag{11}$$

Substituting the last expression in (9) we obtain

$$\begin{aligned} F(x_n) &= -\frac{1}{2}F''(x_{n-1})(y_{n-1} - x_{n-1})^2 + \frac{1}{2}F''(\xi_n)(x_n - x_{n-1})^2 \\ &= \frac{1}{2}\left[F''(\xi_n) - F''(x_{n-1}) - F''(\xi_n)\Gamma_{n-1}F''(x_{n-1})(y_{n-1} - x_{n-1})\right. \\ &\quad \left. + \frac{1}{4}F''(\xi_n)\Gamma_{n-1}^2F''(x_{n-1})^2(y_{n-1} - x_{n-1})^2\right](y_{n-1} - x_{n-1})^2. \end{aligned} \tag{12}$$

Then for $n = 1$, if $x_1 \in \Omega$, we have

$$\|F(x_1)\| \leq \frac{1}{2}\left[K\|\xi_1 - x_0\| + Ma_0 + \frac{1}{4}Ma_0^2\right]\|\Gamma_0 F(x_0)\|^2. \tag{13}$$

From (11) we get

$$\|x_1 - x_0\| \leq \left(1 + \frac{a_0}{2}\right)\|y_0 - x_0\| \leq \left(1 + \frac{a_0}{2}\right)\|\Gamma_0 F(x_0)\|.$$

Using (10) and

$$\|\xi_1 - x_0\| = \theta\|x_1 - x_0\| \leq \theta\left(1 + \frac{a_0}{2}\right)\|\Gamma_0 F(x_0)\|$$

in (13) we obtain

$$\begin{aligned}\|\Gamma_1 F(x_1)\| &\leq \|\Gamma_1\|\|F(x_1)\| \\ &\leq \frac{1}{2}f(a_0)\|\Gamma_0\|Ma_0\left(K\theta\left(1 + \frac{a_0}{2}\right)\frac{1}{M^2\|\Gamma_0\|} + \frac{4 + a_0}{4}\right)\|\Gamma_0 F(x_0)\|^2\end{aligned}$$

or

$$\begin{aligned}\|\Gamma_1 F(x_1)\| &\leq \frac{f(a_0)}{2}a_0^2\left[K\theta\left(1 + \frac{a_0}{2}\right)\frac{1}{M^2 m} + \left(1 + \frac{a_0}{4}\right)\right]\|\Gamma_0 F(x_0)\| \\ &\leq \frac{f(a_0)}{8}a_0^2(4 + a_0)\left(1 + \frac{2K\theta}{M^2 m}\right)\|\Gamma_0 F(x_0)\| \\ &= f(a_0)g(a_0)\|\Gamma_0 F(x_0)\|\end{aligned}$$

and (II_1) is true. To prove (III_1) notice that

$$\begin{aligned}M_1\|\Gamma_1\|\|\Gamma_1 F(x_1)\| &\leq Mf(a_0)\|\Gamma_0\|f(a_0)g(a_0)\|\Gamma_0 F(x_0)\| \\ &\leq f^2(a_0)g(a_0)a_0 = a_1\end{aligned}$$

and

$$\begin{aligned}\|x_2 - x_1\| &\leq \|y_1 - x_1\| + \frac{1}{2}M\|\Gamma_1\|\|\Gamma_1 F(x_1)\|\|y_1 - x_1\| \\ &\leq \left(1 + \frac{a_1}{2}\right)\|y_1 - x_1\| = \left(1 + \frac{a_1}{2}\right)\|\Gamma_1 F(x_1)\|,\end{aligned} \tag{14}$$

and (IV_1) is true. Using

$$\begin{aligned}s\|x_1 - x_0\| &\leq \|y_0 - x_0\| + \frac{1}{2}\|\Gamma_0\|M\|\Gamma_0 F(x_0)\|\|y_0 - x_0\| \\ &\leq \left(1 + \frac{a_0}{2}\right)\|y_0 - x_0\| \\ &\leq \left(1 + \frac{a_0}{2}\right)\eta < \frac{1 + a_0/2}{1 - \gamma\Delta}\eta = R\eta\end{aligned}$$

