Habilitation à Diriger des Recherches

presentée à

l' UNIVERSITÉ PAUL SABATIER de Toulouse

École Doctorale MITT

Specialité : Optimisation Numérique et Recherche Opérationnelle

par

Sonia CAFIERI
École Nationale de l'Aviation Civile

From local to global and back: A closed walk in Mathematical Programming and its Applications

Soutenue à Toulouse le 10 décembre 2012
devant le Jury composé de :

Emilio CARRIZOSA	Professeur, Universidad de Sevilla (Spain)	*Rapporteur*
Philippe MAHEY	Professeur, Université Blaise Pascal-Clermont 2	*Rapporteur*
Panos PARDALOS	Professeur, University of Florida (USA)	*Rapporteur*
Jean-Baptiste HIRIART-URRUTY	Professeur, Université Paul Sabatier	*Examinateur*
Bertrand JOUVE	Professeur, Université Lumière Lyon 2	*Examinateur*
Frédéric MESSINE	Maître de Conférences-HDR, ENSEEIHT-INPT	*Examinateur*
Marcel MONGEAU	Professeur, École Nationale de l'Aviation Civile	*Examinateur*

Abstract

An overview of my research contributions in optimization is provided, passing through mixed-integer nonlinear optimization, nonlinear continuous local optimization and network (graph) clustering.

The first chapter deals with mixed-integer nonlinear programming and deterministic global optimization, and presents contributions concerning theoretical investigations as well as applications to real-world problems. We mainly discuss about convex relaxations and automatic reformulations of mathematical programming problems, aiming at enhancing the efficiency of Branch-and-Bound algorithms. Focusing on polynomial programming, we investigate tight convex relaxations for multilinear monomials and the generation of compact relaxations of polynomial problems based on a special reformulation-linearization technique. Among applications, a special attention is devoted to real-life problems arising in Air Traffic Management. We propose new mathematical models and solution approaches from mixed-integer optimization on the one hand, and optimal control on the other hand.

A few topics in nonlinear continuous optimization are described in the second chapter. Interior point methods for quadratic programming and their linear algebra kernels (KKT systems) are first discussed. The focus is on iterative methods for the KKT systems and related issues, such as preconditioning techniques and convergence properties. The other discussed topic relates, again, to air traffic problems. This concerns the mentioned optimal control-based approaches that lead to nonlinear problems.

The third chapter presents my main results in the area of network clustering. The problem of identifying clusters in networks can be formulated using mathematical programming and usually leads to a combinatorial optimization problem. My contributions concern clustering criteria and corresponding clustering methods. A special attention is devoted to exact methods, used either to solve the whole optimization problem or, locally, subproblems arising in hierarchical heuristics, or to refine solutions previously obtained by other methods.

Résumé

Ce document propose un parcours de mes travaux de recherche en optimisation, en passant par l'optimisation mixte en variables entières, l'optimisation non-linéaire continue locale et le *clustering* dans les réseaux (graphes).

Le premier chapitre traite de la programmation non linéaire mixte en variables entières et de l'optimisation globale déterministe. Il présente des contributions relatives à des investigations théoriques ainsi que des applications à des problèmes concrets. Nous discutons principalement de relaxations convexes et de reformulations automatiques de problèmes de programmation mathématique, dans le but d'améliorer l'efficacité des algorithmes de *Branch-and-Bound*. Dans le cadre de la programmation polynomiale, nous avons étudié des relaxations convexes pour les monômes multilinéaires et la génération de relaxations compactes de problèmes polynomiaux basés sur une technique spécifique de reformulation-linearisation (RLT). Parmi les applications, une attention particulière est portée à des problèmes qui se posent dans la gestion du trafic aérien. Nous avons proposé de nouveaux modèles mathématiques et des approches de résolution basées d'une part sur l'optimisation mixte en variables entières et d'autre part sur le contrôle optimal.

Deux thèmes de l'optimisation continue non-linéaire sont décrits au deuxième chapitre. Des méthodes de point intérieur pour la programmation quadratique et leurs noyaux d'algèbre linéaire (systèmes KKT) sont d'abord discutées. L'accent est mis sur les méthodes itératives pour les systèmes KKT et sur des questions connexes, telles que les techniques de préconditionnement et les propriétés de convergence. L'autre sujet discuté concerne, encore une fois, des problémes de trafic aérien. Il porte sur les approches déjà mentionnées de contrôle optimal qui conduisent à des problèmes non-linéaires.

Le troisième chapitre présente mes principaux résultats dans le domaine du *clustering* dans les réseaux. Le problème de l'identification de clusters dans les réseaux peut être formulé en utilisant la programmation mathématique et conduit généralement à un problème d'optimisation combinatoire. Mes contributions concernent les critères de classification et les méthodes de clustering correspondantes. Une attention particulière est portée aux méthodes exactes utilisés pour résoudre l'ensemble du problème d'optimisation ou, localement, les sous-problèmes survenant dans des heuristiques hiérarchiques, ou enfin dans le raffinement des solutions obtenues précédemment par d'autres méthodes.

Acknowledgements

I would like to express my gratitude to Emilio Carrizosa, Philippe Mahey and Panos Pardalos for agreeing with enthusiasm to be referees for my Habilitation à Diriger des Recherches. I also thank Jean-Baptiste Hiriart-Urruty, Bertrand Jouve, Frédéric Messine and Marcel Mongeau who made me the honour of participating in my jury.

I would like to address special thanks to Marcel Mongeau for his strong encouragement and support. The achievement of this work would not have been possible without his help. I am also grateful for his precious comments and suggestions on the manuscript. I would like to thank Frédéric Messine for being an invaluable source of motivation as well. We have a great collaboration that I hope will continue for years.

Moving to France allowed me to pursue my work as a researcher and played a crucial role in my professional development. I express my gratitude to the colleagues that welcomed me, first at École Polytechnique in Paris and then at ENAC in Toulouse. My thoughts go to Pascal Brisset who welcomed me in his group at ENAC, it would have been a pleasure to continue collaborating with him. I also extend warm thanks to the members of the Laboratory MAIAA at ENAC for the good working atmosphere and the fruitful exchanges.

The work presented in this document owes much to the collaboration with a number of coauthors from around the world. I would like to thank all of them for their precious collaboration and insightful discussions. I would especially like to thank Leo Liberti who motivated my interest in the field of Mixed-Integer Optimization. I am indebted to him for giving me the first opportunity to work at such a great institution as École Polytechnique and for his advice and encouragement to continue my career in France.

I would like to express my special thanks and admiration to Pierre Hansen. Most of the results I obtained on graph clustering are the outcome of our joint research. I could benefit of his outstanding knowledge of Optimization and Mathematical Programming. I learned a lot from his clarity of thinking and benefited from his open-mindedness towards trying new ideas. I also have to thank him for providing me with valuable opportunities to work at GERAD in Montréal.

Finally, I have the pleasure to supervise the Ph.D. thesis of Loïc Cellier, who I thank for the joint work on a very interesting topic.

Contents

To my father
Francesco Cafieri

Preface

This document is a synthesis of my research activity about a few themes in optimization, which I became interested in since my Ph.D.

Treated topics go from nonlinear continuous (local) optimization to mixed-integer nonlinear programming and global optimization, passing through combinatorial optimization for network (graph) clustering. This corresponds to the evolution of my research interests and the progression of my career, which led me to work within different research groups in various countries.

The presentation is organized according to my recent scientific interests and to the relevance between the topics rather than following a chronological order. The three main research themes are presented in three distinct chapters, followed by appendices corresponding to a selection of my publications integrally reported.

This report is not intended to be technical, so theorem proofs, mathematical formulation details, as well as detailed numerical results, are only reported in the papers presented in the appendices.

Perspectives for future research are drawn at the end of the document, and an overview of my other academic activities is also reported.

Introduction and Main Research Contributions

Optimality has been often observed in nature, has inspired artists and has assumed over the centuries aesthetical and also metaphysical valences. In nature, it has been associated to balance and harmony, which seem to obey minimality principles. Dante Alighieri summed up a common sentiment in Middle-Age, saying "Omne superfluum Deo et Naturae displiceat." (D. Alighieri, De Monarchia I. xiv. 1 2-3), meaning "All that is superfluous is displeasing to God and Nature." More recently, our world dominated by the market seeks strategies to achieve optimal results, maximizing profits and minimizing costs. This search for optimality, i.e., minimizing or maximizing some function under given constraints, is in fact attracting a growing interest in more and more application domains, motivating the research in the fascinating field of Mathematical Optimization, at the interface with Computer Science and Operations Research.

The need for mathematically modelling complex real systems as optimization problems unsurprisingly led to the development of a variety of subfields in Optimization, as well as a variety of solution approaches. While providing a complete taxonomy of optimization problems is out of the scope of this report, we recall the two main classes of *Continuous Optimization* and *Discrete and Integer Optimization*, the first dealing with continuous (real) variables and the second dealing with discrete structures or variables restricted to take only integer values. A number of significant subclasses can be identified. *Nonlinear Programming* (Nonlinearly Constrained Optimization) (NLP), aiming to minimize a nonlinear function subject to nonlinear constraints, represents the more general continuous optimization problems class, with significant sub-categories like *Linear Programming* (LP) and *Quadratic Programming* (QP). *Combinatorial Optimization* deals with problems on discrete structures (e.g. graphs).
Mixed-Integer Nonlinear Programming (MINLP) deals with the most general optimization problems encompassing the above-mentioned two classes, involving both continuous and discrete variables and nonlinear constraint functions. These are among the most challenging computational optimization problems, arising in countless applications from

various areas. While research on mixed-integer linear optimization is quite advanced, MINLP is still considered an emerging area that is likely to grow in the coming years.

Numerical methods for optimization problems can be classified with respect to the properties of their target problem and on the type of guarantees that they provide for the final solution. They broadly fall into two classes. *Deterministic methods* terminate, under suitable assumptions, with a solution which is guaranteed to be optimal (or ϵ-optimal) or with an indication that there is no feasible solution. Typical descent-type methods for continuous optimization (gradient methods, Newton's method, etc.) fall in this category. They are also classified as *local* optimization methods to emphasize that the solution found is optimal only with respect to its neighborhood. Deterministic methods to find *global* optima of nonlinear and more generally MINLP problems perform (implicitly) exhaustive searches generally based on the exploration of a search tree, implementing strategies to speed up the search avoiding to check every possible configuration. *Heuristic methods*, which represent the second relevant class of methods, on the contrary, do not provide a guarantee of optimality for the computed solution. The most popular heuristic methods are based on local searches and evolutionary algorithms. This report will mainly refer to deterministic methods.

In my career I had the chance to work on various topics from different areas of optimization. During my Ph.D. (University of Naples) my research interests mainly involved continuous QP with a focus on linear algebra issues arising in Interior Point methods for QP. My post-doctorate at École Polytechnique (Paris) was in the context of a project on *reformulations* in Mathematical Programming mainly addressing mixed-integer nonlinear problems, that led me to develop interest in MINLP. In that period I also became interested in combinatorial optimization, mainly in the context of *graph clustering*, thanks to my collaboration with Pierre Hansen (GERAD & HEC Montréal). Since my recruitment at École Nationale de l'Aviation Civile (Toulouse), aeronautical applications have been among my research topics and represent an interesting domain of application of the expertise I acquired previously. In the context of aeronautical applications, I am applying mixed-integer optimization as well as optimal control techniques. Hence, optimal control brought me back to nonlinear programming, closing the path form local to global and back.

I discovered various aspects of several optimization sub-domains along this closed walk, also moving a step forward from a purely academic research towards a growing interest for real-life applications. Along this amazing path, a few elements constitute a common thread: Mathematical Programming based modeling for the addressed problems, a special attention for deterministic (exact) methods, and the development of software tools to test the investigated approaches.

In the reminder of this section I briefly present the main research topics I have dealt with and my main contributions. They will be described in more details in the following

chapters. Each chapter focuses on one theme, presenting the main contributions. The expertise in these different themes, crossed with upcoming projects, help developing perspectives for the next years. Perspectives, highlighting tracks for future research, are drawn at the end of this report.

MINLP and Global Optimization

Mixed-integer optimization and (deterministic) global optimization constitute one of my main research topics since a few years. In this context, my contributions concern theoretical investigations, specially in the framework of reformulations in mathematical programming and convex relaxations to enhance the efficiency of Branch-and-Bound based algorithms, as well as applications to concrete problems.

The best known method to find exact or at least ϵ-approximate global solutions to mixed-integer nonlinear problems is the spatial Branch-and-Bound algorithm, which rests on computing lower bounds to the value of the objective function to be minimized on each region that it explores. These lower bounds are often computed by solving convex relaxations of the original problem. Within this framework, [Cafieri et al., 2010c, Belotti et al., 2012b, Cafieri et al., 2012c] present contributions in polynomial programming.
In [Cafieri et al., 2010c], we study convex relaxations of quadrilinear monomials. Different convex relaxations can be obtained for multilinear terms by applying associativity in different ways. We rewrite quadrilinear terms as products of bilinear and trilinear terms (for which convex envelopes are known) and derive corresponding convex relaxations. Using a general technique, we formally establish the intuitive fact that any relaxation for k-linear terms that employs a successive use of relaxing bilinear terms (via the bilinear convex envelope) can be improved by employing instead relaxations of trilinear terms (via the trilinear convex envelope). We present a thorough computational analysis which helps establishing which relaxations are strictly tighter, also confirming the results of our investigation by testing on real-life problems.
In [Belotti et al., 2012b], 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.

Reduced RLT constraints (rRLT) are a special class of Reformulation-Linearization Technique constraints. They apply to nonconvex (both continuous and mixed-integer) quadratic programming problems subject to systems of linear equality constraints. In [Cafieri et al., 2012c], we present an extension to the general case of polynomial programming problems and discuss the derived convex relaxation. We also show how to perform rRLT constraint generation so as to reduce the number of inequality constraints in the relaxation, thereby making it more compact and faster to solve.

Relaxations of nonconvex problems fall more generally in the area of reformulations. Reformulation techniques are often used to re-cast a particular type of problem into a

formulation which is amenable to be solved by a given algorithm. In [Liberti et al., 2009], we present a survey of existing reformulations, some example applications, and describe the implementation of a software framework for reformulation and optimization.
This software, the Reformulation-Optimization Software Engine (ROSE), is described in more details in [Liberti et al., 2010]. In Mathematical Programming a considerable amount of symbolic transformations is essential to solve difficult optimization problems. We describe how ROSE performs (automatic) symbolic computation on mathematical programming formulations.

The algorithmic efficiency of a Branch-and-Bound algorithm depends on many factors, among which the width of the bounding box for the problem variables at each Branch-and-Bound node naturally plays a critical role. The practically fastest box-tightening algorithm is known as FBBT (Feasibility-Based Bounds Tightening). In [Belotti et al., 2012a], we model FBBT by using fixed-point equations in terms of the variable bounding box, and we treat these equations as constraints of an auxiliary mathematical program. We demonstrate that the auxiliary mathematical problem is a linear program, which can of course be solved in polynomial time.

Contributions in [Cafieri and Durand, 2012] and [Cafieri et al., 2012e] focus on real-life applications.
An interesting application arising in Air Traffic Management is the resolution of aircraft conflicts, that occur when aircraft get "too close" to each other according to their predicted trajectories. The problem can be modeled as a global optimization problem. In [Cafieri and Durand, 2012], we propose modeling and resolution techniques based on mixed-integer nonlinear optimization.
The optimal design of electrical machines can also be mathematically modeled as a mixed-integer nonlinear optimization problem. In [Cafieri et al., 2012e], we investigate the impact of different mathematical formulations on the results obtained using an optimization solver widely used in the engineering community.

Nonlinear Continuous Optimization

A few topics in Nonlinear Programming (NLP) constituted my first research interests, and I recently had the occasion to work again in the context of NLP, thanks to an optimal control approach I got interested in. Thus, my contributions in this area focus on two specific topics, the one related to Interior Point methods and their linear algebra kernels, and the other arising in Optimal Control for a real-world application.

Interior Point algorithms for nonlinear programming and related linear algebra issues represented the main research topic of my Ph.D. and the first stage of my post-doctoral activity. Interior Point methods are effective and widely-employed methods for nonlinear programming problems. When using these methods, a crucial issue is the availability of efficient and robust computational linear algebra kernels, especially for the solution of

the linear system (the KKT system) that arises at each iteration. We focused on the development, the analysis and the implementation of iterative methods for the KKT systems, addressing quadratic programming. Contributions are reported in [Cafieri et al., 2006, Cafieri et al., 2007a, Cafieri et al., 2007c, Cafieri et al., 2007b, Cafieri et al., 2007d]. A few key ingredients for the success of an iterative approach are specially investigated, such as the use of suitable preconditioners, adaptive stopping criteria, and the reassessment of the convergence theory of the Interior Point method subsequent to the computation of inexact solutions of the linear systems.

The case of quadratic programming subject to variables bound constraints only is analyzed in [Cafieri et al., 2006], where theoretical and computational issues are dealt with and an effective solution strategy for the KKT systems is proposed, based on a Conjugate Gradient method with an Incomplete Cholesky factorization-based preconditioner with limited and predictable memory requirements.

In [Cafieri et al., 2007a], we analyze the behavior of a *Constraint Preconditioner* with the Conjugate Gradient algorithm. The main contribution is the proof, for KKT systems in the augmented form deriving from linear inequality constraints and nonnegative variable bounds, of the equivalence with a suitable preconditioned Conjugate Gradient applied to the positive-definite normal equations. As a consequence, no breakdown occurs and the algorithm converges even if the augmented system matrix is indefinite.

Iterative solvers allow us to use adaptive accuracy requirements in the solution of the KKT systems to avoid unnecessary iterations when the current Interior Point iterate is far from the solution. Adaptive stopping criteria deriving from the convergence theory of the considered Interior Point method are devised in [Cafieri et al., 2007c].

In [Cafieri et al., 2007b], we propose an approximation strategy for the constraint preconditioner, with the aim of reducing the cost of the preconditioner application.

The Interior Point framework of choice is the Potential Reduction method. In this context, the main contribution, described in [Cafieri et al., 2007d], is the extension of the global convergence properties of the method in order to take into account an inexact solution of the inner linear KKT systems.

A contribution concerning a real-life application appears in [Cellier et al., 2012]. Specifically, the same application in Air Traffic Management considered in [Cafieri and Durand, 2012] is dealt with, this time applying an optimal control approach. A direct method, typically used in optimal control, leads to the solution of a NLP problem. Specific strategies tailored on the target application are proposed to reduce the computational complexity of the standard approach. We propose a decomposition of the problem at hand and the combination of a direct and an indirect optimal control method.

Network Clustering

Networks, or graphs, are a basic and versatile tool for the study of complex systems in a variety of domains. Typically, the vertices of a graph are associated with the entities of the system under study, and edges express whether a relation defined on all pairs of vertices holds or not for each such pair. A topic of particular interest in the study of complex networks is the identification of clusters. Broadly speaking, a cluster of a graph is a subset of vertices such that there are more edges within the cluster than edges joining it to the outside. *Clustering* on networks, also known as *community detection*, refers to this interesting topic in the study of complex networks and has become in few years one of my main research interests.

The problem of identifying clusters can be formulated using mathematical programming and usually leads to a combinatorial optimization problem. Several models were proposed. One of the most studied and exploited is the maximization of the so-called *modularity*. A specific behavior of the modularity function is analyzed in [Cafieri et al., 2010b]. Using the definition of modularity, graphs are compared to a *null model* where the degree distribution is maintained while edges are placed at random, where there will be loops and possibly multiple edges. We derive sharp bounds on the expected number of loops, and their impact on the modularity of simple graphs. Then, we propose modified null models associated with graphs without loops and without multiple edges.

Contributions in [Aloise et al., 2010, Cafieri et al., 2011, Cafieri et al., 2012b, Cafieri et al., 2012d] concern solution methods for the maximizing modularity optimization problem, with a focus on exact methods.
In [Aloise et al., 2010], we assess and advance the state of the art of exact algorithms for modularity maximization. We extend the two exact algorithms in the literature, namely the one by Xu et al.(2004) based on a mixed-integer quadratic formulation and the row generation for clique partitioning of Grötschel and Wakabayashi (1989), using column generation methods. We raise the size of instances solved exactly.

In [Cafieri et al., 2011], we propose a hierarchical divisive heuristic which is locally optimal in the sense that each of the successive bipartitions is performed in a provably optimal way, solving the bipartition problem by an exact method. Reformulations of the proposed mathematical programming model for bipartitions, to enhance efficiency of the divisive heuristic, are then explored in [Cafieri et al., 2012b].