and

$$\begin{aligned}\|y_1 - x_0\| &\le \|y_1 - x_1\| + \|x_1 - x_0\| \le \left(\frac{\gamma}{f(a_0)} + 1 + \frac{a_0}{2}\right)\eta \\ &= \left(1 + \frac{a_0}{2}\right)\left(1 + \frac{\Delta\gamma}{1 + a_0/2}\right)\eta < \left(1 + \frac{a_0}{2}\right)(1 + \Delta\gamma)\eta \\ &< \frac{1 + a_0/2}{1 - \gamma\Delta}\eta = R\eta\end{aligned}$$

and (14) we have

$$\|x_2 - x_0\| \le \|x_2 - x_1\| + \|x_1 - x_0\| \le R\eta.$$

Thus, $y_1, x_2 \in \overline{B(x_0, R\eta)}$ and (V_1) is true. Now, following an inductive procedure and assuming

$$y_n, x_{n+1} \in \Omega \text{ and } \left(1 + \frac{a_n}{2}\right) a_n < 1,\ n \in \mathcal{N}, \tag{15}$$

the items $(I_n) - (V_n)$ are proved.
Notice that $\Gamma_n > 0$ for all $n = 0, 1, \ldots$. Indeed, if $\Gamma_k = 0$ for some k, then due to statement (I_n), we have $\|\Gamma_n\| = 0$ for all $n \ge k$. As a consequence, the iteration (2), as well as (3) and (4), terminated after kth step, i.e., the convergence of iterations does not hold. To establish the convergence of $\{x_n\}$ we only have to prove that it is a Cauchy sequence and that the above assumptions (15) are true. We note that

$$\begin{aligned}\left(1 + \frac{a_n}{2}\right)\|\Gamma_n F(x_n)\| &\le \left(1 + \frac{a_0}{2}\right) f(a_{n-1}) g(a_{n-1}) \|\Gamma_{n-1} F(x_{n-1})\| \\ &\le \left(1 + \frac{a_0}{2}\right)\|\Gamma_0 F(x_0)\| \prod_{k=0}^{n-1} f(a_k) g(a_k).\end{aligned}$$

As a consequence of Lemma 4 it follows that

$$\prod_{k=0}^{n-1} f(a_k) g(a_k) \le \prod_{k=0}^{n-1} \gamma^{3^k} \Delta = \Delta^n \gamma^{1+3+3^2+\cdots+3^{n-1}} = \Delta^n \gamma^{\frac{3^n - 1}{2}}.$$

So from $\Delta < 1$ and $\gamma < 1$, we deduce that $\prod_{k=0}^{n-1} f(a_k) g(a_k)$ converges to zero by letting $n \to \infty$.

We are now ready to state the main result on convergence for ().

Theorem 1. *Let us assume that $\Gamma_0 = F'(x_0)^{-1} \in \boldsymbol{L}(Y, X)$ exists at some $x_0 \in \Omega$ and $(c_1) - (c_4)$ are satisfied. Suppose that*

$$0 < a_0 < \frac{1}{2d}, \text{ with } \quad d = 1 + 2\omega, \quad \omega = \frac{K}{M^2 m}. \tag{16}$$

Then if $\overline{B(x_0,R\eta)} = \{x \in X; \|x-x_0\| \leq R\eta\} \subseteq \Omega$ *the sequence* $\{x_n\}$ *defined in (2) and starting at* x_0 *has at least R-order three and converges to a solution* x^* *of the Eq. (1). In that case, the solution* x^* *and the iterates* x_n, y_n *belong to* $\overline{B(x_0,R\eta)}$, *and* x^* *is the only solution of Eq. (1) in* $B(x_0, \frac{2}{M\beta} - R\eta) \cap \Omega$. *Furthermore, we have the following error estimates:*

$$\|x^* - x_n\| \leq \left(1 + \frac{a_0}{2}\gamma^{\frac{3^n-1}{2}}\right)\gamma^{\frac{3^n-1}{2}} \frac{\Delta^n}{1-\Delta\gamma^{3^n}}\eta. \tag{17}$$

The proof is the same as Theorem 3.1 in [7, 8].