The exact algorithm for graph bipartition, which is the basis of the proposed divisive heuristic, is also exploited in an approach applied as a post-processing to heuristic methods in order to improve their performances, in [Cafieri et al., 2012d]. Starting from a given partition, we test with the exact algorithm for bipartitioning if it is worthwhile to split some communities, or to merge two of them, or to combine the two actions.

Contributions in [Cafieri et al., 2010a] and [Cafieri et al., 2012a] concern other network clustering criteria and corresponding optimization algorithms. In [Cafieri et al.,

2010a], we extend the definition of community in the weak sense of Radicchi et al. (2004) into a criterion for a bipartition to be optimal: one seeks to maximize the minimum, for both classes of the bipartition, of the ratio of inner edges to cut edges. We define this way the *edge-ratio* criterion. A hierarchical divisive algorithm is proposed for identifying communities, where bipartitions are made in an optimal way according to the introduced criterion. This includes an exact solution of the problem of detecting indivisible communities.

In [Cafieri et al., 2012a], we first define an enumerative algorithm to list all partitions in the strong sense (i.e., with only strong communities, according to the definition of Radicchi et al. (2004)) of a network of moderate size. We then extend the concept of strong communities to *almost-strong* communities and adapt the algorithm, obtaining more informative partitions.

Applications to real-life problems: the ATM context

When I started working at ENAC - École Nationale de l'Aviation Civile - I also began to be interested in real-life applications arising in Air Traffic Management (ATM). A number of challenging problems arising in ATM constitute in fact interesting research topics, particularly in Operations Research and Optimization. ATM applications, specially air traffic conflict detection and resolution, take therefore a prominent place in my research activity among other real-life applications. They are also the main topic of a Ph.D. thesis that I am currently supervising and of a research project recently funded by the ANR (French National Agency of Research) of which I am responsible and that will be developed during the next three years. The contributions in this area will be described in Chapters 1 and 2. Here we draw the main lines of the ATM context and the impact of optimization therein.

The air-traffic level currently attained in Europe is around tens of thousands of flights per day and it is expected to be multiplied by a factor of two during the next 20 years [151]. Air traffic is at the core of the social and economic dynamism of our society, and an efficient air-traffic management has evidently a deep impact on the social, economic, environmental and industrial context. Air traffic safety is provided by a series of filters, each filter ensuring a level of traffic that can be handled by the next filter. For example, the CFMU (Central Flow Management Unit) provides aircraft take-off slots that guarantee that the density of aircraft does not exceed the control capabilities of air traffic controllers. Increasing levels of traffic, however, raise the problem of managing traffic in such a way as to increase the capacity of control in the air sectors. These issues lead naturally to optimization problems. Many questions such as "can we automatically solve all potential aircraft conflicts before take-off and en-route, and how can we make this optimization robust to uncertainty?" have so far not been satisfactorily answered and need to be addressed. Furthermore, the introduction of MTCD (Medium Term

Conflict Detection) in the context of recent European studies focuses the attention on medium-term en-route aircraft conflicts as a privileged area of investigation, that needs the development of suitable models and solution algorithms able to manage large-scale problems.

Some problems in ATM, like aircraft conflict avoidance, which is of particular interest in my research activity, naturally lead to MINLP models. MINLP allows one to simultaneously consider discrete (combinatorial) decisions and modeling the complex nonlinear processes characterizing ATM systems. A potential aircraft conflict occurs when aircraft are too close to each other according to their predicted trajectories. Detection and resolution of air traffic conflicts in tactical phases (i.e., en-route flights) are specially interesting and impact the workload of air traffic controllers. Air traffic control on the ground is still widely performed manually by air traffic controllers watching the traffic movements on a radar screen and giving instructions to pilots. The need for automatic tools to integrate human work, shifting responsibilities from the ground to the air, is evident.

It is worth remarking that in the ATM framework, the European project SESAR (Single European Sky ATM Research) [137] gives the guidelines to go towards the increase of the airspace capacity while significantly reducing its environmental impact, in order to evolve towards the future sky. It is in this context that most of the research on aeronautics and air traffic management is currently being performed. In France, this is the case for most of the research conducted in Toulouse (as one of the bases of the European aerospace industry and research, it is a well-established center of academic and industrial research focusing on aeronautics and air traffic management), and specially at ENAC (The French University of Civil Aviation), where I am currently working.

Chapter 1

Mixed-Integer Nonlinear Optimization and its Applications

In this chapter we present our contributions in the area of Mixed-Integer Nonlinear Optimization. We first briefly present mixed-integer non-linear problems and the main concepts at the basis of the (spatial) Branch-and-Bound algorithmic framework, which is the best know method to solve these problems to global optimality. It rests on some key ingredients, such as convex relaxations of the original problem. It is in this context that the contributions discussed in Sect. 1.2.1 and 1.2.2, concerning convex relaxations for polynomial problems, are proposed. More generally, they fall in the category of reformulations in mathematical programming. Reformulation techniques and contributions on automatic reformulation software are discussed in Sect. 1.3. These contributions, in the framework of deterministic global optimization, are not intended for a specific application, though they may constitute fundamental ingredients of efficient numerical methods for the solution of real-life problems. Sect. 1.4 and 1.5 are devoted to applications. The contribution discussed in Sect. 1.4, in particular, concerns Air Traffic Management, that has become a privileged domain of application in my research activity. Sect. 1.5 discusses a contribution in another domain of applications, namely the optimal design of electrical machines.

1.1 MINLP and global optimization

1.1.1 Mixed-integer non-linear optimization

Mixed-Integer Nonlinear Programming (MINLP) deals with the most general optimization problems, involving both continuous and discrete variables and nonlinear (in general, non-convex) objective and/or constraint functions. This kind of problems arises in countless applications from a wide range of domains, where simultaneously considering

discrete (combinatorial) decisions and nonlinear processes is crucial for modeling. Important sources of nonlinearity arise for example in chemical engineering [43, 3, 107], which is the source of the probably most influential research on MINLP, production planning [71], transportation [49], energy production [111], bioinformatics [89], electrical machines design [103, 102], and many other fields of real-life applications. A special attention is devoted in this report to MINLP problems arising in Air Traffic Management (see Sect. 1.4).

The general MINLP problem is written in the following form:

$$(\mathcal{P}) \begin{cases} \min\limits_{x \in \mathbb{R}^n} & f(x) \\ s.t. & g(x) \leq 0 \\ & x_i^L \leq x_i \leq x_i^U \\ \forall j \in Y & x_j \in \mathbb{Z} \end{cases} \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^m \to \mathbb{R}^n$, $x^L, x^U \in \mathbb{R}^n$, $Y \subseteq \{1, \cdots, n\}$. In general, functions f and g are assumed to be continuously differentiable, but not convex. Non-convexity is an additional source of difficulty: for non-convex problems, the continuous relaxation of $(\mathcal{P})$ might have local optima which do not coincide with global ones. Convex MINLPs represent a special case of MINLP problems, to which a lot of attention has been devoted by the mathematical programming community with significant results. Special subclasses of MINLP are given by Mixed-Integer Linear Programs (MILP), where functions f and g are linear, and Non-Linear Programs (NLP), where there are no integrality constraints on the variables.
(Non-convex) MINLP problems, being a generalization of MILP problems, are NP-hard (see [51, 113]). Beside their theoretical complexity, these problems are computationally challenging.

While research on mixed-integer linear optimization is quite advanced, MINLP is considered an emerging area that is likely to grow in the coming years, with an expected scientific impact on every domain where quantitative decision making based on discrete and nonlinear mathematical optimization arises. The attention for this emerging research area is shown not only by the increasing number of scientific publications, but also by dedicated cyber-infrastructures (http://minlp.org/) and conferences: IMA Hot Topics Workshop in 2008, European Workshop on MINLP in 2010, Exploratory Workshop on Mixed Integer Non-Linear Programming in 2010, and streams in Operations Research conferences (http://www.euro-2012.lt/streams#Mixed-IntegerNon-LinearProgramming) are just a few examples.

The focus of the research on MINLP is actually twofold. A significant part of this research mainly focuses on solution methods and algorithms with the aim of raising their efficiency (possibly to the level of algorithms for MILP) and providing highly viable

practical tools. Another part is more oriented towards real-life applications, aiming to propose MINLP-based models able to describe realistically even complex systems [125]. These two aspects are of course complementary and mutually reinforcing: theoretical advances are often validated by computational tests on real-life problems, while MINLP models benefit from advances on algorithm developments, this having an impact on their practical solution. The contributions presented in this chapter belong to both categories. The reciprocal impact may not be evident because the modeling proposed for the main application, arising in Air Traffic Management (see Sect. 1.4), does not contain elements allowing to employ results described e.g. in Sect. 1.2.1 and 1.2.2, though they may likely be exploited in different modelings of the same problem.

1.1.2 Deterministic global optimization of MINLP

Deterministic Global Optimization of MINLPs is primarily performed by means of Branch-and-Bound (BB)-type algorithms. BB algorithms perform a recursive search for the global optimum on a search tree, the nodes of which represent subproblems of the original problem, obtained by partitioning the original solution space. Several basic concepts are adapted from BB algorithms developed for the subclass of MILP problems. For a MILP, a continuous (linear) relaxation is first obtained by ignoring the integrality constraints on the variables. The solution of this relaxation gives a lower bound on the objective function optimal value of the original problem. Let x^r be the solution of the relaxation. One can then select an integer-constrained variable x_i which has a fractional value x_i^r in the relaxation solution and generate two subproblems, adjoining to the problem respectively the constraints $x_i \leq \lfloor x_i^r \rfloor$ and $x_i \geq \lfloor x_i^r \rfloor + 1$. The procedure to obtain a lower bound is referred to as *bounding*, while *branching* refers to the partitioning step which generates the two new nodes of the search tree. The algorithm is applied recursively to the subproblems, thus generating the tree. Under given conditions, a node can be removed from consideration (*pruning*). The algorithm termination occurs when all nodes have been solved or pruned, or some threshold is met for the difference between the best solution value found and the lower bounds on the still unsolved subproblems.

In the most general case, when some of the functions f and g are non-linear and non-convex, the bounding step becomes much harder. In this case, the continuous relaxation is a non-convex non-linear (NLP) problem which may have many local minima. Thus, to compute a global solution of the relaxed problem, a further relaxation step is usually performed, computing a *convex relaxation* of the original MINLP. Hence, at each step a lower bound for the objective function value is computed by solving globally a convex relaxation, and an upper bound is computed by solving locally the NLP problem. Additionally, branching is usually allowed on continuous as well as on discrete variables. This is done by taking a continuous variable x_i with current domain box $[\bar{x}_i^L, \bar{x}_i^U]$, choosing a

branching point $x_b \in [\bar{x}_i^L, \bar{x}_i^U]$, and generating the two new subproblems by considering $[\bar{x}_i^L, x_b]$ and $[x_b, \bar{x}_i^U]$.

This algorithm framework is also referred to as *spatial Branch-and-Bound*, where the term "spatial" refers to the recursive partition of the Euclidean space where the problem is defined into smaller and smaller regions. Summarizing, its basic steps are as follows (the reader is referred to [149, 150, 126, 42, 82]):

Bounding. Compute a lower bound by constructing and solving a convex relaxation of $(\mathcal{P})$. Compute an upper bound by solving the NLP locally.

Pruning. Discard a subproblem when:

(a) a global optimum for the node was found (pruning by optimality);

(b) the node was proved to be infeasible;

(c) a lower bound for the problem at the node has higher value than the incumbent, i.e., the value of the objective function evaluated at the current best optimum (pruning by bound).

Otherwise, do a branching step.

Branching. Generate two new subproblems of the original one whose feasible region is a subset of the feasible region of the original problem, by branching on continuous as well as on discrete variables.

Bounds tightening. Optionally, try to reduce the variables bounds.

The selection of branching variables and points, as well as that of the node of the search tree to be considered as the current subproblem to process, is usually made by means of heuristic procedures.

The spatial Branch-and-Bound (sBB) is an ϵ-approximation algorithm, meaning that the computed solution x^* is such that $f(x^*)$ differs at most by ϵ from the global optimal value of the objective function f [160].

It has to be considered as an algorithmic framework. Several interesting variants have been developed. In the Branch-and-Reduce variant [134, 135, 156] one attempts to reduce the domains of the variables, to improve the computation of lower bounds. The α-BB method [11, 5, 4] is based on a specialized technique for constructing under-estimators in the generation of the convex relaxations.

It is worth mentioning that an alternative successful approach for global optimization of MINLP is again in the Branch-and-Bound framework, but instead of relying on the *spatial* concept and on convex relaxations, it is based on interval arithmetic techniques. In the *interval Branch-and-Bound* [99, 100, 69], interval arithmetic rules are used to compute bounds. Recent versions also incorporate strategies based on the use of affine arithmetics [119]. The contributions described in the rest of this chapter are mainly in the context of spatial Branch-and-Bound. Therefore, a few aspects relevant to sBB are further discussed. Interval Branch-and-Bound is however to be considered as a promising

research direction for global optimization of the real-life applications on which I am currently focusing.

From the basic concepts of the sBB algorithm recalled above, it is evident that there are a few key ingredients that may have a significant impact on the algorithm efficiency. They have become crucial points of investigations themselves, generating an extensive literature. Convex relaxations of different kinds of functions or specialized problems, bound tightening techniques, branching rules, etc. are a few examples of topics in this literature.

1.1.3 The role of reformulations and convex relaxations

To solve a mathematical programming problem, it is often useful to re-cast it into a different formulation, in such a way that, for example, it is amenable to be solved by a given algorithm acting on a given formulation, or so that it exhibits interesting properties to be exploited in a given algorithmic framework. This is the case for a MINLP problem that one aims to solve via a spatial BB algorithm. Spatial BB-based solvers for MINLPs generally take their input problem in standard form, and are based on convex relaxations for the computations of bounds. In this context, our research focused, on the one hand, on the investigation of reformulations in mathematical programming and the development of software tools to carry out reformulations *automatically*, and on the other hand, on the study of convex relaxations for problems arising in polynomial programming. Our contributions are detailed in the next sections of this chapter.
We recall here a few basic concepts.

Basic definitions

A *reformulation* of a mathematical program P is a mathematical program Q which shares some properties with P (e.g., the set of optima) but is better than P in some sense. A more thorough discussion is presented in Sect. 1.3.

Convex relaxations are special cases of reformulations. A reformulation is a *relaxation* if one or more constraints are eliminated. Convex relaxations, in particular, are convex problems whose solution provides a (lower, for a minimization problem) bound on the objective function value at the optimum.

To define convex relaxations for nonconvex problems, and thus computing bounds for such problems, the concept of convex under-estimator is essential.

Definition 1.1. Let f be a function on a domain X. A *convex under-estimator* of f over X is a convex function c such that

$$\forall x \in X \quad c(x) \leq f(x).$$

The largest possible convex under-estimator is called convex envelope:

Definition 1.2. Let f be a function on a domain X. The *convex envelope* of f over X is

$$\forall x \in X \quad conv_{f,X}(x) = sup\left\{c(x) : c(x') \leq f(x') \quad \forall x' \in X \text{ and } c \text{ is convex}\right\}$$

Definition 1.3. 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.

MINLP standard-form reformulation

A way to construct a convex relaxation automatically by means of a particular reformulation (called MINLP standard-form [150],[Liberti et al., 2009]) was first proposed in [97] and is currently exploited in most existing sBB algorithms [135, 5, 150, 84, 156, 15]. It consists in generating a linearization of all nonlinear nonconvex terms arising in the objective and in the constraints, by using the following procedure:

- replace each nonconvex term by a linearizing variable w;
- adjoin to the formulation the variable w and its corresponding *defining constraint*, which has the form: $w =$ nonconvex term;
- replace each defining constraint by a convex under-estimator and a concave over-estimator (or convex/concave envelopes when they are available).

This iterative procedure for constructing a convex relaxation is symbolic rather than numeric, in the sense that it performs structural changes to the formulation of (1.1), adjoining variables and constraints, and replacing terms with variables and constraints with other constraints.

Factorable problems

Factorable problems are problems involving functional forms that can be written recursively, relative to basic operations like sums and products, using a finite number of elementary functions.

For example, a function f of three continuous variables x_1, x_2, x_3 with $x_3 \geq 0$, defined by $f = x_1 \times (x_2 + \sqrt{x_3})$ can be rewritten recursively introducing the new variables $w_1 = \sqrt{x_3}$, $w_2 = x_2 + w_1$ and $w_3 = x_1 \times w_2$.

For this kind of problems, it is possible to construct easily a convex relaxation by means of the MINLP standard-form reformulation.

These concepts will be used in Sect. 1.2.1 and 1.2.2.

1.2 Contributions in polynomial programming

This section discusses the ideas that we (together with some co-authors) developed concerning the generation of tight convex relaxations for the special class of polynomial problems.

A *polynomial programming problem* is a (possibly Mixed-Integer) Nonlinear Program in the following general form:

$$\begin{cases} \min\limits_{x} & f(x) & & \\ & g(x) & \leq & 0 \\ & x^L \leq & x & \leq x^U \\ \forall\, i \in Z & & x_i & \in \mathbb{Z}, \end{cases} \tag{1.2}$$

where $x, x^L, x^U \in \mathbb{R}^n$, $Z \subseteq \mathcal{N} = \{1, \ldots, n\}$, $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ are polynomial functions of x. In general, the feasible region of (1.2) can be a nonconvex set, or the objective function may be nonconvex on the feasible region. It is this most general case that has been primarily of interest in our research.

Often, problem (1.2) only includes continuous variables. However, because the relaxation procedures described in the rest of this section also holds in the presence of integrality constraints on the variables, and even when f, g include transcendental terms such as logarithm, exponentials and trigonometric functions, our results also apply to rather general MINLP polynomial problems.

1.2.1 Convex relaxations of quadrilinear monomials

In [Cafieri et al., 2010c], with J. Lee and L. Liberti, we undertook a computational as well as a theoretical investigation of the relative tightness of four relaxations of quadrilinear monomials (monomials of order 4), then also investigated in [Belotti et al., 2012b].

A *multilinear function* (on a vector space) is a function that is linear in each of its coordinates, i.e., fixing all but one of its arguments, the result is a linear function of the unfixed argument. 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, where multilinear monomials often arise. All polynomial functions are factorable (relative to multiplication), thus the symbolic reformulation described above (Sect. 1.1.3) for factorable problems applies to (1.2): high-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 low-order monomial is replaced by an additional variable, and an equality constraint defining the additional variable in terms of the monomial it

replaces, is adjoined to (1.2). This operation is carried out recursively, until the functions f, g are linear forms. At this stage, 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. The tightness of the resulting relaxation rests on the availability of tight convex relaxations of monomials of low degree.

Because of this, numerous efforts have studied convex relaxations of lower order product terms more generally in recursively factorable formulations. Convex envelopes in explicit form are currently known for concave/convex univariate functions [2, 148], bilinear terms [7, 97], trilinear terms [104, 105], univariate monomials of odd degree [92] and fractional terms [155]. More advanced practical techniques for generating tight convex envelopes computationally are given in [57, 58]. A natural generalization of bi- and trilinear functions are functions that are known to have *vertex polyhedral* convex envelopes. In [106], 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) were shown to have vertex polyhedral convex envelopes by [132]. An implication of this result is that many of the concepts mentioned in the preceding paragraph can be used for general mutilinear functions. The extension of such results to define convex envelopes for multilinear functions (and generalizations of them) has been discussed in [141, 153, 154, 155], among other references.

In [Cafieri et al., 2010c] and [Belotti et al., 2012b] we focused on quadrilinear monomials, i.e., functions in the form $x_1x_2x_3x_4$, with the aim to investigate tight convex relaxations, starting from the observation that this kind of monomials arises often in polynomial programs (in the mathematical formulation of several real-life problems, see [89, 79]) and it is the multivariate monomial of smallest degree for which the convex envelope is not explicitly known (via linear inequalities) in general. For bilinear monomials x_ix_j, for example, the known convex envelopes described by the McCormick's inequalities exist [7, 97]:

$$\begin{aligned} w_{ij} &\geq x_i^L x_j + x_j^L x_i - x_i^L x_j^L \\ w_{ij} &\geq x_i^U x_j + x_j^U x_i - x_i^U x_j^U \\ w_{ij} &\leq x_i^L x_j + x_j^U x_i - x_i^L x_j^U \\ w_{ij} &\leq x_i^U x_j + x_j^L x_i - x_i^U x_j^L. \end{aligned}$$

(where w_{ij} is the linearizing variable for $x_i x_j$), and for trilinear monomials a (significantly larger) number of inequalities defining the convex envelope have been described too [105, 104]. It is worth noticing that, very recently, 44 inequalities describing the convex envelope of quadrilinear monomials have been described in an M.Sc. thesis [13] for the simplest of the quadrilinear cases, the one having all bounds in the nonnegative orthant. It is not known how many cases there will be in total, but it is evident that the large number of inequalities and possible cases makes the task of implementing these envelopes in a computer program very hard and even impracticable. Also, very recently, an approach based on the duality theory to generate envelopes has been proposed in [31], with interesting computational results.

In [Cafieri et al., 2010c] and [Belotti et al., 2012b] we addressed 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. We specially addressed quadrilinear monomials, but our results apply more generally to factorable functions. Our contribution is twofold:

- We formally established, using a general method that is not limited to monomials, that any relaxation for 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 established by means of a computational assessment, based on the comparison of enclosing polytopes, which relaxation is strictly tighter.

Let us consider a quadrilinear term $x_1x_2x_3x_4$ and let $B = [x^L, x^U]$. Exploiting associativity of the product, the quadrilinear term can be rewritten in different ways, as products of monomials of degree two and three:

$$((x_1x_2)x_3)x_4, \quad (x_1x_2)(x_3x_4), \quad (x_1x_2x_3)x_4, \quad (x_1x_2)x_3x_4$$

(this, up to renaming the variables, exhausts the possibilities). 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, introducing linearizing variables, in many different ways according to the way associativity is applied. Specifically, relative to the considered term groupings, the following sets are obtained:

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

The known convex/concave envelopes for both bilinear [7, 97] and trilinear terms [105, 104] are then used to derive four convex relaxations of S^4: a bilinear envelope is exploited recursively thrice for the first two cases; a trilinear envelope followed by a bilinear envelope and a bilinear envelope followed by a trilinear envelope are exploited in the other two cases. A question then arises naturally: which corresponding relaxation is tighter? To answer this question, we investigated the problem from a theoretical as well as a computational point of view. The theoretical framework we provide is, in fact, more general to investigate relaxation strengths and can be applied to any factorable mathematical program in order to compare pairs of relaxations. However, it does not give an indication on the relative tightness of the considered relaxations, thus motivating a through computational assessment.

The theoretical result that we provided in [Cafieri et al., 2010c] can be summarized in the following way: *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 this may appear a quite intuitive result, because of the inherently recursive nature of factorable functions and of the fact that we deal with a recursive symbolic procedure for constructive the convex relaxation, we did not find it easy to prove this result formally. For this purpose, we used theoretical tools from the formal languages community. More precisely, we used a formal language to express the functions used in the objective and constraints of a mathematical program, and we defined a semantic of strings of this language to establish a formal comparison. The strings of this language are built recursively from operators and constant and variable symbols. These strings are in bijection with the "expression trees" mentioned in much of the sBB literature [84, 15], although we chose a presentation style following the formal-languages community [110]. Our theorem establishes that

Theorem 1.4. *For any relaxation of $x_1x_2\cdots x_k$ ($k \geq 3$) using any bilinear envelopes recursively, there is a relaxation employing also trilinear envelopes that is at least as tight.*

In particular, if applied to the convex relaxations of $x_1x_2x_3x_4$, the theorem establishes that S^{23} is at least as tight as S^{222} and $\tilde{S}^{222}$, and that S^{32} is at least as tight as S^{222}. Practical implications of Theorem 1.4 are important insofar as most sBB software employs the grouping leading to the slackest relaxation.

To evaluate the relative tightness of the four considered relaxation, we carried out extensive numerical experiments [Cafieri et al., 2010c]. The idea was to compare the four underestimation schemes mainly in terms of volume of the corresponding polytopes enveloping the nonconvex quadrilinear surface. This method of comparison, introduced in [81], is interesting because it is independent of any objective function. The polytopes

were projected on $(x, f(x) := x_1x_2x_3x_4) \in \mathbb{R}^5$ in order to compare the results, given that exploiting envelopes for bilinear and trilinear terms leads to an increased number of variables, and thus to polytopes belonging to $\mathbb{R}^7$ (S^{222} and $\tilde{S}^{222}$) and $\mathbb{R}^6$ (S^{23} and S^{32}). Then, their volumes were computed and extensive numerical tests were carried out. By this comparison of enclosing polytopes, which may be employed for the comparison of relaxations of other factorable functions, we got a significant indication of the strength of the considered relaxations. The results showed that the smallest values of volumes correspond to relaxations involving the composition of trilinear and bilinar envelopes (confirming Theorem 1.4), and in particular the best results for more than 80% of the considered instances were obtained using relaxation S^{23}. This is interesting because neither Theorem 1.4 nor basic intuition can help in establishing which relaxation is the tightest in practice. An example of the overall pattern in terms of variations of volumes of the enveloping polytopes for a selection of instances is shown in Figure 1.1, where the four lines correspond to the four linearizations and the points to the instances obtained by progressively tightening the bounds to simulate a BB. After tightening the bounds on a variable, there are no examples where one curve goes from far below to far above another: this suggests that tightening the bounds has a comparable effect on the different relaxations.

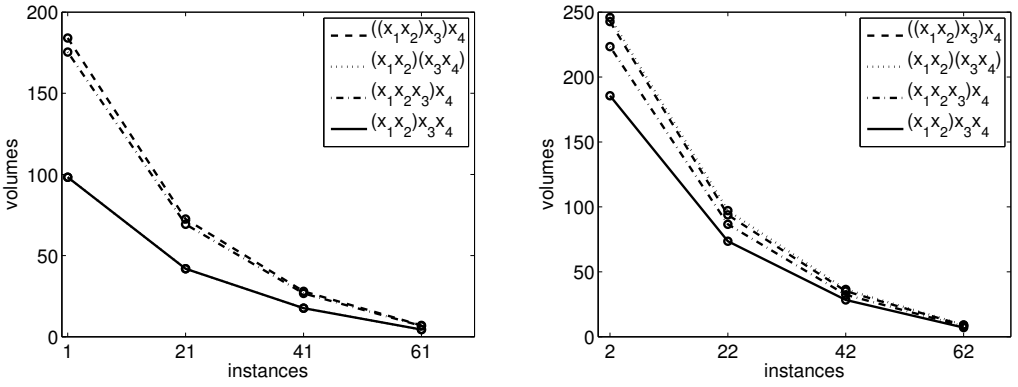


FIGURE 1.1: Volumes of enveloping polytopes corresponding to S^{222}, $\tilde{S}^{222}$, S^{32}, S^{23}.

These results were confirmed by numerical tests carried out on problems in the literature. The strength of the proposed relaxations were further investigated in a sBB algorithm in [Belotti et al., 2012b]. To that effect, we implemented the computation of the four relaxations for quadrilinear monomials in the general purpose solver for MINLP `COUENNE` [15], which is based on a sBB algorithm in the framework described in Sect. 1.1.2. Computational experiments were carried out running `COUENNE` on some instances of the Molecular Distance Geometry Problem (MDGP) [90] and confirmed the different impact of the considered relaxations on the performance of a spatial Branch-and-Bound. From the theoretical point of view, we were able to prove again Theorem 1.4

by means of a different development. In [Belotti et al., 2012b] we provided in fact an alternative proof of the same result, using concepts and methods from the community of formal languages in theoretical Computer Science, thus providing a result that is therefore also readable by a community other than that of Optimization.

1.2.2 Compact RLT-based relaxations

In [Cafieri et al., 2012c], with several co-authors, we have dealt again with reformulation techniques in MP and convex relaxations with the aim of improving the performance of sBB algorithms for global optimization of polynomial NLPs and MINLPs, tightening the bound computed by solving a convex relaxation at each sBB node. The target problem is a polynomial program subject to linear constraints, i.e., problem (1.2) subject to the constraint $Ax = b$, where A is a full rank $m \times n$ matrix and $b \in \mathbb{R}^m$.

The proposed reformulation belongs to the class of the well-known Reformulation-Linearization Technique (RLT) and extends to polynomial programs a special sub-class of RLT originally introduced by Liberti [91, 83] for quadratic problems. The RLT technique was introduced by H. Sherali and developed by Sherali and co-authors in a sequence of papers published from the 1980s onwards [145, 142, 1, 144, 143] with extensions to various type of nonconvex discrete and continuous problems. An extension of the RLT to polynomial programming is described in [139], and to more general factorable programming problems in [146]. The basic idea is to form new constraints by considering multiplications of bound factors (i.e., terms like $(x_i - x_i^L)$ and $(x_i^U - x_i)$) by linear constraint factors (i.e., the left-hand side of a constraint such as $g_i(x) - b_i$, where $g(x) = b$ or $g(x) \leq b$ are linear problem constraints). Since bound and constraint factors are always non-negative, so are their products: this way one can generate sets of valid problem constraints. The technique is then characterized by two steps: a *reformulation step* in which certain additional (nonlinear) valid inequality constraints are automatically generated, and a *linearization step* in which each product term is replaced by a single continuous (linearizing) variable w, called *RLT variable*. The corresponding RLT-based linear relaxation is obtained via this substitution upon dropping the defining constraints (w equal to the product term) from the formulation. This technique finds its practical limitation in the extremely large number of adjoined constraints. Usually the set of adjoined constraints is redundant (i.e., some of the constraints are linear combination of other ones), whereas other constraints are inactive. Some heuristic techniques [145, 140] were proposed to help filter out RLT constraints that are redundant. The special class of *reduced RLT* was introduced by L. Liberti [91] to address this drawback in the case of quadratic problems subject to linear constraints (and possibly nonlinear constraints), based on the observation that the presence of linear equality constraints in

the original problem allows the generation of only those linear RLT constraints that are guaranteed to replace a set of quadratic terms.

In [Cafieri et al., 2012c], we made two original contributions.

- We extended rRLT theory from quadratic to polynomial problems.
- We proposed a method to obtain a more compact (i.e.,with fewer constraints) convex relaxation. The compact version may be weaker than the rRLT one, but experiments show that the loss in tightness is greatly offset by the gain in CPU time required to solve it.

Extension of rRLT to polynomial problems

We recall here the basic steps of the reduced RLT extended to polynomial problems (more technical details are in [Cafieri et al., 2012c]). Let $Q = \{2, \dots, q\}$, $x_{j_1} \cdots x_{j_p}$ a monomial appearing in the original problem, where $p \in Q$, and let $J = (j_1, \dots, j_p)$ be the corresponding finite index sequence. An equivalence relation is defined to take into account symmetries arising in the monomials by commutativity, such that the set of equivalence classes $\bar{\mathcal{N}}^p$ is used to quantify over, when indexing added variables w_J, for all $p \in Q$.

The reduced RLT is extended to polynomial programming through the following steps: *Reformulation*: multiply the original linear constraints $Ax = b$ by all monomials $\prod_{\ell \leq p-1} x_{j_\ell}$; *Linearization*: replace each monomial by the corresponding added variable $w_{(J',j)}$, where $J' \in \bar{\mathcal{N}}^{p-1}$, obtaining the reduced RLT system (rRLTS), with $\mathbf{w}_{J'} = (w_{(J',1)}, \dots, w_{(J',n)})$:

$$\forall p \in Q, J' \in \bar{\mathcal{N}}^{p-1} \quad A\, \mathbf{w}_{J'} = b w_{J'}, \tag{1.3}$$

and adjoin the corresponding defining constraints $w_J = \prod_{\ell \leq |J|} x_{j_\ell}$.

By simple substitutions, rRLTS is proved to be equivalent to the *companion system*:

$$\forall p \in Q, J' \in \bar{\mathcal{N}}^{p-1} \quad A\, \mathbf{z}_{J'} = 0. \tag{1.4}$$

We were then able, in [Cafieri et al., 2012c], to extend to polynomial programs the main results holding for quadratic problems [91]: the companion system, with rank ρ, implies ρ of the rRLT defining constraints, which therefore can be dropped from the formulation without weakening it. More precisely, first let us note that, since (1.4) is a linear homogeneous system, there is a matrix M such that the companion system is equivalent to $Mz = 0$, the columns of which are indexed by sequences J. The columns of M can be partitioned in basic and nonbasic columns, say B and N the sets of associated indices. Two sets can then be defined: the set C containing the original linear constraints and all the defining constraints, and the set R_N containing the original linear constraints, the rRLT constraints and the defining constraints corresponding to $J \in N$.

The theorem we introduced states that the two sets are equivalent, or in other words, that rRLT constraints allow to obtain an *exact reformulation.*

Theorem 1.5. *For each partition B, N into basic and nonbasic column indices for the companion system $Mz = 0$, we have $C = R_N$.*

A convex relaxation for the reformulated problem is readily obtained by applying monomial convexification methods in the literature, such as the one presented in Sect. 1.2.1.

Compact rRLT

For any given linear system there is in general more than one way to partition the variables in basics and nonbasics. The idea was to choose the set N in such a way as to yield a more compact relaxation, i.e., with fewer constraints, without worsening the bound excessively. More precisely, the basic question is: how to choose N such that the monomial relaxations that are dropped define "large volumes", and are therefore more likely to be dominated by the relaxations of monomials in N? This question led (in [91] for quadratic problems) to the concept of *convexity gap* and to a method to determine which terms to retain in the formulation.

Given a function $f : X \subseteq \mathbb{R}^n \to \mathbb{R}$, the convexity gap is the volume of the convex relaxation of the set $S = \{(x, w) \mid w = f(x)\}$. For example, for a bilinear term $x_i x_j$, the convex relaxation of S is a tetrahedron easily obtained by exploiting the McCormick inequalities, and its volume is obtained by using the Cayley-Menger formula in 3 dimensions [70]. For a general monomial, one can observe that, by associativity and upon replacement by appropriate variables, for any monomial $\mu(x)$ of degree $p \in Q$ it is always possible to express the monomial as a product of multilinear factors and compute the convexity gap using the results holding for multilinear monomials. This approach is similar to the standard reformulation exploited by sBB implementations in view to obtain the convex relaxation of general monomials, and more specifically to the approach followed for quadrilinear monomials and described in Sect. 1.2.1.

If J is the (ordered) sequence of p variable indices appearing in a monomial $\mu(x)$, let V_J denote the convexity gap for $\mu(x)$ and let $V^{S,p} = \sum_{\substack{J \in S \\ |J|=p}} V_J$ for any set S, and $V^S = \sum_{p \in Q} V^{S,p}$. The idea is to select the basic/nonbasic sets of column indices of the companion system (1.4) in such a way that $V^{N,p}$ is minimized (equivalently, $V^{B,p}$ is maximized) for all p. There is however an added complication compared to the quadratic case: the volumes corresponding to monomial terms of different degree (e.g., V_{ij} and V_{ijk} for x_{ij} and x_{ijk} respectively) are expressed in different units of measure, so summing up V_J for Js of different sizes may not make much sense. This led us to define a

multi-objective problem:

$$\left.\begin{array}{rl} \forall p \in Q & \max V^{B,p} \\ M_B \text{ is a basis of (1.4)} & \end{array}\right\} \tag{1.5}$$

We then showed that (1.5) is equivalent to a single-objective problem:

Theorem 1.6. *Any solution B of (1.5) maximizing V^B also maximizes $V^{B,p}$ $\forall p \in Q$.*

The single-objective problem $\max\{v^B \mid M_B \text{ a basis of (1.4)}\}$ has a matroidal structure and can therefore be solved using a greedy algorithm.
It is worth noticing that a different treatment of the essentially the same concepts was recently presented in [138], with the notable difference that in [138] only a bases of A is employed instead of the (larger) companion system, and semidefinite cuts are also adjoined to the considered formulation.

Numerical results [Cafieri et al., 2012c] show that the compact rRLT linear relaxation (rRLT-C), obtained without considering the constraints relaxing monomial terms corresponding to basic columns of the companion system, generally requires less time to solve, and yields bounds that are not much worse than those given by the rRLT formulation: the cumulative bound worsening is 0.07% against a time improvement of nearly 40%. This is very encouraging with respect to enhancing spatial Branch-and-Bound algorithms implementing RLT-based relaxations.

1.3 Automatic reformulations

The contributions described in the previous sections were mainly around the concept of convex relaxations, which play an important role in deterministic Branch-and-Bound-based algorithms for global optimization. Such a work is however in the framework of the wider research topic of *reformulations*. Reformulations in Mathematical Programming (MP) have been recently systematically studied and classified, also appearing as an autonomous domain in MP. Definitions and systematics of reformulations have been proposed by L. Liberti in [86, 85], motivated by the observation that there was not a common framework to study reformulations, though they are widely used in mathematical programming. My activity as post-doctoral researcher at École Polytechnique in Paris was in the context of an ANR-funded project on reformulations in MP, leading me to get interested in this topic. With Leo Liberti, we specially worked attempting to move some steps in the direction of *automatic* reformulations. Our book chapter [Liberti et al., 2009], like [86], aims to provide a common framework, though being much more computationally oriented, providing symbolic algorithms and a first description of the implementation of a software framework for automatic reformulations.

Several different definitions of reformulation have been proposed. The one we gave in Sect. 1.1.2 is a basic informal definition. Basically, one expects that reformulations

keep properties like the set of optima, and exhibit better properties for example from the point of view of algorithms application. Reformulations are indeed important with respect to the choice and efficiency of the solution algorithms, not only because typically solvers require their input problems to be cast in a standard form, but also because solver performances can be different depending on the formulation used. An interesting example is provided by the reduced RLT reformulation described in Sect. 1.2.2: given, for example, a quadratic problem with linear equality constraints, it is reformulated into a different quadratic problem with more linear equality constraints (rRLTS) and fewer quadratic terms. The original and the reformulated problems both belong to the same class of optimization problems, but a spatial Branch-and-Bound performs better on the reformulation, the convex relaxation of the reformulated problem being tighter.

The idea of L. Liberti was to propose an unified data structure for mathematical programming formulations, categorizing reformulations in essentially four types: *opt-reformulations*, which preserve all optimality properties; *narrowings*, which preserve at least one global optimum; *relaxations*, which are based on dropping constraints, variable bounds or types; and *approximations*, which are one of the above types "in the limit".

Building on these basic elements, in [Liberti et al., 2009] we mainly focused on the computational aspects, aiming to provide tools to carry out a number of reformulations automatically. *Automatically* in this context means that, given a formulation P, a reformulation solver has to provide (performing symbolic computations) a new formulation Q of the same problem in a more convenient form, in such a way that the user that aims for example to use an optimization algorithm do no need to manipulate formulations nor to be aware of solution algorithm details (a few capabilities of this type are available in mathematical programming language environments like AMPL[48] and GAMS[26]). This effort to provide automatic reformulation tools led us to the development of a software framework for reformulations in MP: ROSE - Reformulation Optimization Software Engine. This software is also described in [Liberti et al., 2010]. ROSE provides symbolic (as opposed to numerical) methods for manipulating MPs. It can perform basic and complex symbolic analyses and manipulation tasks on all formulation elements, including all expressions appearing in objective(s) and constraints in a MINLP problem. ROSE consists of around 50Klines of GNU C++ code and is currently distributed through COIN-OR [93]. ROSE's structure is based on the structure of a software previously developed by L. Liberti and of which ROSE is the direct descendant. ROSE consists of a simple modular architecture based on two main classes (`Problem` and `Solver`) and a separate library (Ev3) for storing and manipulating expressions. The overall architecture is depicted graphically in Fig. 1.2. The client (either the user or AMPL) constructs and configures a problem, selects and configures a solver, then solves a problem using the selected solver, and finally collects the output from the problem. In our research, we currently use ROSE's reformulation capabilities with

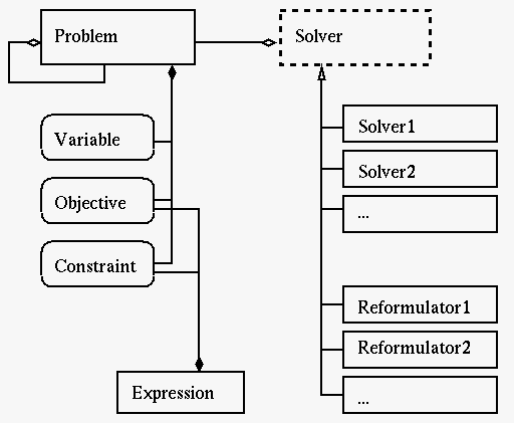


FIGURE 1.2: ROSE architecture. Rectangles indicate `class`es (with dashed meaning virtual), rounded boxes indicates `structs`, relation links conform to UML: void diamonds indicate aggregation (to maintain a reference of), filled diamonds indicate composition (to maintain a copy of), triangles indicate inheritance.