5 Convergence Study of Modifications of the Chebyshev Method

The convergence of the proposed modifications (3) and (4) is studied analogously as those of Chebyshev method. The difference is only to prove assumption (II_n) for these methods. Therefore, we turn our attention only to the proof of assumption (II_n). At first, we consider a modification (3). For this, if $x_n, y_n \in \Omega$ we obtain from Taylor's formula

$$F(x_n) = -\frac{1}{2}F''(\eta_{n-1})(y_{n-1}-x_{n-1})^2 + \frac{1}{2}F''(\xi_n)(x_n - x_{n-1})^2, \tag{18}$$

where

$$\begin{aligned} \eta_{n-1} &= (1-w)x_{n-1} + wz_{n-1}, \\ \xi_n &= \overline{\theta}x_n + (1-\overline{\theta})x_{n-1}, \quad 0 < \omega, \overline{\theta} < 1. \end{aligned}$$

According to (3) we have

$$x_n - x_{n-1} = \left(I - \frac{1}{2\theta}\Gamma_{n-1}(F'(z_{n-1}) - F'(x_{n-1}))\right)(y_{n-1} - x_{n-1}).$$

Substituting the last expression into (18) we get

$$\begin{aligned} F(x_n) = &\frac{1}{2}\left(F''(\xi_n) - F''(\eta_{n-1})\right)(y_{n-1}-x_{n-1})^2 \\ &+\frac{1}{2}F''(\xi_n)\left[-\frac{1}{\theta}\Gamma_{n-1}(F'(z_{n-1}) - F'(x_{n-1}))(y_{n-1}-x_{n-1})^2\right. \\ &\left.+\frac{1}{4\theta^2}\Gamma_{n-1}^2(F'(z_{n-1}) - F'(x_{n-1}))^2(y_{n-1}-x_{n-1})^2\right]. \end{aligned} \tag{19}$$

Then, for $n=1$, if $y_0 \in \Omega$, we have

$$\|F(x_1)\| \leq \left[\frac{K}{2}\|\xi_1-\eta_0\| + \frac{M}{2\theta}\|\Gamma_0\|M\theta\|y_0-x_0\| + \frac{M}{8\theta^2}\|\Gamma_0\|^2M^2\theta^2\|y_0-x_0\|^2\right]\|y_0-x_0\|^2.$$

Since $\xi_1-\eta_0 = \overline{\theta}(x_1-x_0) - w\theta(y_0-x_0)$, it follows

$$\|\xi_1-\eta_0\| \leq \overline{\theta}\|x_1-x_0\| + w\theta\|y_0-x_0\| \leq \left(\overline{\theta}\left(1+\frac{a_0}{2}\right)+w\theta\right)\|y_0-x_0\|.$$

If we take $\hat{\theta} = \max(\overline{\theta}, w\theta)$, then we get the following estimate:

$$\|F(x_1)\| \leq \left\{K\hat{\theta}\left(1+\frac{a_0}{2}\right)\frac{M^2\|\Gamma_0\|}{M^2\|\Gamma_0\|}\|\Gamma_0F(x_0)\|^2 + \frac{M^2\|\Gamma_0\|}{2}\|\Gamma_0F(x_0)\|^2 + \frac{M^3}{8}\|\Gamma_0\|^2\|\Gamma_0F(x_0)\|^3\right\}\|\Gamma_0F(x_0)\|.$$

Therefore, we have

$$\|\Gamma_1F(x_1)\| \leq f(a_0)g(a_0)\|\Gamma_0F(x_0)\|, \quad g(a_0) = \frac{a_0^2(4+a_0)}{8}d_1 \text{ with } d_1 = 1+2.5\omega.$$