AMPL's considerable set of numerical solvers in order to obtain solutions of complex MINLPs. Examples of reformulators implemented in ROSE include a Smith reformulation (`Rsmith`) [150] (MINLP standard-form reformulation, see Sect. 1.1.3), which isolates all the nonlinearities of the problem in constraints with a simple structure (these are then replaced by appropriate convex relaxations), and several relaxation reformulators (chained to the `Rsmith`), such as `Rconvexifier`, which produces a convex relaxation of several nonlinear functions, and `RQuarticConvex`, which produces convex relaxations of quadrilinear monomials. In particular, for the investigation of convex relaxations of quadrilinear terms [Cafieri et al., 2010c, Belotti et al., 2012b] (see Sect. 1.2.1), ROSE was used both to produce convex relaxations and to automatically write input data to other software packages (e.g. `cdd` [50]).

An example of automatic reformulations carried out by ROSE may probably help to understand better its capabilities. Let us consider the following optimization problem whose objective function contains four nonlinear terms. In order to ease the reading, we use an intuitive description format for MINLPs (note that the symbol '<' stands here for '$\leq$' and that we use an explicit boundary ($1e^{30}$) for dealing with infinity). ROSE is able to find a convex relaxation for the problem using the convexifier reformulator (`Rconvexifier`).

Original Problem

```
# ROSE problem: convexifier
# Problem has 3 variables and 1 constraints
# Variables:

variables = -1 < x1 < 1 / Continuous,
-2 < y2 < 3 / Continuous,
1 < t3 < 2 / Continuous;

# Objective Function:
objfun = min [ (2*x1^2)+(y2^3)
              +((x1)*(y2))+((x1)/(t3)) ];
# Constraints:
constraints = [ 2 < (x1)+(y2) < 1e+30 ];
```

ROSE Reformulation

```
# ROSE problem: convexifier
# Problem has 9 variables and 18 constraints
# Variables:

variables = -1 < x1 < 1 / Continuous,
-2 < y2 < 3 / Continuous,
1 < t3 < 2 / Continuous,
0 < w4 < 2 / Continuous,
-8 < w5 < 27 / Continuous,
-3 < w6 < 3 / Continuous,
-1 < w7 < 1 / Continuous,
-12 < w8 < 33 / Continuous,
0.5 < z9 < 1 / Continuous;

# Objective Function:
objfun = min [ w8 ];

# Constraints:
constraints = [ 2 < (x1)+(y2) < 1e+30 ],
[ 0 < (w4)+(w5)+(w6)+(w7)+(-1*w8) < 0 ],
[ -2 < (4*x1)+(w4) < 1e+30 ],
[ -2 < (-4*x1)+(w4) < 1e+30 ],
[ -0.5 < (2*x1)+(w4) < 1e+30 ],
[ -0.5 < (-2*x1)+(w4) < 1e+30 ],
[ -2 < (-3*y2)+(w5) < 1e+30 ],
[ -54 < (-27*y2)+(w5) < 1e+30 ],
[ -1e+30 < (-6.75*y2)+(w5) < 6.75 ],
[ -1e+30 < (-12*y2)+(w5) < 16 ],
[ -2 < (2*x1)+(y2)+(w6) < 1e+30 ],
[ -3 < (-3*x1)+(-1*y2)+(w6) < 1e+30 ],
[ -1e+30 < (-3*x1)+(y2)+(w6) < 3 ],
[ -1e+30 < (2*x1)+(-1*y2)+(w6) < 2 ],
[ 0.5 < (-0.5*x1)+(w7)+(z9) < 1e+30 ],
[ -1 < (-1*x1)+(w7)+(-1*z9) < 1e+30 ],
[ -1e+30 < (-1*x1)+(w7)+(z9) < 1 ],
[ -1e+30 < (-0.5*x1)+(w7)+(-1*z9) < -0.5 ];
```

The reformulation process is performed in various steps. The first step consists in reformulating the problem to the Smith standard form: each nonconvex term in the objective function is replaced by an added variable w and defining constraints of the form $w = nonconvex\,term$ are added to the problem. The objective function of the reformulated problem is one linearizing variable only, that is the sum of all the added variables, and a constraint for this equation is also added to the problem. Then, each defining constraint is replaced by a convex under-estimator and concave over-estimator of the corresponding nonlinear term. In particular, the term `2*x1^2` is treated as a

convex univariate function `f(x)` and a linear under-estimator is obtained by considering five tangents to `f` at various given points, an over-estimator is obtained by considering the secant through the points `(x1^L,f(x1^L)),(x1^U,f(x1^U))`, where `x1^L,`and `x1^U` are the bounds on `x1.`For the term `y2^3,` where the range of `y2` includes zero, the linear relaxation given in [92] is used. McCormick's envelopes are considered for the bilinear term `x1*y2.`The fractional term is reformulated as bilinear by considering `z=1/t3` and McCormick's envelopes are exploited again. Finally, the defining constraints are removed, obtaining the final reformulation (of the relaxation type).

1.4 ATM application: mixed-integer optimization for aircraft conflict avoidance

Problems arising in Air Traffic Management (ATM) are a privileged area of investigation at École Nationale de l'Aviation Civile, and became my favorite field of application since the beginning of my work in this University. The ATM context has been drawn in the Introduction. We particularly focus on the problem of aircraft conflict avoidance, where mixed-integer optimization naturally arises. A different approach, based on optimal control techniques, is presented in Chapter 2.

Aircraft sharing the same airspace are said to be potentially in *conflict* when they are too close to each other according to their predicted trajectories, i.e., their relative horizontal and vertical distances do not both satisfy two given safety-distance constraints. Detection and resolution of aircraft conflicts, also referred to as *aircraft deconfliction*, is one of the most crucial issues in Air Traffic Management to guarantee air traffic safety. As mentioned in the Introduction, deconfliction is in fact one of the problems that urgently need to be addressed to ensure a higher level of automation in ATM and consequently more efficiency and safety, as the traffic is rapidly growing. Also, these problems still deserve investigation from both the identification of suitable mathematical models and the development of efficient and reliable algorithms and solution methods. The main challenge is to propose mathematical formulations that are able to model the complex choices characterizing the target problems without assuming any unrealistic simplifying hypoteses, and that are amenable to be solved by efficient algorithms. My work in this context is primarly focused on devising mathematical programming formulations from mixed-integer optimization and on designing deterministic-based solution approaches.

Modeling aircraft conflict avoidance is strictly dependent on the strategy chosen to solve conflicts, i.e., to keep a separation between aircraft. The most commonly exploited way is based on the idea of achieving separation through changing either the trajectory (heading) or the flight level of the aircraft involved in a conflict. This kind of separation maneuvers is the one usually exploited by air traffic controllers. Another way is based

on the idea of separating aircraft by slightly changing their speeds but keeping the predicted trajectories. More precisely, according to the European ERASMUS project [24], the speed regulation must occur in a reasonable small range (namely, from -6% to +3% of the original speed), to perform a *subliminal control*, which is not even perceived by air traffic controllers. It is on this last strategy that our work is based.

An optimization problem arises from the need to perform conflict avoidance while deviating as little as possible from the original aircraft flight plan, i.e., minimizing the impact of the separation maneuvers on the flight efficiency. To this aim, various solution strategies have been proposed for the corresponding optimization problem (see e.g.[77]). In [Cafieri and Durand, 2012] (see also [29, 28]), the aim was to use mathematical programming and specifically mixed-integer optimization, proposing a new MINLP model and deterministic global solution approaches as opposed to the widely used evolutionary computation [39, 36, 38, 37]. Recent advances in mixed-integer linear and nonlinear programming are, in fact, opening new perspectives for aircraft deconfliction, as showed by the literature in this area, started with contributions like [124] and very recently enriched by interesting publications [8, 9, 130, 131, 164], furtherly motivating our work.

MINLP formulations appear in fact to be the natural candidates for the addressed ATM problems, where the need for modeling logical choices suggests the simultaneous presence of mixed (continuous-integer) variables, and nonlinear constraints arise from separation condition modeling. In [Cafieri and Durand, 2012], we propose a very general modeling for aircraft deconfliction, whose main ideas are the following.

First, the aircraft separation condition, which represents the main constraint in a mathematical programming model for deconfliction, is reformulated in such a way to eliminate the time t from the equation. Starting from its basic expression (vectorial form) for aircraft i and j flying at the same flight level $||\mathbf{x}_{ij}^r(t)|| \geq d$, where d is the minimum required separation distance (usually, 5 NM, with 1 NM (Nautical Mile) = 1852 m) and $\mathbf{x}_{ij}^r(t)$ is the relative distance between aircraft i and j at the instant time t, it is reformulated to:

$$(x_{ij}^{rd})^2 - \frac{(\mathbf{v}_{ij}^r \mathbf{x}_{ij}^{rd})^2}{(v_{ij}^r)^2} - d^2 \geq 0. \tag{1.6}$$

This expression is obtained, under the hypothesis that aircraft speed changes occur instantaneously at a given instant time and hence uniform motion laws can be applied, by rewriting $\mathbf{x}_{ij}^r(t)$ in terms of the relative initial position of the aircraft x_{ij}^{rd} and the product of their relative speed $\mathbf{v}_{ij}^r$ by the time, and by substituting the expression of the minimum instant time t_m for which the separation is attained:

$$t_m = -\frac{\mathbf{v}_{ij}^r \mathbf{x}_{ij}^{rd}}{(v_{ij}^r)^2}. \tag{1.7}$$

In this way, with (1.6) not depending on the time variable t anymore, the number of constraints is reduced with respect to more classical approaches based on time discretization. Nevertheless, in our model (1.6) is imposed in several time windows, as detailed in the following.

Second, to obtain a very general model, where no conditions are imposed on the order and on the instant time of execution of the separation maneuvers, the main idea is to deal with the different time windows where aircraft fly with their original (known) speed v or with a changed speed $v+q$, where q represents a possible positive or negative speed change. Each aircraft k can modify its speed at any instant time t_{1k} during its trajectory and go back to its original speed at any instant time t_{2k}. Instant times t_{1k} and t_{2k} are therefore unknown for each aircraft, and for each pair of conflicting aircraft the order of the respective instant times of speed change is not known. For each pair of aircraft, by considering all possible permutations of these instant times, 6 different time configurations are obtained, each one characterized by 5 time intervals. An illustrative picture for a pair of aircraft is given in Fig. 1.3, where T represents the time horizon (usually around 20-30 minutes) during which the air traffic on an air sector is observed and potential conflicts are to be solved.

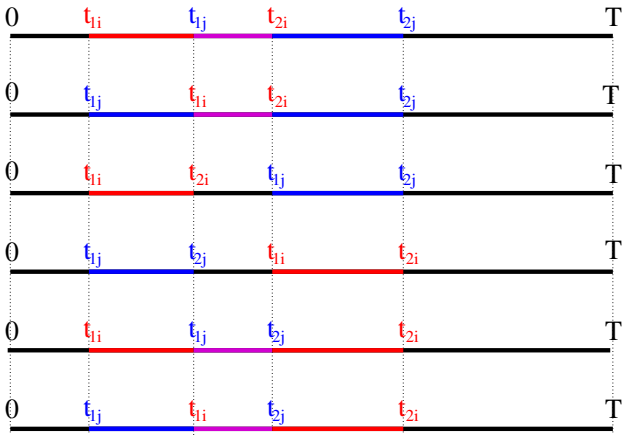


FIGURE 1.3: Time configurations in the proposed mathematical programming model for conflict avoidance of a pair of aircraft i (in red) and j (in blue)

The decision variables in the proposed model are: variables q_k, expressing the speed change for each aircraft k, instant times t_{1k} and t_{2k}, and variables used to handle time configurations and time windows. Binary variables are used to state the order of instant times for each time configuration, and additional continuous variables are used to express, in each time window, the initial position for each aircraft and relative distances and relative speeds for each pair of aircraft. Several objective functions are possible, depending on the optimization that one aims to perform. With the aim of deviating as less as possible from the original flight plan, we minimize the sum, over the set of aircraft, of speed changes together with the length of the time interval during which the

speed changes occur. The constraints are used to handle time configurations and time windows as well as to express aircraft separation conditions in each time window. In particular, the condition (1.6) has to be imposed for each pair of aircraft, in each of the 5 time intervals, and for each time configuration:

$$\forall i,j,\ \forall h \in \{1,\dots,5\},\ \forall \ell \in \{1,\dots,6\} \quad \left(y_{lh} y_{rh} \left((x^{rd}_{ijh})^2 - \frac{(v^r_{ijh} x^{rd}_{ijh})^2}{(v^r_{ijh})^2} - d^2 \right) \right) \geq 0 \tag{1.8}$$

where y are binary variables used to check if $t_m \in [t_s, t_{s'}]$ for each time interval $[t_s, t_{s'}]$. The proposed model is therefore a MINLP problem which is computationally challenging, due to the large number of (binary and continuous) variables and of constraints used to handle time configurations and time intervals, in particular the nonlinear nonconvex constraints used to impose aircraft separation.

We attempted to solve the deconfliction problem to global optimality by means of a deterministic solver for global optimization. Using a general-purpose solver for MINLPs, namely `COUENNE` [15], which implements a spatial Branch-and-Bound based on convex relaxations (see Sect. 1.1.2), we were able to solve problems involving up to $n = 6$ aircraft. Specifically, we considered as a testbed a set of n aircraft in 2-dimensional space, placed on a circle of a given radius and pointing toward the center of the circle. This kind of test problem, although it does not correspond to a realistic situation, represents a good trade-off between simplicity of illustration and difficulty of resolution (all aircraft are in conflict with each other). See Figure 1.4.

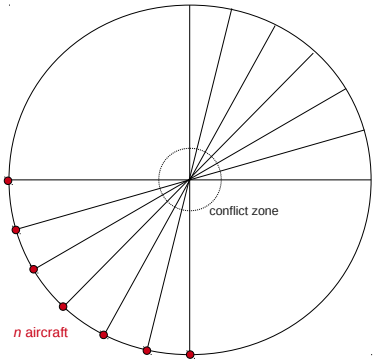


FIGURE 1.4: n conflicting aircraft flying towards the center of a circle.

Increasing the number of aircraft n, the number of variables (in particular binary) and of constraints largely increase. Thus, high memory and time requirements do not allow us to obtain optimal solutions for larger dimensions of the problem by a standard sBB algorithm. This in fact leaves room for further investigation of deterministic global methods which possibly take advantage from the characteristics of the target problem. This is in our perspectives of research in this domain.

The difficulty of obtaining global exact solution by standard sBB led us to investigate alternative approaches, where eventually the global optimality is forsaken in exchange for the computational efficiency. We have chosen to keep a deterministic approach at least *locally*, proposing a solution strategy where exact solutions are computed locally on subproblems of the original problem. The idea is to decompose the overall problem into subproblems involving only a small number of aircraft and to perform deconfliction exactly on these subproblems, then combining all local solutions. The overall procedure is thus a heuristic, based on Mathematical Programming and local exact solutions, in a framework sometimes defined as *matheuristic*. The algorithm is therefore very different from heuristic algorithms usually proposed to solve the considered ATM problem, mainly belonging to the area of evolutionary computation. The proposed algorithm is exact on the subproblems (locally optimal) and is globally optimal if all conflicts are solved when combining local solutions. Moreover, it is based on a few basic steps which can be eventually developed in different ways, such as local-solution computations and local searches. In this sense, the algorithm may be intended as a general algorithmic framework tailored on the problem. The problem decomposition is based on the concept of *cluster*, defined as the transitive closing on conflicting pairs of aircraft [59]. An illustrative example is given in Figure 1.5.

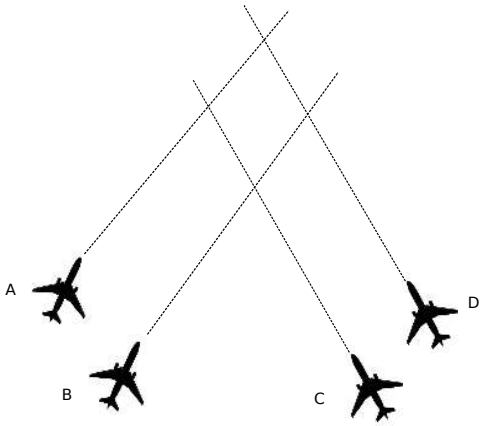


FIGURE 1.5: Example of aircraft cluster A,B,C,D
(A in conflict with B and C, and B with D).

A sketch of the proposed algorithm is given in Alg. 1. Its main ingredients are the following:

- At each step, a solution of *ncl* deconfliction problems (aircraft clusters) is performed by using an exact solver. Then, the obtained results are combined and the procedure is iterated until all conflicts are solved;
- Each iteration is performed while preserving as much as possible the information outcoming from the previous local solutions (this is specially taken into account in the update of aircraft speeds);

- A simple local search is performed to update the aircraft speeds, testing a number of candidates in the neighborhoods of the initial speed values and choosing the one that minimizes a measure of the conflict severity;
- When speeds are modified, the sharp bounds imposed by the ERASMUS directives are checked and speeds are adjusted accordingly.

Algorithm 1 Aircraft deconfliction

/* n = number of aircraft; ncl = number of aircraft clusters; v_i = speed of aircraft i */
/* $maxit$ = number of allowed iterations; $nconfl$ = number of aircraft conflicts */
Require: $n, ncl, v_i \; \forall i = \{1, \dots, n\}$
compute $nconfl$, $it \leftarrow 0$
while ($nconfl > 0$ and $it < maxit$) **do**
  $it \leftarrow it + 1$
  for all $k \leq ncl$ **do** solve the deconfliction problem k ($\rightarrow$ compute new speeds $\bar{v}_i$)
  compute $nconfl$
  if $nconfl > 0$ **then**
    for all conflicting aircraft j **do**
      if $\bar{v}_j > v_j$ **then** randomly increase $\bar{v}_j$
      else randomly decrease $\bar{v}_j$
      choose $\bar{v}_j$ to have $\min f = \sum_j$ (max violation of separation constraints)$/nconfl$
    end for
  end if
  check if $\forall i \leq n \; \bar{v}_i \in [-6\%v_i, +3\%v_i]$
  for all $i \leq n$ **do**
    if $\bar{v}_i \notin [-6\%v_i, +3\%v_i]$ **then** randomly change $\bar{v}_i$ to have $\bar{v}_i \in [-6\%v_i, +3\%v_i]$
  end for
end while
Return: final speeds, objective function value

Numerical results obtained again considering the problem of n aircraft on the circle are very promising: problems with up to $n = 10$ aircraft (45 conflicts at the same time) are solved in reasonable time, efficiently performing a subliminal control on aircraft speeds.

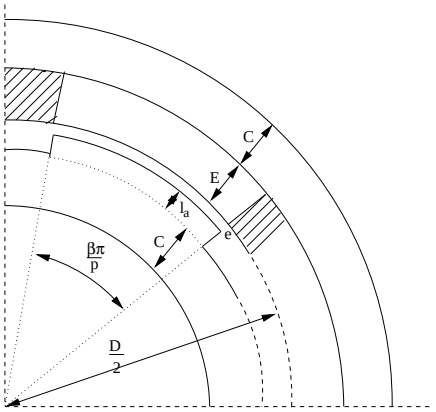
1.5 Application to the optimal design of electrical machines

Optimal design problems aim to find structures, or shapes, optimizing some given criteria, for example minimizing a certain cost functional, while satisfying given constraints. They are typical of engineering applications where one has to design complex structures, e.g. airplane wings, automobile bodies, electronic circuits, etc. In the last decades, the progresses made in mechanical modeling, numerical analysis and optimization, and the increased power of computers, led to a wide use of simulation and computer calculation in optimal design. Interdisciplinary teams often obtain very good achievements by making it profitable to bring their different expertise. This is the case in Toulouse for the work made on the optimal design of electrical machines by the group of the LAPLACE

laboratory for the physical and engineering part, and F. Messine and co-workers for the mathematical optimization part, as the product of a longstanding collaboration. In fact, it is thanks to F. Messine that I got interested in this kind of applications.