Analogously, for the modification (4), we have

$$F(x_n) = -\frac{1}{2}\left[\left(1+\frac{b}{2}\right)F''(\eta_{n-1}) - \frac{b}{2}F''(\zeta_{n-1})\right](y_{n-1}-x_{n-1})^2 + \frac{F''(\xi_n)}{2}(x_n-x_{n-1})^2, \tag{20}$$

$$\begin{aligned}\xi_n &= \alpha x_{n-1} + (1-\alpha)x_n, \qquad \alpha \in (0,1),\\ \eta_{n-1} &= \theta x_{n-1} + (1-\theta)y_{n-1}, \quad \theta \in (0,1),\\ \zeta_{n-1} &= wx_{n-1} + (1-w)z_{n-1}, \quad w \in (0,1).\end{aligned}$$

Notice that

$$\begin{aligned}\xi_n - \eta_{n-1} &= (1-\theta)(x_{n-1}-y_{n-1}) + \rho(x_n-x_{n-1})\\ &= (\rho-(1-\theta))(y_{n-1}-x_{n-1}) - \frac{\rho}{2}\Gamma_{n-1}D_n(y_{n-1}-x_{n-1})^2,\\ \eta_{n-1}-\zeta_{n-1} &= (1-w)(x_{n-1}-z_{n-1}) + \lambda(y_{n-1}-x_{n-1})\\ &= (1-w+\lambda)(y_{n-1}-x_{n-1}),\\ \text{where } \rho &= 1-\alpha, \qquad \lambda = 1-\theta,\end{aligned}$$

$$x_n - x_{n-1} = \left[I - \frac{1}{2}\Gamma_{n-1}\left(\left(1+\frac{b}{2}\right)F''(\eta_{n-1}) - \frac{b}{2}F''(\xi_{n-1})\right)(y_{n-1}-x_{n-1})\right] \times (y_{n-1}-x_{n-1}).$$

Substituting the last expression into (20) we have

$$F(x_n) = \frac{1}{2}B_n(y_n - x_{n-1})^2 - \frac{F''(\xi_n)}{2}\Gamma_{n-1}D_n(y_n - x_{n-1})^3 + \frac{F''(\xi_n)}{8}\Gamma_{n-1}^2 D_n^2 (y_{n-1}-x_{n-1})^4, \tag{21}$$

where

$$B_n = F''(\xi_n) - F''(\eta_{n-1}) - \frac{b}{2}\left(F''(\eta_{n-1}) - F''(\xi_{n-1})\right),$$
$$D_n = \left(1+\frac{b}{2}\right)F''(\eta_{n-1}) - \frac{b}{2}F''(\xi_{n-1}).$$

If $\xi_n, \eta_{n-1}, \zeta_{n-1} \in \Omega$ then we have

$$\|B_n\| \le K\left[|\beta - (1-\theta)| - \frac{b}{2}(1-w+\gamma)\right]\|y_{n-1}-x_{n-1}\| + \frac{K\|\Gamma_{n-1}\|\beta}{2}M\|y_{n-1}-x_{n-1}\|^2,$$
$$\|D_n\| \le M.$$

Using these expressions we get

$$\|\Gamma_n F(x_n)\| \le f(a_{n-1})\frac{a_{n-1}^2}{2}\left\{\frac{K}{M^2 m}\hat{d} + \left(1+\frac{a_{n-1}}{4}\right)\right\}\|\Gamma_{n-1}F(x_{n-1})\|,$$

where

$$\hat{d} = |\beta - (1-\theta)| - \frac{b}{2}(1-w+\gamma) + \beta a_{n-1} < 3 + a_{n-1} < 4\left(1+\frac{a_{n-1}}{4}\right),$$

$$|\beta - \gamma| < 1 \qquad 0 < 1 - w + \gamma < 2.$$

Then we obtain

$$\|\Gamma_n F(x_n)\| \le f(a_{n-1})g(a_{n-1})\|\Gamma_{n-1}F(x_{n-1})\|$$
$$g(a_{n-1}) = \frac{a_{n-1}^2(4+a_{n-1})}{8}d_2, \qquad d_2 = 1+4\omega.$$

For the modifications (3) and (4) the cubic convergence theorem 1 is valid, in which d equals to $1+5\omega$ and $1+4\omega$, respectively.

It should be mentioned that in [4] was constructed a family of predictor-corrector methods free from second derivative. But these methods, except the case $A2_0$, require more computational cost even as compared to the modification (3).