A large literature exists about the design of electromagnetical actuators based on optimization algorithms and analytical models [101, 99, 66, 122, 129, 147, 171]. These problems can be formulated as mixed-integer constrained optimization problems, where integer variables represent quantities that can assume only discrete values and nonlinearity often arises in constraints coming from mechanical considerations. Thus, the MP formulation is usually a MINLP program. In [Cafieri et al., 2012e], we chose to analyze the problem from the point of view of a designer who wants to use an optimization solver to find the optimal configuration in the design of an electrical machine without taking care of the details of the implemented optimization algorithm. Deterministic global optimization methods have been proposed for such design problem, based on interval analysis [99, 101] and other global solvers can be applied as well. However, similarly to the ATM problem described in Sect. 1.4, the MINLP problem may be very difficult to solve to global optimality by standard solvers. Furthermore, powerful solvers are not always publically available (this is the case for interval analysis based solvers) or are not easy to use for non-specialists. Based on the observation that in practice, in the context of engineering applications, one usually prefers to resort to well-known and easy-to-use local optimization solvers like the MatLab's `fmincon` function (Mathworks) [112, 122, 129, 171], we tried to analyze the problems that a designer may have to face when using this kind of solvers. Thus, rather than considering a classical MINLP approach, we investigated, based on the theory of reformulations in mathematical programming (see Sect. 1.3), the efficiency and reliability of the widely used solver `fmincon` when handling different mathematical formulations (in a multistart setting, for a wider exploration of the feasible domain), with the aim of providing guidelines for designers in practical engineering applications.

We focused on the design of a simple slotless electrical rotating permanent magnet machine, presented in Fig. 1.6, whose descriptive equations comes from approximations of Maxwell's equations (taken in the quasi-static mode) and mechanical considerations.



$$\Gamma_{em} = \frac{\pi}{2\lambda}(1-K_f)\sqrt{k_r\beta E_{ch}E}D^2(D+E)B_e$$
$$E_{ch} = AJ_{cu} = k_r E J_{cu}^2$$
$$p = \frac{\pi D}{\Delta_p}$$
$$K_f = 1.5p\beta\frac{e+E}{D}$$
$$C = \frac{\pi\beta B_e}{4pB_{iron}}D$$
$$B_e = \frac{2l_aP}{D\ln\left(\frac{D+2E}{D-2(l_a+e)}\right)}$$

where Γ_{em} is the *electromagnetic torque*, $D(m)$ is the *bore diameter*, λ the *diameter over length ratio*, $E(m)$ the *winding thickness*, β the *polar arc factor*, k_r a *coefficient of occupation*; the global heating up of the winding is rather roughly modeled by E_{ch}, $J_{cu}(A/m^2)$ is the *current areal density*, p is the *number of pole pairs*, Δ_p the *polar step*, K_f is a *semi-empiric magnetic leakage coefficient*, $e(m)$ is the *thickness of the mechanical air-gap*, $C(m)$ is the *thickness of yoke*, B_{iron} is the *magnetic field in the iron*, $B_e(T)$ is the *no-load magnetic radial flux density*, $l_a(m)$ is the *thickness of the permanent magnets* and P the *magnetic polarization*.
A few entities appearing in the equations, Γ_{em}, P, k_r, B_{iron}, E_{ch} and Δ_p, have fixed values as in [122], while other parameters, like D, λ, $l_a(m)$, E, C, β, B_e, J_{cu}, K_f, e and p can vary inside given intervals.

FIGURE 1.6: The considered permanent magnet machine

Starting from a general mathematical programming formulation like:

$$\mathcal{P}: \begin{cases} \min\limits_{\substack{x\in\mathbb{R}^n \\ y\in\mathbb{R}^m}} & f(x,y) \\ s.t. & \\ & g_i(x,y) \leq 0, \forall i \in \{1,\cdots,p\}, \\ & h_i(x,y) = 0, \forall i \in \{1,\cdots,q\}, \\ & y_i = A_i(x, y_{J_i}), \forall i \in \{1,\cdots,m\}, \\ & \underline{x_i} \leq x_i \leq \overline{x_i}, \forall i \in \{1,\cdots,n\}, \\ & \underline{y_i} \leq y_i \leq \overline{y_i}, \forall i \in \{1,\cdots,m\}. \end{cases}$$

where $J_i \subseteq \{1,\cdots,m\} \setminus \{i\}$ (y_i depends explicitly or implicitly on x by recursive calls to A_j functions and there is no cycle in the definition of y_i, hence $y_i = A_i(x, y_{J_i}) = A_i^R(x)$; as a vectorial notation, $y = A^R(x)$), we investigated the impact of reformulations

obtained by replacing the occurrences of y_i in $(\mathcal{P})$ by $A_i^R(x)$:

$$\mathcal{R}: \begin{cases} \min\limits_{x\in\mathbb{R}^n} & f(x, A^R(x)) \\ s.t. & \\ & g_i(x, A^R(x)) \leq 0, \forall i \in \{1, \cdots, p\}, \\ & h_i(x, A^R(x)) = 0, \forall i \in \{1, \cdots, q\}, \\ & \underline{x_i} \leq x_i \leq \overline{x_i}, \forall i \in \{1, \cdots, n\}, \\ & \underline{y_i} \leq A_i^R(x) \leq \overline{y_i}, \forall i \in \{1, \cdots, m\}. \end{cases}$$

Problem $(\mathcal{R})$, reformulation of $(\mathcal{P})$, is obtained by removing m variables (all the variables y) and changing m equality constraints ($y_i = A_i(x, y_{J_i})$) to $2m$ inequality constraints ($\underline{y_i} - A_i^R(x) \leq 0$ and $A_i^R(x) - \overline{y_i} \leq 0$). Formulations $(\mathcal{P})$ and $(\mathcal{R})$ are mathematically equivalent: all feasible solutions of $(\mathcal{P})$ are solutions of $(\mathcal{R})$ and reciprocally, and a global optimal solution for one problem is a global solution for the other.

In practice, we obtained 6 distinct but equivalent formulations of the considered design problem, varying in size from 7 to 11 continuous variables, from 3 to 7 equality constraints and from 0 to 6 inequality constraints. In these formulations, when the number of variables increases, the nonlinearities of the equations decrease yielding simpler optimization problems but with more variables. Since the solver performance is roughly directly proportional to both the number of variables and the number of nonlinearities in the objective and constraints, a natural trade-off situation arises.

We computationally investigated the impact of the 6 different formulations on the performance of `fmincon`. This solver being a local solver, also dependent on the choice of the starting point, we considered a standard multistart approach. Numerical results gave interesting indications to the designer. Comparing results in terms of percentage of local minima and percentage of best local minima found, best and worst values of the local minimum found, best and average CPU time, the impact of the different formulations is evident. We recall that the problem is mixed-integer, with an integer variable given by the number p of pole pairs, in $\{1, \ldots, 10\}$. The problem was first treated as a continuous problem, fixing the value of p, to obtain continuous formulations that can be solved by the considered optimization solver. Then the case where p is free was handled adjoining polynomial constraints. Again, the formulations performed differently. We found these remarks interesting from the point of view of a designer, who may be unaware of the impact of formulation differences on the optimization solver of choice.

Chapter 2

Nonlinear Continuous Optimization and its Applications

Nonlinear Programming (NLP) is the subclass of MINLP that includes problems where nonlinearities arise in the objective and/or the constraints and all variables are continuous. NLP problems are fundamental problems in optimization and specific numerical methods have been developed for their solution, usually local optimization methods, opposed to global methods discussed in Chap. 1 (of which they often constitute an important part). NLP techniques are also particularly efficient in solving optimal control problems.

In this chapter, we present contributions on two specific topics falling in the area of NLP. The first stage of my research activity (corresponding to my Ph.D. and subsequent work) focused on Interior Point methods for Quadratic Programming and specially on their linear algebra kernels. The presented contributions concern the convergence properties of an Interior Point algorithm (Sect. 2.1) and the main issues arising in the iterative solution of the linear system arising at each iteration of such an algorithm (Sect. 2.2). Recently, I got interested again in NLP methods in the context of an optimal control approach for a problem arising in Air Traffic Management, which is the topic of a Ph.D. thesis that I am supervising. The related contribution concludes the chapter (Sect. 2.3).

An exhaustive discussion of both theoretical and practical aspects of nonlinear programming can be found in some reference books, e.g. [121, 20].

2.1 Interior Point methods

Interior Point methods (IPs) represent one of the main classes of numerical methods for NLP. A recent interesting survey by J. Gondzio [55] traces the main characteristics of Interior Point methods and the developments of the research on these methods after about 25 years from their first description [72]. The focus is specially on polynomial

complexity, which make them particularly attractive, and on linear algebra kernels, on which the efficiency of IPs strongly depends. Interior Point methods and related linear algebra issues were my main research topic since my Ph.D. The focus was on convex Quadratic Programming (QP), which arises as a mathematical model of several real-world applications and as a fundamental problem in the class of NLP.

The basic idea of Interior Point methods is to compute a sequence of approximations of an optimal solution belonging to the interior of the feasible set. They are in fact widely interpreted as algorithms which follow a path of centers (the *central path*) on their way towards an optimal solution. Actually, most of the current algorithms generate iterates which stay in the interior of the positive orthant, but are infeasible for linear constraints (*infeasible* IP algorithms), with the significant advantage that they do not require a usually difficult-to-compute initial feasible point.
Primal-dual IP methods, which use explicitly both primal and dual variables, are the most successful and powerful class of IP methods.

We recall here the main characteristics of these methods, focusing on their application to quadratic programming problems.
Let us consider a convex QP problem in the following form:

$$\begin{array}{ll} \text{minimize} & q(x) = \frac{1}{2}x^T Q x + c^T x \\ \text{subject to} & Ax \geq b, \quad x \geq 0 \end{array} \tag{2.1}$$

where $Q \in \Re^{n \times n}$ is symmetric positive semidefinite, $A \in \Re^{m \times n}$, with $m \leq n$, $c, x \in \Re^n$, $b \in \Re^m$ and the inequalities are interpreted component-wise. We assume that A has full rank m (if this is not the case, the problem has redundant constraints that can be removed). From (2.1) we can easily obtain the pair of primal and dual problems:

$$\mathcal{P} \equiv \left\{ \begin{array}{ll} \text{minimize} & p(x) = \frac{1}{2}x^T Q x + c^T x \\ \text{subject to} & Ax - b = z, \quad x \geq 0, \quad z \geq 0 \end{array} \right. \tag{2.2}$$

and

$$\mathcal{D} \equiv \left\{ \begin{array}{ll} \text{maximize} & d(x,y) = b^T y - \frac{1}{2}x^T Q x \\ \text{subject to} & Qx + c - A^T y = s, \quad s \geq 0, \quad y \geq 0 \end{array} \right. \tag{2.3}$$

where $z \in \Re^m$, $s \in \Re^n$, $y \in \Re^m$, and z and s are primal and dual slack vectors, respectively. Since there is no gap in the optimal solution between the primal and the dual objective values, a way to check optimality is to compute the difference between the two objective function values (*duality gap* Δ).

The optimality conditions (Karush-Kuhn-Tucker (KKT) conditions) are given by the following nonlinear system (matricial form):

$$\begin{bmatrix} Qx + c - A^T y - s \\ Ax - z - b \\ XSe \\ YZe \end{bmatrix} = 0, \quad w \geq 0 \tag{2.4}$$

where $w = (x, y, s, z) \in \Re^{(2\times n+2\times m)}$, X, Y, S, Z are the diagonal matrices of the vectors x, y, s, z, respectively and e is a vector of all ones of appropriate dimension.
The last two equations are the *complementarity conditions*, since they imply that the nonzero elements of the vectors x and s, y and z, must be in complementary locations. Note that, since the objective function is convex and the feasible set is convex, the KKT conditions (2.4) are not only necessary, but also sufficient.

The way the complementarity conditions are dealt with makes the difference between IPs and the other well-known class of methods for NLP, the *Active Set* methods (among which we remember the simplex method for linear programming), see e.g. [55]. Given a complementarity condition $XSe = 0$ (respectively $YZe = 0$), also written as $x_i s_i = 0 \, \forall i \in \{1, \ldots, n\}$, Active Set methods at each step make a prediction of the set of indices for which x_i is not equal to zero and force the corresponding s_i to be zero, then solving a quadratic subproblem reduced to the variables whose indices belong to the selected *working set*, thereby moving along the boundary of the feasible set. Interior Points methods perturb the complementarity conditions by replacing $x_i s_i = 0$ with $x_i s_i = \mu$ and driving the perturbation parameter μ to zero, forcing the convergence to optimality. Basically, IP methods minimize a *barrier function*, adding a logarithmic barrier term in the objective to replace inequality constraints, and they compute at each step a new approximation of the solution moving along a direction, which is computed by solving the perturbed system of optimality conditions (KKT) by applying the Newton method.

A general (feasible) primal-dual IP framework for solving (2.1) is described in Figure 2.1, where $w = (x, y, s, z)$ is an approximation of the solution, $\delta w = (\delta x, \delta y, \delta s, \delta z)$ is a search direction and θ is the step length on such a direction.

In this report, we specially refer to Potential Reduction algorithms, which represent the IP framework considered in [27] and related contributions. IP *Potential Reduction algorithms* are based on the idea of minimizing a logarithmic barrier function called *potential function*. The aim to decrease the potential function as much as possible at each iteration of the algorithm can always be achieved under suitable assumptions, and a corresponding bound on the number of iterations necessary to obtain a desired accuracy in the solution can be computed. For a survey on Potential Reduction methods for

! initialization
choose w^0 strictly feasible, $\mu > 0$
$k = 0$

! iterations
while (convergence criterion not satisfied) **do**
compute a search direction δw^k by solving the system:

$$\begin{bmatrix} Q & -A^T & -I & 0 \\ A & 0 & 0 & -I \\ S^k & 0 & X^k & 0 \\ 0 & Z^k & 0 & Y^k \end{bmatrix} \begin{bmatrix} \delta x^k \\ \delta y^k \\ \delta s^k \\ \delta z^k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -X^k S^k e_n + \mu e_n \\ -Y^k Z^k e_m + \mu e_m \end{bmatrix},$$

compute θ^k such that $w^k > 0$
update the approximation of the solution as $w^{k+1} = w^k + \theta^k \delta w^k$
update μ
$k = k + 1$
endwhile

FIGURE 2.1: A general feasible primal-dual IP framework.

Linear Programming and its extensions the reader is referred to Todd [157].

Complexity result in the case of inexact solution of KKT systems

The focus is on the Potential Reduction (PR) algorithm where the perturbed KKT system, which represents the main computational kernel (see Sect. 2.2), is solved at each PR iteration by an iterative method. When inexact directions are computed, the theory of the IP method has to be reanalyzed, to find suitable conditions that such directions must satisfy in order to guarantee convergence. In [Cafieri et al., 2007d], we proved that the convergence is guaranteed if the residual of the KKT system at each PR iteration satisfies a suitable condition which relates the size of the residual to the duality gap, i.e. to a measure of the progress of the PR method towards the optimal solution.
We consider the infeasible constrained PR method by Mizuno et al. [108] based on the Tanabe-Todd-Ye symmetric primal-dual potential function [152, 158]:

$$\Phi(w) = \rho \log\left(x^T s + z^T y\right) - \sum_{i=1}^{n} \log\left(x_i s_i\right) - \sum_{i=1}^{m} \log\left(z_i y_i\right),$$

where $w = (x, y, s, z) > 0$ and $\rho > n + m$. The following relation holds between the potential function and the duality gap $\Delta = x^T s + z^T y$:

$$\Delta \leq \exp\left(\left(\Phi(w) - (n+m)\log(n+m)\right) / \left(\rho - (n+m)\right)\right); \tag{2.5}$$

therefore, in order to decrease Δ to 0, the method generates a sequence $\{w^k\}$ that drives Φ toward $-\infty$. The main result, which extends the convergence theory of [108] to the inexact computation of the directions, is the following [Cafieri et al., 2007d]. It shows

that convergence is achieved if the norm of the residual of the KKT system at each PR iteration is bounded by a fraction of the ratio between the duality gap and the parameter ρ in the potential function.

Theorem 2.1. *Let $\rho \geq n + m + \sqrt{n+m}$. Suppose that the direction δw^k satisfies the KKT system with a residual $\tilde{r}^k$ such that*

$$\|\tilde{r}^k\| < \frac{\sqrt{3}}{4}\frac{\Delta^k}{\rho}. \tag{2.6}$$

Then, in the feasible case, a step length $\theta^k > 0$ exists such that

$$\Phi(w^k + \theta^k \delta w^k) - \Phi(w^k) < -\delta, \quad \delta > 0. \tag{2.7}$$

In the infeasible case, let $\alpha > 0$ such that an optimal solution $w^* = (x^*, y^*, s^*, z^*)$ exists with $\|w^*\|_\infty \leq \alpha$. If $w^0 = \gamma\alpha(e, e, e, e)$, with $\gamma \in (0, 1]$, then a step length $\bar{\theta}^k \in (0, 1]$ exists, depending on γ, such that

$$\Phi(w^k + \bar{\theta}^k \delta w^k) - \Phi(w^k) < -\bar{\delta}, \quad \bar{\delta} > 0, \tag{2.8}$$

$$\Delta^{k+1} \geq (1 - \bar{\theta}^k)\Delta^k. \tag{2.9}$$

Condition (2.7) and conditions (2.8) and (2.9) not only ensure the convergence of the feasible and the infeasible PR method, respectively, but also lead to standard polynomial complexity results.

2.2 Linear algebra issues in IP methods

Primal-dual IP algorithms compute at each iteration a search direction by applying a Newton step to the nonlinear system of the KKT conditions for the primal-dual problem perturbed with a suitable parameter μ. The linear system deriving by the application of a Newton step has the following form (see Fig. 2.1) and is also referred to as the *KKT system*:

$$\begin{bmatrix} Q & -A^T & -I & 0 \\ A & 0 & 0 & -I \\ S & 0 & X & 0 \\ 0 & Z & 0 & Y \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \\ \delta s \\ \delta z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -XSe_n + \mu e_n \\ -YZe_m + \mu e_m \end{bmatrix}. \tag{2.10}$$

This is a system of $2n+2m$ equations in $2n+2m$ unknowns, which is nonsingular under the assumption that the matrix A has full row rank. IP algorithms differ in the choice of the parameter μ when solving system (2.10).

The solution of the system (2.10) is one of the most critical issues in IP methods. The efficiency and effectiveness of IP algorithms is therefore strongly related to Linear

Algebra algorithms that they use. This symbiotic relationship [123] has also motivated recent advances in numerical Linear Algebra and motivated my Ph.D. thesis [27].
A solution could be obtained by factorizing the whole large system and by solving the factorized system, but usually reduced forms to smaller systems are preferred. There are two main stages of reduction and different solution strategies can be developed for the reduced systems:

- the *augmented system* (by eliminating δs and δz):

$$\begin{pmatrix} Q+E & -A^T \\ -A & -F \end{pmatrix} \begin{pmatrix} \delta x \\ \delta y \end{pmatrix} = \begin{pmatrix} -Se + X^{-1}\mu e \\ Ze - Y^{-1}\mu e \end{pmatrix}, \tag{2.11}$$

  where $E = X^{-1}S$ and $F = Y^{-1}Z$ (with diagonal positive elements, since IP methods generate iterates $w = (x, y, s, z) > 0$).

- the *normal equations* - dual ordering form (by furtherly eliminating δy):

$$(Q + E + A^T F^{-1} A)\delta x = -Se + X^{-1}\mu e - A^T Y e + A^T Z^{-1}\mu e. \tag{2.12}$$

The normal equations have a smaller size than the augmented system and their matrix is symmetric positive definite (spd), while the augmented system matrix is symmetric indefinite, with n positive and m negative eigenvalues (more precisely, it is quasi-definite [162]). On the other hand, the matrix of the augmented system is sparse if Q and A are sparse, while that of the normal equations can result dense even if Q and A are sparse, e.g. when A has a dense column, and it is usually denser anyway. Finally, at each IP iteration, the matrix of the normal equations must be entirely recomputed, while only the diagonal coefficients of the augmented system must be updated. In both augmented system and normal equations, the sparsity pattern of the matrix is unchanged during the whole IP algorithm.

A natural question arises: is it more convenient to solve (2.11) and (2.12) by a direct or by an iterative method?

Direct methods are widely used to solve the systems (2.11) and (2.12) in well-established IP software (e.g., `LOQO` [163], `MOSEK` [12], `OOQP` [52]). The Cholesky factorization is usually applied to the normal equations, and variants of the LDL^T factorization, differing essentially for the pivot selection rule, are applied to the augmented system. Sparse matrix reordering strategies and further "ad hoc" strategies are exploited to deal with the fill-in problem (for a discussion on the direct approach see, e.g., [10, 169]). Note that with direct methods the increasing ill-conditioning of the system matrix is not a severe problem. Indeed, under quite general assumptions, the computed Newton directions are accurate enough to ensure progress toward the optimal solution [45, 168, 170]. When dealing with large-scale problems, the cost of using direct solvers may become

prohibitive in terms of both memory and time requirements. In this case, iterative solvers offer a viable alternative. The contributions presented in the next sections are about the use of iterative methods for solving the KKT system.

Contributions related to the iterative solution of KKT systems

When an iterative method is used to solve the KKT system at each iteration of an IP method, the following issues arise:

- The use of effective preconditioners is mandatory to obtain useful Newton directions, because of the unavoidable ill-conditioning of the linear systems.
- Suitable strategies can be devised to use adaptive accuracy requirements in the solution of the system, in order to avoid unnecessary iterations when the current IP iteration is far from the solution of the problem.
- The convergence theory of the IP method has to be reassessed to take into account the inexact computation of the search directions (see Sect. 2.1).

These items were addressed by contributions [Cafieri et al., 2006],[Cafieri et al., 2007a], [Cafieri et al., 2007c] and [Cafieri et al., 2007d].

Preconditioning

As the solution at each IP iteration approaches the boundary of the feasible set of the primal-dual problem $(\mathcal{P}, \mathcal{D})$, some entries of E and F can become very large (because the primal and the dual variables approach the boundary of the positive orthant), leading to an increasing ill-conditioning. Ill-conditioning deteriorates the rate of convergence of most iterative methods, like Krylov subspace methods, that have to be used with suitable preconditioners: this is one of the most critical difference with direct methods.

A successful class of preconditioners for the augmented system is that of *Constraint Preconditioners (CPs)*, which in recent years have attracted the interest of many IP researchers and, more generally, of people working on saddle-point problems (see e.g. [19, 33, 40, 54, 73, 80, 94, 127, 133, 44]; for an overview of CPs see [16]). They are symmetric indefinite preconditioners having the same block structure as the augmented system matrix, with the upper-right and lower-left blocks unchanged.
In [Cafieri et al., 2007a] we presented a suitable Constraint Preconditioner for the augmented system matrix arising from convex QP problems with inequality constraints (2.1):

$$P = \begin{pmatrix} diag(Q) + E & -A^T \\ -A & -F \end{pmatrix}$$

(applied through a sparse direct factorization) and we analyzed the behavior of this preconditioner with the Conjugate Gradient (CG) algorithm. In particular, we showed

that with a suitable choice of the starting point the Conjugate Gradient algorithm applied to the preconditioned (indefinite) augmented system behaves as if it were applied to suitably preconditioned spd normal equations. More precisely, since the matrices of the augmented system and P are not spd, a breakdown could occur in the CG algorithm; furthermore, the CG convergence properties are not guaranteed. However, we showed that this is not the case when the starting guess is chosen in such a way that the initial residual has the last m components equal to 0. Indeed, by using such a starting guess, the Conjugate Gradient with Constraint Preconditioner (CPCG) behaves as if it were applied to a spd system, as stated by the following theorem.

Theorem 2.2. *In the CPCG algorithm, if*

$$\delta x^0 = 0, \quad \delta y^0 = -F^{-1}b_2, \tag{2.13}$$

then the vectors $\delta x^i, r_1^i, z_1^i$ and p_1^i, i.e., the approximate solution, the residual, the preconditioned residual and the search direction, are equal respectively to the corresponding vectors at the i-th iteration of the CG algorithm applied to the normal equations $(C + A^T F^{-1} A)\delta x = b_1 - A^T F^{-1} b_2$, (with $C = Q + E$, $b_1 = -Se + X^{-1}\mu e$ and $b_2 = Ze - Y^{-1}\mu e$) with preconditioner $\tilde{P} = diag(Q) + A^T F^{-1} A$ and null starting guess.

Starting from this equivalence, we proved the convergence results for the Conjugate Gradient method with constraint preconditioner P:

Corollary 2.3. *If the starting guess (2.13) is chosen, then*

i) CPCG does not break down;

ii) it converges in at most n iterations;

iii) the norm of the error on the computed solution at each step is bounded by a quantity depending on the initial error and on the spectral properties of the sdp preconditioned matrix.

Adaptive stopping criteria of inner iterations

When using iterative methods for the (inner) KKT systems in IP methods, one can relate at each step the accuracy of the solution of the system to the quality of the current IP iterate, to reduce the computational cost. The idea is to use adaptive inner iterations stopping criteria that require low accuracy when the outer IP iterate is far from the optimal solution and require higher accuracy as soon as the IP iterate approaches the solution. This is deeply analyzed in [Cafieri et al., 2007c] in the framework of a Potential Reduction algorithm (feasible and infeasible versions), although we believe that other IP algorithms can benefit from the proposed criteria. The basic idea is to relate the accuracy in the solution of the KKT system to the convergence properties of the

Potential Reduction (PR) algorithm (see Sect. 2.1). In particular, a stopping criterion directly deriving from the Potential Reduction convergence results relates the accuracy in the solution of the KKT system to the current duality gap value Δ. Basically, one requires that the residual norm is less than some factor of the initial residual norm: $\frac{||r^i||}{||r^0||} \leq tol^k_{CG}$ and tol^k_{CG} is chosen depending on Δ (the value of Δ decreases when the PR iterates approach the solution, thereby requiring a higher accuracy in computing the Newton direction). Other stopping criteria are derived from modifications of this criterion, aimed at reducing the number of inner iterations, also combined with computational strategies devised to deal with a possible slowdown of convergence. Extensive numerical experiments helped identifying the criterion that achieves the best trade-off between the computational cost and the ability to reduce the infeasibility at each iteration of the infeasible Potential Reduction algorithm.

Approximating the preconditioner

A Constraint Preconditioner (CP) is often applied through its sparse direct factorization. Although this factorization is generally less expensive than the factorization of the (unpreconditioned) system matrix, in large-scale problems it may still account for most of the execution time of a single IP iteration. For this reason, a few strategies have been proposed in the literature to approximate the preconditioner, to reduce the computational cost of its application (CP using a sparse approximation of A instead of the original matrix [18]; CP based on an incomplete factorization of the Schur complement [17]; CP approximated using incomplete Schilders' factorizations [34]).

In [Cafieri et al., 2007b], we proposed an alternative approach which consists in using for multiple IP iterations the CP that has been computed at a certain iteration. We then devised different strategies for selecting the PR iterations in which the preconditioner has to be recomputed. The number of iterations for which the preconditioner is kept fixed should be chosen taking into account that the use of an approximate preconditioner is expected to increase the number of inner iterations. Therefore, an effective strategy should be such that the time saved in the factorization of the CP pays off the time increase due to the larger iteration count. In particular, an adaptive approach, in which the same CP factorization is reused until its effectiveness deteriorates in terms of number of iterations required to solve the system, appeared very promising. This dynamic choice of the iteration when the CP has to be recomputed achieves the best trade-off between the reduction of the time for the factorization of the preconditioner and the increase of the time for solving the augmented system.

2.3 ATM application: optimal control for aircraft conflict avoidance

Conflict avoidance problems arising in Air Traffic Management (ATM), already introduced in Chapter 1, Sect. 1.4, represent the main application domain of the Ph.D. thesis of Loïc Cellier, who I am currently supervising. The focus is again on modeling and deterministic solution methods, but this time an optimal control approach is developed. Optimal Control gives rise to NLP problems when a certain type of solution methods, called *direct methods*, is applied.

Aircraft conflict avoidance (see Chap. 1, Sect. 1.4) can be naturally interpreted as a dynamic system on which one can act by means of a command (*control*) to move the system from an initial state to a final state while satisfying an optimality criterion, hence it can be formulated as an optimal control problem. To achieve aircraft separation by speed adjustments, the control can be naturally chosen as the aircraft acceleration and the optimality criterion as the minimization of a cost function depending on aircraft speed modifications. In [Cellier et al., 2012], with L. Cellier and F. Messine, we proposed the following model $(\mathcal{P})$ for an n-aircraft problem in a planar configuration (common flight level), where x_i, v_i and u_i are respectively the position, the velocity and the acceleration (control) of aircraft i, and $I = \{1, ...,\text{n}\}$. The trajectories are kept unchanged, while optimal control models for aircraft conflict avoidance in the literature usually put the trajectory as a command on the system, see e.g. [21, 159].

$$
(\mathcal{P}) \begin{cases}
\min\limits_{u} \quad \sum\limits_{i=1}^{n} \int_{t_0}^{t_f} u_i^2(t)dt & \\
\dot{v}_i(t) = u_i(t) & \forall t \in [t_0, t_f],\ \forall i \in I \\
\dot{x}_i(t) = v_i(t) d_i & \forall t \in [t_0, t_f],\ \forall i \in I \\
\underline{u_i} \leq u_i(t) \leq \overline{u_i} & \forall t \in [t_0, t_f],\ \forall i \in I \\
\underline{v_i} \leq v_i(t) \leq \overline{v_i} & \forall t \in [t_0, t_f],\ \forall i \in I \\
x_i(t_0) = x_i^0 \qquad v_i(t_0) = v_i^0 & \forall i \in I \\
x_i(t_f) = x_i^f \qquad v_i(t_f) = v_i^f & \forall i \in I \\
D^2 - \parallel x_i(t) - x_j(t) \parallel^2 \leq 0 & \forall t \in [t_0, t_f],\ \forall i < j
\end{cases}
$$

In $(\mathcal{P})$, t, t_0 and t_f are the time, the initial time and final time respectively, D is the minimum required horizontal separation distance between two aircraft and d_i is the direction (heading) of the i^{th} aircraft. The final time t_f of maneuvers is fixed and identical for all aircraft (time horizon). For each aircraft i, velocity v_i and acceleration u_i are bounded, in particular v_i may be required to satisfy the bounds imposed by the ERASMUS project (see Sect. 1.4).

A *direct method* for an optimal control problem like $(\mathcal{P})$ is a typical solution approach based on a time discretization. It leads to a nonlinear continuous optimization problem, which can be solved by applying numerical methods for NLP, such as Interior Point methods described in Sect. 2.1 for quadratic programming. The complexity of the NLP corresponding to the direct method is $O(np)$ for the number of variables and $O(n^2p)$ for the number of constraints, where n and p are the number of aircraft and the number of time subdivisions respectively. For example, even on a simple conflict problem with only 2 aircraft and 15 seconds of time discretization step in a time window of 30 minutes, the corresponding nonlinear problem has 240 variables and more than 1000 constraints. As a consequence, the NLP problem arising from the application of a direct method may be very difficult to solve. In [Cellier et al., 2012], we presented strategies tailored on the problem at hand, aimed to handle medium and large-scale problems.
First, we propose to distinguish two discretization steps, for the *detection* of potential conflicts and for the *resolution* of conflicts respectively (detection and resolution can be performed at the same time by applying a direct method). We propose to use a step for detection tight enough to check if all constraints are respected, and a step for resolution, used to decide the time frequency at which the values of the controls are computed, larger than the previous one. For example, we used 15 seconds for the detection and 1 or 5 minutes for the resolution. Using two different time discretization steps, the number of variables and constraints of the nonlinear optimization problem to be solved are reduced.

Second, we propose a strategy of decomposition of the conflict avoidance problem in such a way to obtain subproblems which can be solved by different optimal control approaches, namely a direct and an indirect method, drastically reducing the computational complexity of the whole problem. The idea is to consider different regions of the considered air-space, depending on the aircraft separation constraints. We define *zone* the region where, for an aircraft pair, separation constraints have to be verified because potential conflicts occur, and *postzone* the following region where all the conflicts have been solved and therefore the aircraft are already separated. Analogously, *prezone* is the region crossed before entering the zone. In this case, a pre-processing step has to be performed to detect potential conflicts: given aircraft predicted trajectories, one can check intersections of the trajectories and localize spatial regions where the separation constraints must be checked. More precisely, for each aircraft i, one can identify a point x_i^{enter} of its trajectory which corresponds to entering the zone of potential conflict with another aircraft, and consequently identify the time t_i^1 to reach x_i^{enter} using the highest speed $\overline{v_i}$. Dually, one can identify the point to leave the zone and the corresponding time t_i^2 to reach the exit point x_i^{exit} using the lower speed $\underline{v_i}$. For n aircraft, setting the entry zone time equals to $t_1 := \min_{i\in\{1,\dots,n\}} t_i^1$ and the exit zone time $t_2 := \max_{i\in\{1,\dots,n\}} t_i^2$, we define conflict time phases for the whole problem. The *prezone*, the *zone* and the

postzone correspond respectively to the time periods $[t_0, t_1]$, $[t_1, t_2]$ and $[t_2, t_f]$, t_f being the time horizon. Figures 2.2 and 2.3 illustrate the proposed decomposition, from the spatial and temporal point of view.

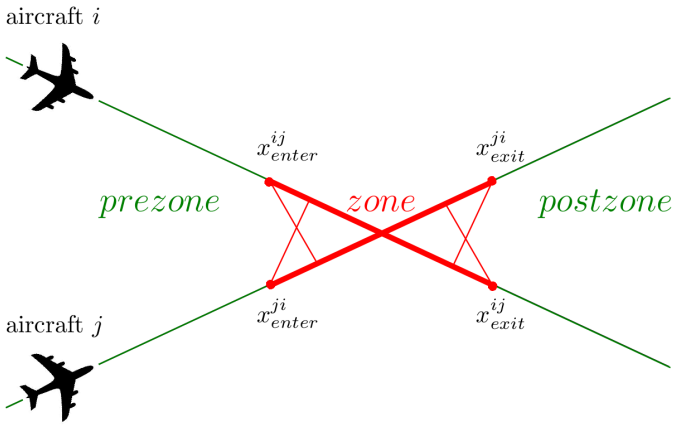


FIGURE 2.2: Decomposition of the aircraft deconfliction problem: spatial interpretation

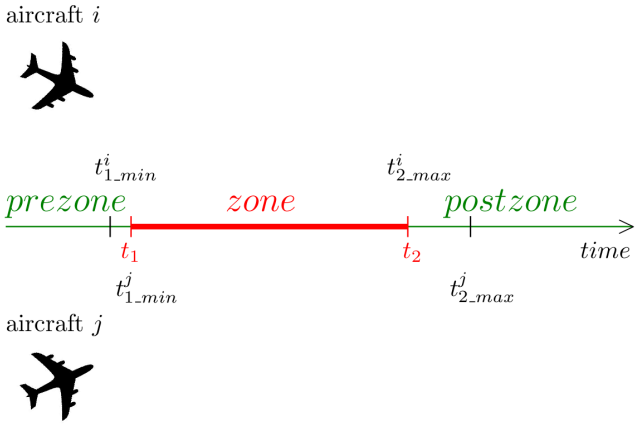


FIGURE 2.3: Decomposition of the aircraft deconfliction problem: temporal interpretation

As a first step of investigation, we focused on the decomposition of the problem into *zone* and *postzone* (the prezone will be the subject of further research).
With this decomposition, the direct method is applied only before the postzone, instead of on the whole time horizon. Numerical integrators of Euler-type are used to approximate the ordinary differential equations describing the system dynamics and different time discretization steps mentioned above are exploited. The *postzone* (time window $[t_2, t_f]$) is characterized by the absence of separation constraints, which are difficult state constraints (involving state variables). Thus, it represents a subproblem easier to solve than the initial problem, on which we can apply an *indirect method*. An indirect

method is based on the known Pontryagin's maximum principle (PMP), which gives a necessary condition for optimality and allows to obtain an analytical solution. For each aircraft i, the following optimal control subproblem $(\mathcal{P}_i)$ can be solved independently (the velocity and acceleration constraints are checked *a posteriori*).

$$(\mathcal{P}_i)\begin{cases} \min\limits_{u_i} \quad \displaystyle\int_{t_2}^{t_f} u_i^2(t)dt & & \\ \quad \dot{v}_i(t) = u_i(t) & \forall t \in [t_2, t_f] & \\ \quad \dot{x}_i^X(t) = v_i(t)d_i^X & \forall t \in [t_2, t_f] & \\ \quad \dot{x}_i^Y(t) = v_i(t)d_i^Y & \forall t \in [t_2, t_f] & \\ \quad x_i^X(t_2) = x_i^{X_{t_2}} & x_i^Y(t_2) = x_i^{Y_{t_2}} & v_i(t_2) = v_i^{t_2} \\ \quad x_i^X(t_f) \text{ free} & x_i^Y(t_f) \text{ free} & v_i(t_f) = v_i^{t_f} \end{cases}$$

Following the classical indirect approach, writing the Hamiltonian H, deriving H to obtain the co-state equations and applying the PMP, we finally obtain the following solution system:

$$\begin{cases} u_i(t) = \dfrac{v_i^{t_f} - v_i^{t_2}}{t_f - t_2} \ , \\ v_i(t) = \dfrac{v_i^{t_f} - v_i^{t_2}}{t_f - t_2}(t - t_f) + v_i^{t_f} \ , \\ x_i^X(t) = \dfrac{v_i^{t_f} - v_i^{t_2}}{t_f - t_2} d_i^X \dfrac{t^2}{2} + (v_i^{t_f} - \dfrac{v_i^{t_f} - v_i^{t_2}}{t_f - t_2} t_f) d_i^X t \\ \qquad - (\dfrac{v_i^{t_f} - v_i^{t_2}}{t_f - t_2}(t_2 - t_f) + v_i^{t_f}) d_i^X t_2 + x_i^{X_{t_2}} \ , \\ x_i^Y(t) = \dfrac{v_i^{t_f} - v_i^{t_2}}{t_f - t_2} d_i^Y \dfrac{t^2}{2} + (v_i^{t_f} - \dfrac{v_i^{t_f} - v_i^{t_2}}{t_f - t_2} t_f) d_i^Y t \\ \qquad - (\dfrac{v_i^{t_f} - v_i^{t_2}}{t_f - t_2}(t_2 - t_f) + v_i^{t_f}) d_i^Y t_2 + x_i^{Y_{t_2}} \ . \end{cases}$$

Hence, starting from t_2, the problem can be solved analytically. Numerical results show that the proposed strategy of decomposition and hybridization of optimal control methods significantly reduces the computational time of resolution of the conflict avoidance problem, thus allowing to raise the size of the problems that can be solved using optimal control.

Chapter 3

Network Clustering

Network clustering problems can be formulated using mathematical programming and usually lead to combinatorial optimization problems. After a brief introduction to clustering on networks, this chapter presents my main contributions in this area, issue of a fruitful collaboration with Pierre Hansen and co-workers from GERAD in Montréal (Sylvain Perron, Gilles Caporossi and Daniel Aloise) and from École Polytechnique in Paris (Leo Liberti and Alberto Costa). These contributions concern clustering criteria and corresponding clustering methods and are mainly around exact methods, used either to solve the whole optimization problem or, locally, subproblems arising in hierarchical heuristics, or to refine solutions previously obtained by other methods.

3.1 Clustering on networks

Networks have been identified as an extremely useful representation of complex systems in a wide variety of domains. The most prominent examples include social networks, describing individuals and their interactions and relationships, telecommunication networks, such as the World Wide Web, transportation networks, biological networks, and many more. A detailed introduction to networks has recently been given by Newman [117]. Complex networks, mathematically represented by *graphs*, have thus been extensively studied in the last decade, specially by the physicists and computer scientists communities. This also led to several important discoveries, such as the power law distribution of degrees [14] and the small world property [167].

Some topological features of networks are studied to better understand the underlying complex systems. Such systems usually consist of many interacting components. Thus, their structure can be understood by identifying the way the nodes of the corresponding networks are connected to each other. A modular structure characterizes many complex systems, meaning that they contain subgroups of entities sharing some common properties. A topic of particular interest in the study of complex networks is therefore the identification of modules, also called *clusters* or *communities*, the last term

being widely used in the physicists' literature (originally referred to social networks). Speaking informally, a community is a subset of nodes which are more densely linked compared to the rest of the network. For example, a community in a social network can be constituted by individuals sharing a common interest or location, in a biological network by entities with a common function, in the World Wide Web by web-pages having a common topic or language, etc. Community detection is therefore very useful to identify some properties of the system described by the studied network starting from its structural features. It also allows us to study modules individually based on their properties, or to visualize and analyze to a higher level very large and complex networks by compressing its modules in single nodes [116]. The reader is referred to Fortunato [46] for a recent extensive and thorough survey of that very active research domain.