6 Mild Convergence Conditions

In order to obtain mild convergence conditions for these methods we first consider inexact Newton method (IN) for (1):

$$F'(x_k)s_k = -F(x_k) + r_k, \tag{22}$$

$$x_{k+1} = x_k + s_k, \quad k = 0,1,\ldots,x_0 \in \Omega \tag{23}$$

The terms $r_k \in R^n$ represent the residuals of the approximate solutions s_k [3,4]. We consider a local convergence result [3,4]:

Theorem 2. *Given $\eta_k \leq \overline{\eta} < t < 1, k = 0,1,\ldots$, there exists $\varepsilon > 0$ such that for any initial approximation x_0 with $\|x_0 - x^*\| \leq \varepsilon$, the sequence of the IN iterates (22) satisfying*

$$\|r_k\| \leq \eta_k \|F(x_k)\|, \qquad k = 0,1,\ldots \tag{24}$$

converges to x^*.

Moreover we know that the IN converges superlinearly when $\eta_k \to 0$ as $k \to \infty$. Now we analyze the connection between the inexact Newton method and the Chebyshev method (2) and its modifications (3) and (4). To this end we rewrite (2)–(4) in the form (22) with

$$r_k = F'(x_k)s_k + F(x_k) = -\frac{1}{2}F''(x_k)(y_k - x_k)^2,$$

$$r_k = -\frac{1}{2\theta}(F'(z_k) - F'(x_k))(y_k - x_k),$$

and

$$r_k = -\frac{1}{2}\left(\left(1 + \frac{b}{2}\right)F(y_n) + bF(x_n) - \frac{b}{2}F(z_n)\right),$$

respectively.

Theorem 3. *Let us assume that $\Gamma_0 = F'(x_0)^{-1} \in \mathscr{L}(Y,X)$ exists at some $x_0 \in \Omega$, and the conditions (c_1)–(c_3) are satisfied. Suppose that $0 < a_0 < 0.5$. Then the sequences $\{x_k\}$ given by (2), (3) and (4) converge to x^*.*

Proof. We first observe that the sequence $\{a_k\}$ given by (7) and (8) with $d = 1$ is decreasing, i.e.,

Table 1 The number of iterations

Examples	x_0	NM	CM	MOD 1			MOD 2		
(I)	1.5	7	4	5	5	5	4	4	5
(II)	2.0	5	5	3	4	4	4	4	3
(III)	1.5	6	6	3	4	4	3	3	3
(IV)	1	6	4	6	4	4	4	4	-

$$0 < a_{k+1} < a_k < \cdots < a_1 < a_0 < \frac{1}{2}. \tag{25}$$

It is easy to show that for residuals r_k of all the methods (2),(3) and (4) hold the following estimation

$$\|r_k\| \leq \frac{a_k}{2}\|F(x_k)\|, \qquad \left(\eta_k = \frac{a_k}{2}\right). \tag{26}$$

From (25) and (26) follows $\eta_k \to 0$ as $k \to \infty$. Then by Theorem 2 the methods (2)–(4) converge to x^*. □

The assumptions in Theorem 3 are milder than cubic convergence condition in Theorem 1 with $d > 1$.

7 Numerical Results and Discussion

Now, we give some numerical examples that confirm the theoretical results. First, we consider the following test equations:

$$\begin{aligned}
&(I) \quad x^3 - 10 = 0,\\
&(II) \quad x^3 + 4x^2 - 10 = 0,\\
&(III) \quad ln(x) = 0,\\
&(IV) \quad \sin^2 x - x^2 + 1 = 0.
\end{aligned}$$

All computations are carried out with a double arithmetic precision, and the number of iterations, such that $\|F(x_n)\| \leq 1.0e-16$, is tabulated (see Table 1). We see that the third-order MOD 1 and MOD 2 takes less iterations to converge as compared to second-order Newton's method (NM).