A network, or graph, $G = (V, E)$ is composed of a set V of n vertices and a set E of m edges which join pairs of vertices. Vertices are associated with the entities of the system under study and edges express that a relation defined on all pairs of vertices holds or not for each such pair. A subgraph $G_S = (S, E_S)$ of a graph G induced by a set of vertices $S \subseteq V$ is a graph with vertex set S and edge set E_S equal to all edges with both vertices in S. Such a subgraph corresponds to a cluster (or module or community) and clustering on networks aims at finding a partition of V into pairwise disjoint nonempty subsets $V_1, V_2, \ldots, V_N$ inducing subgraphs of G. Roughly speaking, one seeks clusters which contain more inner edges (with both vertices in the same cluster) than cut edges (with vertices in different clusters). It is worth noticing that the terms *community identification*, *graph clustering* and *graph partitioning* are often used interchangeably in this context.

The problem can be formulated using mathematical programming. Several formulations can be considered, depending on the definition of community as well as the criterion chosen to evaluate the quality of the partition. The research in this context is in fact generally essentially addressed, on the one hand, to proposing and evaluating clustering criteria, and on the other hand to devising efficient solution methods for the corresponding optimization problems. My research activity in this area follows both these directions, with a special focus on exact methods. The following sections present my contributions, first in the framework of the well known criterion of *modularity* (Sect. 3.2), then focusing on other criteria (Sect. 3.3).
Unweighted undirected graphs are considered.

3.2 Modularity maximization

Basics

A precise definition of the quality of a partition into communities has been given in a seminal paper by Newman and Girvan [115]. They proposed to compare the fraction

of edges falling within communities to the expected fraction of such edges, giving the following definition, for a partition in communities, of *modularity* function:

$$Q = \sum_s (a_s - e_s)\,, \tag{3.1}$$

where a_s is the fraction of edges in community s and e_s is the expected value of the same quantity in a graph in which the vertices have the same degrees but edges are placed at random. A maximum value of Q near to 0 indicates that the network considered is close to a random one (barring fluctuations), while a maximum value of Q near to 1 indicates strong community structure. Thus, modularity on the one hand can be viewed as a measure of the extent to which the classes of a partition can be considered to be communities (it expresses not only that a community contains a large fraction of the edges, but also that it contains a larger fraction of the edges than would be expected), on the other hand can be maximized to find an optimal partition of a network. Modularity is by far the most popular criterion for community detection, and has spawned in recent years numerous methods to identify communities (see Sect. 3.2.1). It exhibits, in fact, some clear advantages: the modularity function has a clear and simple mathematical description and does not depend on parameters being decided arbitrarily; furthermore, modularity maximization gives an optimal partition together with the number of clusters that is automatically detected, and thus has not to be specified in advance. Finally, one can use mathematical programming to model the community detection problem.

Modularity without loops and multiple edges

Most of the contributions described in this chapter are around the concept of modularity. First, in [Cafieri et al., 2010b] we analyzed a behavior of the modularity function and proposed suitable modifications. Indeed, one has to note that, despite its popularity, some behaviors of modularity are still under investigation and some criticism has been raised in recent literature, see, e.g., [56, 47, 96, 78], the most important being the existence of a resolution limit [47] (in the presence of large clusters, some clusters smaller than a certain size which depends on the number of edges of the network can be undetectable) and the degeneracy of the modularity function [56] (there can be in some cases a large number of partitions, different from each other, all having high modularity values). To address these criticisms a few approaches have been proposed in the literature.

The work in [Cafieri et al., 2010b] was motivated by the observation that, using the standard definition of modularity, graphs are compared to a *null model* containing loops and possibly multiple edges, while usually the graph under study has neither (simple graph). We then derived sharp bounds on the expected number of loops, and their impact on the modularity of simple graphs. The contribution to Q of the loops is equal

in absolute value to $C = \frac{\sum_{u \in V} k_u^2}{4m^2}$, where k_u is the degree of vertex u, and we found

$$C \geq \frac{1}{n} \quad \text{and} \quad C \leq \frac{n}{4n-4}.$$

When the order n of the graph increases, the bounds tend to different limits:

$$\lim_{n \to \infty} \frac{1}{n} = 0 \quad \text{and} \quad \lim_{n \to \infty} \frac{n}{4n-4} = \frac{1}{4}.$$

So, due to loops, at least a small constant will be subtracted from the community dependent part of Q when n increases, but quite a large one must be subtracted in the worst case, even for large n. Furthermore, the lower bound is attained for regular graphs, and tends to be small, while the upper bound is attained for stars, and is large; so, the effect of loops in the null model can be considerable in the worst case.
Then, we proposed modified null models associated with graphs without loops but with multiple edges, graphs with loops but without multiple edges and graphs without loops nor multiple edges. Using conditional probabilities, we provided a modified formula for modularity in the case where loops are excluded from the null model. The original formula (equivalent to (3.1))

$$Q = \frac{1}{2m} \sum_{u,v \in V} (A_{uv} - \frac{k_u k_v}{2m}) \delta(c_u, c_v), \tag{3.2}$$

where A is the adjacency matrix, k_u, k_v the degrees of vertices u, v and $\delta(c_u, c_v)$ is equal to 1 if u and v belong to the same community $c_u = c_v$ and to 0 otherwise, is modified to:

$$Q' = \sum_{u,v \in V : v > u} \left(\frac{A_{uv}}{m} - \frac{k_u k_v}{2m} \left(\frac{1}{2m - k_u} + \frac{1}{2m - k_v} \right) \right) \delta(c_u, c_v), \tag{3.3}$$

where the expected number of edges is computed, using conditional probabilities, to exclude loops.
To address the problem of avoiding multiple edges in the null model, we proposed an algorithm for redistribution of the excess over 1 of the expected number of edges between two vertices to the other edges for which it is not the case. This redistribution is proportional to the edge probabilities. The algorithm can be applied either to the initial null model or to the modified null model in which loops have been eliminated. Experimental results show that the partitions obtained with the standard modularity and with the proposed modifications are often the same, however modularity values in the latter case increase.

3.2.1 Exact methods

Modularity maximization has been proved to be NP-hard [25]. Numerous heuristics have been proposed to maximize modularity, which are based on hierarchical clustering or partitioning algorithms and rely upon several different approaches ([115, 30, 32, 165, 23, 62, 6] are a few examples). In contrast, papers proposing exact algorithms or using mathematical programming are rare for modularity maximization. The main reason is that they can only solve small instances (with about a hundred entities) in reasonable time, while heuristics can solve (approximately) very large instances with up to hundred or thousand entities. However, heuristics do not have either an a priori performance guarantee (finding always a solution with a value which is at least a given percentage of the optimal one), nor an a posteriori performance guarantee (that the obtained solution is at least a computable percentage of the optimal one), opposite to exact algorithms that provide an optimal solution together with the proof of its optimality. We have chosen to advance on the almost-unexplored way of exact algorithms for modularity maximization. There are in fact in our opinion many reasons to focus on exact algorithms, among which: having an exact solution solves the problem of separating possible inadequacies of the model from eventual errors resulting from the use of heuristics, thus communities may be interpreted with more confidence; an exact algorithm may be stopped and the best solution found considered as a heuristic one (it is not uncommon that the optimal solution is found at an early stage of the resolution); an exact algorithm can provide a benchmark of exactly solved instances which can be used to compare heuristics and fine tune them.

In [Aloise et al., 2010], the purpose was to assess and advance the state of the art of algorithms for exact modularity maximization. Only two exact algorithms were known in the literature, one working on a reduction of modularity maximization to clique partitioning and the other working on the direct formulation. We started from these two approaches and proposed two new algorithms. We thus discussed and compared the following four algorithms: (i) the row generation algorithm of [60], which subsumes the algorithm of [25]; (ii) a new column generation algorithm for clique partitioning which enhances the efficiency of that approach; (iii) the mixed integer convex quadratic programming approach of [172]; (iv) another new column generation algorithm which enhances the efficiency of the second approach.

- *Modularity maximization as clique partitioning*

Observe that writing modularity as in (3.2) as a sum of values over all edges of the complete graph K_n, introducing binary variables x_{ij} equal to 1 if vertices i and j belong to the same module and 0 otherwise, and setting $w_{ij} = \frac{1}{m}\left(a_{ij} - \frac{k_i k_j}{2m}\right)$ as the weight on edge (i, j), modularity maximization can be reformulated as a clique partitioning problem. The resulting partition is an equivalence relation, so one can write a model

as in [60, 61] where the objective is $\max \sum_{i<j \in V} w_{ij} x_{ij} - C$ and constraints express reflexivity, symmetry and transitivity. The resulting model is a linear program in 0-1 variables, which has $\frac{n(n-1)}{2}$ variables and $3\binom{n}{3} = O(n^3)$ constraints. It can be solved by a classical row generation approach, however memory size is a limiting factor.

We proposed a column generation algorithm which implicitly takes into account all possible communities (or in other words all subsets of the set of entities under study). The problem of finding simultaneously all communities in an optimal partition is replaced by a sequence of optimization problems for finding one community at a time (more precisely a community which improves the modularity of the current solution). So, the columns correspond to all subsets of V, i.e., to all nonempty communities.

To express this problem, we define $a_{it} = 1$ if vertex i belongs to module t and $a_{it} = 0$ otherwise and then write the model as

$$\max \sum_{t \in T} c_t z_t - C \tag{3.4}$$

$$\text{s.t.} \sum_{t \in T} a_{it} z_t = 1 \qquad \forall i = 1, \ldots, n \tag{3.5}$$

$$z_t \in \{0, 1\} \qquad \forall t \in T, \tag{3.6}$$

where $c_t = \sum_i \sum_{j>i} w_{ij} a_{it} a_{jt}$, i.e., the value of the module indexed by t with $t = 1 \ldots 2^n - 1$. The objective function (3.4) expresses that modularity is equal to the sum of modularities of all selected modules minus a constant corresponding to the diagonal terms. The first set of constraints (3.5) expresses that each entity must belong to one and only one module and the second set of constraints that modules must be selected entirely or not at all. If the integrality constraints (3.6) are replaced by

$$z_t \geq 0, \forall t \in T, \tag{3.7}$$

the upper bound $z_t \leq 1$ being implied by constraint (3.5), one obtains a *relaxation* of (3.4) - (3.6) which is a linear program. Problem (3.4)-(3.5),(3.7) is called the *master problem*. To apply column generation, a *reduced master problem* with considerably fewer columns is solved instead, where the reduced cost associated with column t is equal to $c_t - \sum_i \lambda_i a_{it}$ (λ_i being the current values of the dual variables of the continuous relaxation of (3.4)-(3.5),(3.7)). To add columns progressively, an *auxiliary problem* is solved, which finds a column with positive (negative) reduced cost in case of maximization (minimization). Replacing the coefficients a_{it} by binary variables y_i leads to the auxiliary problem:

$$\max_{y \in \mathbb{B}^n} \sum_i \sum_{j>i} w_{ij} y_i y_j - \sum_i \lambda_i y_i$$

This is a quadratic program in 0-1 variables with a 100% dense matrix of coefficients.

In our experiments, we used a Variable Neighborhood Search (VNS) heuristic [109, 65] as long as it can find a column with positive reduced cost. When VNS fails to find an improving column, we used as exact method a simple branch and bound algorithm [64]. It is well known that column generation algorithms suffer from slow convergence particularly when the optimal solution is *degenerate*, i.e., when such a solution has many variables equal to 0, which is the case for clustering problems. Column generation algorithms also suffer from the plateau effect, i.e., the optimal solution keeps the same value for several or many iterations [161]. To alleviate these defects, we used a variant of the stabilization methods for column generation due to du Merle et al. [35].

- *Modularity maximization using a direct formulation*

Maximizing modularity by the clique partitioning approach has a drawback: it replaces a usually sparse matrix of coefficients by a 100% dense one. An alternative approach is to work directly with the graph $G = (V, E)$ instead of the complete graph K_n. This was done by Xu, Tsoka and Papageorgiou [172] and leads to a 0-1 mixed integer quadratic problem whose continuous relaxation is convex, and which can therefore be solved by `CPLEX`. Xu et al.'s model provides the necessary background for another new column generation algorithm that we proposed. Considering again the definition of Q as a sum over modules of their modularities, Q can be rewritten as

$$Q = \sum_s [a_s - e_s] = \sum_s \left[\frac{m_s}{m} - \left(\frac{D_s}{2m} \right)^2 \right] \tag{3.8}$$

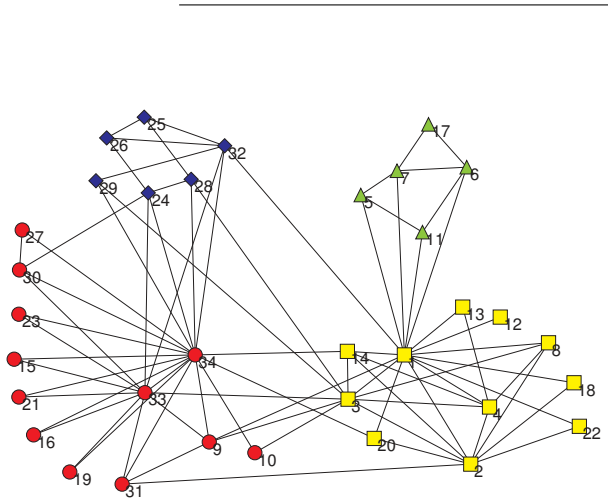
where m_s denotes the number of edges in module s and D_s denotes the sum of degrees k_i of the vertices of module s. Binary variables are then used by Xu et al. to identify the modules to which each vertex and each edge belongs, and a number of constraints are introduced to express that each vertex belongs to exactly one module, that any edge can only belong to module s if both of its end vertices belong to that module, to impose lower and upper bounds on the cardinality of the modules and to break symmetries.
We proposed a column generation algorithm where the master problem is the same as in the previous column generation algorithm, i.e., its equations are given in (3.4)-(3.5),(3.7), and the auxiliary problem is close to the formulation of Xu et al. [172] but much simpler. As a single community is to be determined at a time, it can be written as follows:

$$\begin{aligned}
\max_{x \in \mathbb{B}^n, D \in \mathbb{R}} \quad & \sum_r \frac{x_r}{m} - \left(\frac{D}{2m} \right)^2 - \sum_i \lambda_i y_i \\
\text{s.t.} \quad & D = \sum_i k_i y_i \\
& x_r \leq y_i \qquad \forall r = \{i, j\} \in E \\
& x_r \leq y_j \qquad \forall r = \{i, j\} \in E.
\end{aligned}$$

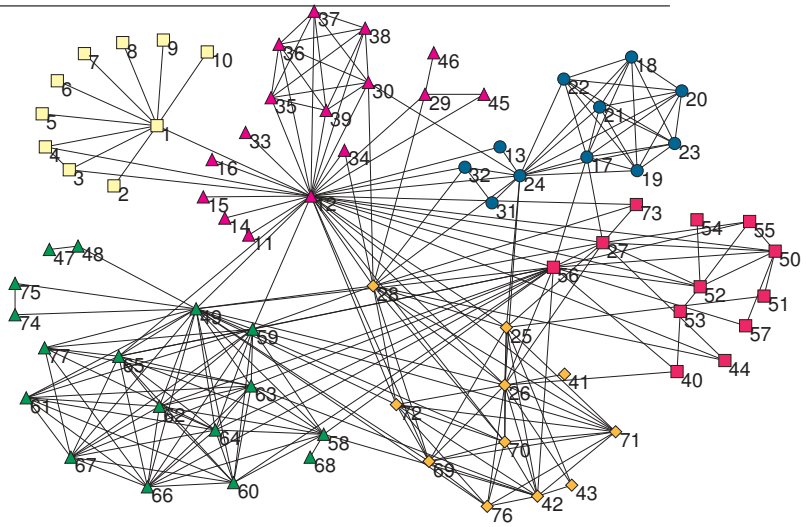
Variable x_r is equal to 1 if edge r belongs to the community which maximizes the objective function and to 0 otherwise. Similarly, y_i is equal to 1 if the i^{th} vertex belongs to the community and 0 otherwise. The objective function is equal to the modularity of the community to be determined minus the scalar product of the current value λ_i of the dual variables times the indicator variable y_i. This is a mixed integer quadratic problem with $n+m$ binary variables and 1 continuous variable, in the objective function, subject to $2m+1$ linear constraints. In the objective function there is a single concave nonlinear term. Clearly, the size of this auxiliary problem is much smaller than that of the direct formulation, particularly for large number of communities. This auxiliary problem is first solved with a VNS heuristic as long as a column with a positive reduced cost can be found. When this is no more the case, `CPLEX` is called to find such a column or prove that there are no more.

A computational comparison on a set of instances from the literature showed that both row-generation and column-generation algorithms based on reformulation of modularity maximization as a clique partitioning are competitive for small instances, but become too time or memory consuming for larger ones due to the rapid increase in the number of variables and constraints; the column generation algorithm reformulated from [172] direct formulation appears to be the best choice since its computing time is comparable for small instances and is the only algorithm able to solve large instances to optimality. Problems are in fact solved much faster than with previous algorithms and larger instances can be tackled, raising the size of exactly solved instances from 105 entities (known in the literature) to 512.

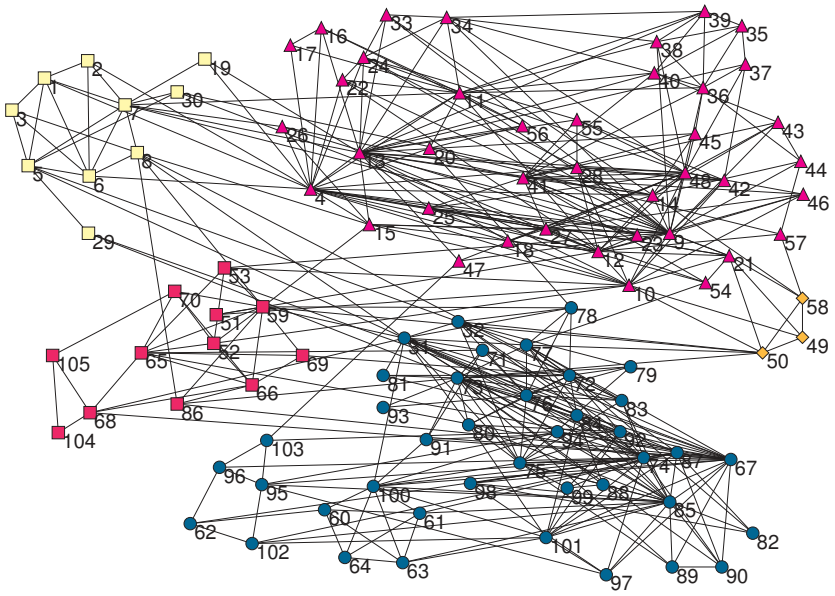
Figure 3.1 shows partitions obtained by exactly maximizing modularity for datasets corresponding to various real world problems, also illustrating the variety of applications studied by network clustering. Zachary's karate club dataset is probably the best known. It describes friendship relations between 34 members of a karate club observed over two years by Zachary [173]. In that period the club split into two groups after a dispute between the club owner and the karate instructor. Hugo's *Les Misérables* network [75] describes the relationships between characters in Victor Hugo's masterpiece [67], with vertices associated to these characters and edges associated with pairs of characters appearing jointly in at least one of the many chapters of the novel. Political book network [76] deals with co-purchasing of political books on Amazon.com and the last network [53] represents the schedule of games between American college football teams in the Fall 2000.



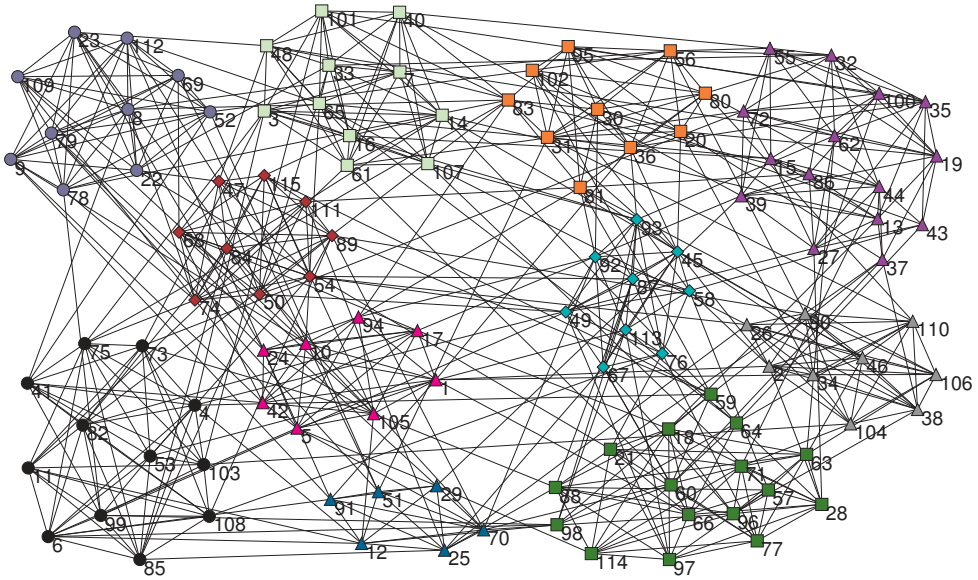
Zachary's karate club network



Hugo's *Les Misérables* network



Political books network



American football network

FIGURE 3.1: Network partitions obtained by modularity maximization on a set of known social networks from the literature.