Now we consider the following systems of equations:

$$(V) \quad F(x) = \begin{pmatrix} x_1^2 - x_2 + 1 \\ x_1 + \cos\left(\frac{\pi}{2}x_2\right) \end{pmatrix} = 0,$$

Table 2 The computational cost of the methods

Methods	Evaluation of F	Evaluation of F'
NM	1	1
MOD 1	1	2
MOD 2	3 (2 when $b = 0$ or $b = 2$)	1

Table 3 The number of iterations

Examples	x_0	NM	CM	MOD 1		MOD 2		
				$\theta = 0.5$	$\theta = 1$	$b = -2$	$b = -1$	$b = 0$
(V)	(0;0.1)	8	5	5	6	3	2	2
(VI)	(0,0,0,1,1,0)	6	6	4	4	4	4	4

$$(VI) \quad F(x) = \begin{pmatrix} x_1x_3 + x_2x_4 + x_3x_5 + x_4x_6 \\ x_1x_5 + x_2x_6 \\ x_1 + x_3 + x_5 - 1 \\ -x_1 + x_2 - x_3 + x_4 - x_5 + x_6 \\ -3x_1 - 2x_2 - x_3 + x_5 + 2x_6 \\ 3x_1 - 2x_2 + x_3 - x_5 + 2x_6 \end{pmatrix} = 0.$$

As seen from the Tables 1–3, that the proposed modifications (MOD 1, MOD 2) are almost always superior to these classical predecessor, the Chebyshev method (CM), because of their convergence order is as same as CM, but these are simpler and free from second derivative.

We also compared the computational cost of two modifications to the classical NM (see Table 2). The numerical results showed that MOD 2 is the most effective method especially when $b = -2$ or $b = 0$.

Conclusion

In this chapter we proposed new two families of methods which include many well-known third-order methods as particular case. We proved third-order convergence theorem for these modifications and as well as for Chebyshev method. The new methods were compared by their performance to Newton's method and Chebyshev method, and it was observed that they show better performance than NM and CM.

References

1. Babajee D.K.R., Dauhoo M.Z., Darvishi M.T., Karami A., Barati A.: Analysis of two Chebyshev like third order methods free from second derivatives for solving systems of nonlinear equations. Comput.Appl.Math **233**, 2002–2012 (2010)

2. Candela V., Marquina A.: Recurrence relations for rational cubic methods II: The Chebyshev method. Computing. **45**, 355–367 (1990)
3. C*â*tinas E.:The inexact, inexact perturbed and quasi Newton methods are equivalent models. Math. Comp. **74** , 291–301 (2004)
4. Dembo R.S., Eisenstat S.C., Steihang T.: Inexact Newton methods. SIAM J.Numer.Anal. **19**, 400–408 (1982)
5. Ezquerro J.A., Hernandez M.A.: A uniparametric Halley type iteration with free second derivative. Int.J.Pure Appl.Math. **61**,103–114 (2003)
6. Ezquerro J.A., Hernandez M.A.: On Halley type iterations with free second derivative. J.Comput.Appl.Math. **170**, 455–459 (2004)
7. Hernandez M.A.: Second derivative free variant of the Chebyshev method for nonlinear equations. J.Optimization theory and applications **104**,501–515 (2000)
8. Hernandez M.A., Salanova M.A.: Modification of the Kantorovich assumptions for semilocal convergence of the Chebyshev. J.Comput.Appl.Math. **126**, 131–143 (2000)
9. Homeier H.H.H.: On Newton type methods with cubic convergence. J.Comput. Appl.Math. **176**, 425–432 (2005)
10. Kou J, Li Y, Wang X.: A modification of Newton method with third order convergence. Appl.Math.Comput. **181**, 1106–1111 (2006)
11. Lambert J.D.:Computational methods in ordinary differential equations. John Wiley and Sons, London, England (1976)
12. Potra F.A., Ptak V.:Nondiscrete induction and iterative processes. Pitman, NY (1984)
13. Stoer Bulirsch : Introduction to numerical analysis, 3rd edition. Springer (2002)
14. Zhanlav T.: Note on the cubic decreasing region of the Chebyshev method. J.Comput.Appl. Math. **235**, 341–344 (2010)
15. Zhanlav T. Chuluunbaatar O.: Some iterations of higher order convergence for nonlinear equations. Vestnik Friendship University of People **3**, 70–78 (2009)(in russian)