3.2.2 Hierarchical divisive heuristic based on local exact solutions

New locally optimal divisive heuristic

A heuristic finds a near optimal partition (or sometimes an optimal partition but without proof of its optimality) in moderate time compared to exact algorithms. In [Cafieri et al., 2011] we considered a heuristic solution of the network clustering problem, but using exact solutions locally. Specifically, we devised a locally optimal hierarchical divisive heuristic. Hierarchical heuristics are in principle devised for finding a hierarchy of partitions implicit in the given network when it corresponds to some situation where hierarchy is observed or postulated. Heuristics of this kind are divided into agglomerative and divisive ones. Hierarchical agglomerative heuristics [118, 30, 32, 165, 23] proceed from an initial partition with n communities each containing a single entity and iteratively merge the pair of entities for which this operation increases most the objective function, until all entities belong to the same community. Hierarchical divisive heuristics [114] proceed from an initial partition containing all entities and iteratively divide a community into two in such a way that the increase in the objective function value (e.g. modularity) is the largest possible, or the decrease in the objective value is the smallest possible. Bipartitions are iterated until a partition into n communities having each a single entity is obtained. In practice, for some objectives including modularity mergings or bipartitions can be ended once they do not improve the objective function value anymore. In a divisive hierarchical heuristic, the subproblem of finding a bipartition is difficult, even more considering that modularity maximization is NP-hard even in the case of two clusters [25]. In fact, the only previous divisive heuristic was that proposed by Newman [114], based on spectral graph theory and refinement by the Kernighan-Lin heuristic [74]. We proposed a new divisive heuristic that is locally optimal, in the sense that each of the successive bipartitions (splitting step) is done in a provably optimal way, using an exact algorithm. To solve exactly the maximizing modularity bipartition problem, we proposed a mathematical programming model which uses binary variables to identify to which community each vertex and each edge belongs (in this respect, our model is similar to that of Xu et al. [172], see Sect. 3.2.1). More precisely, variables $X_{i,j,s}$ for each edge (v_i, v_j) and $s = 1, 2$, and variables $Y_{i,1}$ for $i = 1, 2, \dots n$ are defined in such a way that $X_{i,j,s}$ is equal to 1 if the edge (v_i, v_j) is inside the cluster s and $Y_{i,1}$ is equal to 1 if the vertex v_i is inside the cluster 1, and 0 otherwise. We used the fact that, since a bipartition has to be found, only two sub-modules of the original community have to be considered. Thus, we can express the sum of degrees of vertices belonging to the second community as a function of the sum of degrees of vertices belonging to the first one, and consequently rewrite the expression of the modularity function (3.8). The constraints in the model are used to impose that any edge with end vertices indiced by i and j can only belong to community s if both of its end vertices belong also to that

community, and to express the number of edges of each of the two communities and the sum of vertex degrees of the first one as a function of the variables X and Y. The model contains a single nonlinear but concave term, in the objective function, which is to be maximized. We thus obtained a quadratic convex mixed-integer program that can be solved exactly by `CPLEX` [68].

From a computational comparison with the most known hierarchical agglomerative heuristic [30] and with the divisive heuristic of [114], the proposed heuristic appears to exhibit better performance. Furthermore, the partitions obtained tend to have a modularity value close to that of optimal partitions (with an average error equal to 0.82540%). So, the partitions found can be considered as a fairly good approximation of the optimal ones.

Reformulations-based improvement of the divisive heuristic

In [Cafieri et al., 2012b] we explored reformulations (see Chap. 1, Sect. 1.3) of the mathematical programming model for bipartitions, to enhance efficiency of the proposed divisive heuristic. We presented several reformulations of the original model. They include reformulations aiming to reduce the dimension of the problem (number of variables and constraints), reformulations aiming to linearize nonlinear terms by binary decompositions and a reformulation based on symmetry breaking constraints. The path of reformulations leading to the *best formulation* for the bipartition problem (in this context, the one that provides the optimal solution in less time) appears to be the one passing through a few steps whose effect is to reduce the number of variables and constraints and to adjoin symmetry breaking constraints to the obtained compact formulation.

The proposed reformulations are based on a few considerations on the problem modeling and on known reformulation techniques. For example, we do not actually need to know if an edge is in the cluster 1 or 2, but only if it is within a cluster or not; hence, we can drop the index s of variables X and consider $X_{i,j} = 1$ if $Y_i = Y_j$ and 0 otherwise (where the meaningless index 1 from the Y variables is also dropped). Due to the elimination of the index s from the variables X, their number is halved. Some constraints can be adjoined observing that $X_{i,j}$ can be seen as the negation of the XOR operation between Y_i and Y_j variables, and these constraints can be in turn reduced in number observing that the coefficient of the variables X is positive in the objective function, and we are considering a maximization problem. A further reformulation step is based on the expression of variables X by employing the product of the variables Y_i and Y_j, as $X_{i,j} = 2Y_iY_j - Y_i - Y_j + 1$. Variables X can then replaced using a new set of variables representing the product of Y variables: $\forall (v_i, v_j) \in E_c \quad S_{i,j} = Y_iY_j$ (E_c being the set of edges of cluster c to be bipartitioned), and inequalities are used to describe the relationship between S and Y, which correspond to the classical Fortet's inequalities

for the exact linearization of a product of binary variables. Computational experiments show that the formulation using the S variables outperforms the one with X variables. The final new model is given by:

$$\max \quad \frac{1}{m}\left(\sum_{(v_i,v_j)\in E_c}(2S_{i,j}-Y_i-Y_j)+|E_c|-\frac{1}{2m}\left({D_1}^2+\frac{{D_c}^2}{2}-D_1D_c\right)\right) \tag{3.9}$$

$$\text{s.t.} \quad \forall(v_i,v_j)\in E_c \quad S_{i,j}\leq Y_i \tag{3.10}$$

$$\forall(v_i,v_j)\in E_c \quad S_{i,j}\leq Y_j \tag{3.11}$$

$$D_1=\sum_{v_i\in V_c}k_iY_i \tag{3.12}$$

$$\forall(v_i,v_j)\in E_c \quad S_{i,j}\in\mathbb{R} \tag{3.13}$$

$$D_1\in\mathbb{R} \tag{3.14}$$

$$\forall v_i\in V_c \quad Y_i\in\{0,1\}, \tag{3.15}$$

where in the objective function we use the fact that $\sum_{(v_i,v_j)\in E_C}1=|E_c|$.

Another reformulation of the original model is based on the binary decomposition technique recently employed for mixed-integer quadratic programming in [22]. The term D_1 appearing squared in the objective function can be rewritten as $D_1=\sum_{l=0}^{t}2^la_l$, where a_l are binary variables, and t is a parameter which can be estimated. Using this definition, ${D_1}^2$ can be expressed as

$${D_1}^2=\sum_{l=0}^{t}2^la_l\cdot\sum_{h=0}^{t}2^ha_h=\sum_{l=0}^{t}\sum_{h=0}^{t}2^{l+h}a_la_h=\sum_{l=0}^{t}\sum_{h=0}^{t}2^{l+h}R_{lh}=\sum_{l=0}^{t}2^{2l}a_l+\sum_{l=0}^{t}\sum_{h<l}2^{l+h+1}R_{lh},$$

where R are the variables used to replace the products between the variables a. The Fortet inequalities can be used to express this relationship. The resulting formulation is a linear program in 0-1 variables, which can be written in a more compact form again on the basis of simple considerations on the form of the involved mathematical expressions.

Finally, a symmetry breaking constraint, which is beneficial for the Branch-and-Bound algorithm applied to solve the bipartition problem (for example, the one implemented in `CPLEX`), is easily obtained by fixing the vertex with highest degree to belong to one of the two clusters.

Numerical results show that the proposed reformulations of the quadratic model significantly reduce the computational time to solve the bipartition problem in our divisive heuristic.

3.2.3 Improving clustering solutions by exact splitting

The exact algorithm for graph bipartition which is the basis of our proposed hierarchical divisive heuristic is also exploited in [Cafieri et al., 2012d] to devise an approach to be

applied as a post-processing to heuristic clustering methods in order to improve their performances. Given a partition found by a heuristic, one can apply another heuristic or an exact algorithm to the subnetworks induced by the communities found. This will eventually lead to a new, better, partition. Moreover, this refinement can be based on splitting a community or merging a pair of communities. In our approach, in the spirit of *matheuristics*, an exact algorithm for bipartition (see Sect. 3.2.2) is applied first to the communities considered one at a time, then merging pairs of communities and applying again the bipartition algorithm.

- First, we split each community of the original partition into two sub-communities by applying the exact algorithm for bipartition. If the modularity value corresponding to the obtained bipartition is higher than the one of the original community, then the original community is replaced by the two new communities. Otherwise the two obtained communities are discarded and the original one is kept.

- Second, we merge provisionally pairs of communities and check if this induces an increased value for modularity. For each pair of communities, if the new community containing all vertices of this pair has a modularity value higher than the sum of the modularities of the two original communities, then the new large community is kept in place of the other two. Otherwise, if merge is not beneficial, we try to split the merged community using again the exact algorithm for bipartitioning. Obviously, pairs of clusters to be merged can be selected according to different criteria. We compute the number of edges joining pairs of clusters and sort the pairs by decreasing number of joining links. In this way, we first attempt to improve the current partition by merging clusters which are more strongly connected than others.

Results of computational experiments carried out applying the proposed approach as post-processing to well known available heuristics (the agglomerative hierarchical heuristic of Clauset et al. [30], the partitioning heuristic of Noack and Rotta [120], the multistep greedy with vertex move heuristic of Schuetz and Caflisch [136] and the locally optimal divisive hierarchical heuristic of Sect. 3.2.2) showed the efficiency of the proposed approach, that improved all the results given by the heuristics and in several cases transformed the original partitions into optimal ones.

3.3 Other clustering criteria

Modularity, despite being by far the most employed criterion for network clustering, it is not the only one. As mentioned in Sect. 3.1, interesting research directions concern clustering criteria, the main reason being that currently there is not a criterion that is fully satisfactory in all applications. This in turn motivates future research directions (see Conclusions and Perspectives of this document). In this context, we contributed in [Cafieri et al., 2010a] with a new criterion (*edge-ratio*) and an algorithm for the

corresponding optimization problem, and in [Cafieri et al., 2012a] with an extension of the concept of communities in the *strong sense* and with (exact) algorithms for detecting such communities.

Both contributions build upon the observation that an alternative approach to the maximization of a criterion function, like modularity, for finding communities is based on the satisfaction of reasonable *a priori* conditions to have a community. Radicchi et al. [128] proposed two such conditions defining communities in a strong and a weak sense, respectively. Recall that the degree k_i of a vertex i belonging to V is the number of its neighbors (or adjacent vertices). Let $S \subseteq V$ be a subset of vertices. Then the degree k_i can be separated into two components $k_i^{in}(S)$ and $k_i^{out}(S)$, i.e., the number of neighbors of i inside S and the number of neighbors of i outside S.
A set of vertices S forms a community in the *strong sense* if and only if every one of its vertices has more neighbors within the community than outside:

$$k_i^{in}(S) > k_i^{out}(S), \quad \forall i \in S.$$

A set of vertices S forms a community in the *weak sense* if and only if the sum of all degrees within S is larger than the sum of all degrees joining S to the rest of the network:

$$\sum_{i \in S} k_i^{in}(S) > \sum_{i \in S} k_i^{out}(S).$$

This is equivalent to the condition that the number of edges within S is at least half the number of edges in the cut of S.
The weak condition was used [128] as a local stopping criterion in hierarchical clustering; also, it led Wang et al. [166] to define a community S *indivisible* if there is no bipartition, (S_1, S_2) of S, such that both S_1 and S_2 satisfy the weak condition.

Edge-ratio criterion

In [Cafieri et al., 2010a] we extend the definition of community in the weak sense into a criterion for a bipartition to be optimal: one seeks to maximize the minimum for both classes of the bipartition of the ratio of inner edges to cut edges. More precisely, we consider the ratio of the number of edges within a community to the number of cut edges which have one end point only within that community, i.e., denoting this ratio by $r(S)$, we have:

$$r(S) = \sum_{i \in S} k_i^{in}(S) / \sum_{i \in S} k_i^{out}(S).$$

When dividing S we consider this ratio for both communities S_1 and S_2 and maximize the smallest value, i.e., we address the problem (with $S_1 \cup S_2 = S$, $S_1 \cap S_2 = \emptyset$, $S_1, S_2 \neq \emptyset$):

$$\max_{S_1,S_2 \subset V} \min\left(r(S_1), r(S_2)\right).$$

Solving sequentially this problem yields a hierarchical divisive clustering algorithm, with a clear and well defined criterion, and that is *locally optimal* in the sense that each division is done in an optimal way. To build the divisive algorithm, we first strenghten the weak definition by quantifying how much the number of inner edges is larger than the number of cut edges. This is easily done by introducing a parameter α in the weak condition which then becomes equal to

$$\sum_{i \in S} k_i^{in}(S) \geq \alpha \sum_{i \in S} k_i^{out}(S). \tag{3.16}$$

So, in case of equality, the coefficient α is equal to the ratio of twice the number of edges within the community S divided by the number of edges within the cut of that community. We call it *edge ratio* for short. One can then seek, to perform a bipartition, the maximum value of α for which the network will be divisible: α will be equal to twice the ratio of the number of edges within S divided by the number of edges within the cut of S. The algorithm so includes the problem of detecting indivisible communities.
The mathematical programming formulation for identification of optimal communities according to the edge ratio criterion has a linear objective (i.e., maximization of α), but non linear and non convex constraints (products between α and binary variables arise). However, if α is fixed, a linear program in 0-1 variables is obtained. This suggests to solve the optimal bipartition problem with a dichotomous search on the values of α. An initial value α equal to 1 can first be chosen. If there is no feasible solution for that value, the network is indivisible. Otherwise, the value of α may be doubled and feasibility checked until a value is attained for which the weak condition cannot be satisfied. This gives an upper bound $\bar{\alpha}$ and the previous value of α gives a lower bound $\underline{\alpha}$. Then the dichotomous search proceeds by considering the mid value of the interval $[\underline{\alpha}, \bar{\alpha}]$. The procedure stops when the length $\bar{\alpha} - \underline{\alpha}$ of the current interval is smaller than some given tolerance ϵ. This basic procedure can be accelerated in several ways, including using an initial value of α corresponding to a solution obtained by some heuristic and removing symmetries by fixing a variable.

Comparing the proposed algorithm with modularity maximization, it appears not to suffer from the resolution limit problem (see Sect. 3.2) and usually identifies more communities, often with more precision. An example is presented in Figure 3.2. It consists of two large cliques joined by a single edge and two small cliques joined by an edge and also each joined by an edge to the same large clique. Maximizing modularity

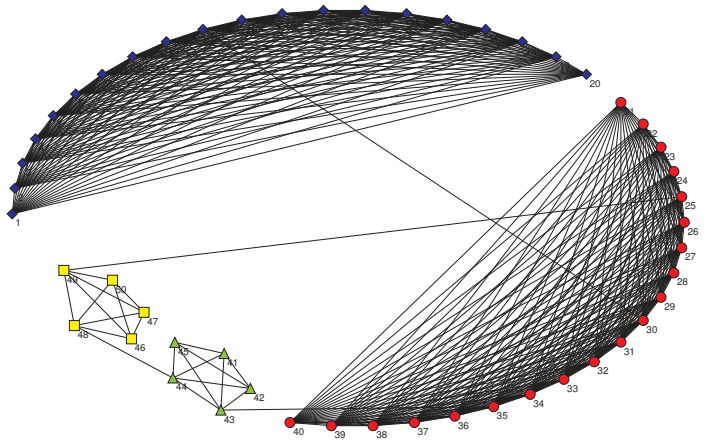


FIGURE 3.2: Partition obtained by the edge ratio algorithm on a dataset consisting of 4 cliques.

gives three communities corresponding to the two large cliques separately and to the union of the small ones, while the edge ratio algorithm gives four communities which correspond to each of the cliques.

Strong and almost-strong communities

In [Cafieri et al., 2012a] we define an enumerative algorithm, called SC (*Strong Communities*), to list all partitions in the strong sense (with only strong communities) of a network of moderate size. The algorithm makes use of two types of labels associated with the vertices and the edges respectively: label l_i associated with vertex v_i, $i = 1, \ldots, n$ (initially $l_i = i$ for all vertices, and at the current iteration the label of the vertex v_i is equal to the smallest label of a vertex of the community to which v_i belongs); the label $t_{i,j}$ associated with edge (v_i, v_j) can take three values $(-1, 0, 1)$. It is equal to -1 if it has already been decided that the vertices v_i and v_j belong to different communities; it is equal to 1 if it has already been decided that vertices v_i and v_j belong to the same community; if no decision has been taken, $t_{i,j} = 0$. The algorithm follows a number of rules, the most important of which concerns vertices with degree 2 and directly follows from the strong condition: if a vertex v_i has degree 2 and neighbors v_j and v_k, then all three vertices v_i, v_j, v_k must belong to the same community. All possible choices for labelling edges according to the strong community definition are considered (with branching rules), and consequently all the partitions corresponding to that definition are generated.

Computational experiments show that the strong condition is not easily satisfied: communities are often too large and correspond in fact to the union of several distinct communities; in the limit one finds a single community regrouping all vertices. The main reason for having large heterogeneous communities appears to be again due to the degree

2 vertices, since the strong condition imposes that them and both of their two neighbors belong to the same community, while these neighbors can be very different in terms of their own neighbors. This suggests to weaken the strong condition for the degree two vertices. We call a community *almost-strong* if it satisfies the condition $k_i^{in} \geq k_i^{out}$ (i.e., the strong condition weakened with a nonstrict inequality) for all vertices of degree two, and the strong condition $k_i^{in} > k_i^{out}$ for the remaining vertices. Minimum modifications brought to the SC algorithm lead to a modified algorithm called ASC (*Almost-Strong Communities*) to find partitions in the almost-strong sense. The algorithm ASC usually gives partitions into more communities than algorithm SC. Moreover, they are usually more intuitively appealing.

3.4 ATM application: airline networks

The algorithms presented in previous sections of this chapter have been always tested on a set of graph instances from the literature describing real-life applications. Examples are given in Fig. 3.1. However, none of these applications was specifically addressed, and the algorithms were developed in full generality for a wide range of applications. In the context of Air Traffic Management (ATM), which is becoming a privileged domain of application for my research activity, a few real-life problems arise (though different from ATM applications addressed in Chap. 1 and 2) that can be investigated by network clustering techniques. Airline networks can be naturally represented by graphs and their structural properties studied to analyze, for example, a geographical (and political) distribution of flights, airline companies strategies, or even identify the most suitable places to build hubs or where the flight density may be increased (for a study of a few aspects of the worldwide air transportation network, see [63]).

These applications have not been yet the topic of a specific contribution and will represent the subject of future research (see also Conclusions and Perspectives of this document). However, network clustering to analyze the distribution of flights has been the topic of a didactic project carried out by students at ENAC, that I proposed and supervised [41]. A graph was built having vertices and edges corresponding respectively to airports and flights between airports realized by airline companies. Network clustering performed by a simple maximizing modularity agglomerative heuristic and by spectral clustering [114] revealed that the most of the flights is in the upper hemisphere, that the most of transatlantic flights from Europe come from airports in the same cluster, and more generally showed, as expected, a geographical distribution of flights broadly corresponding to the different political states (whose airports are more interconnected than airports outside). See Fig. 3.3 for a graphical representation. This kind of applications will be furtherly investigated.

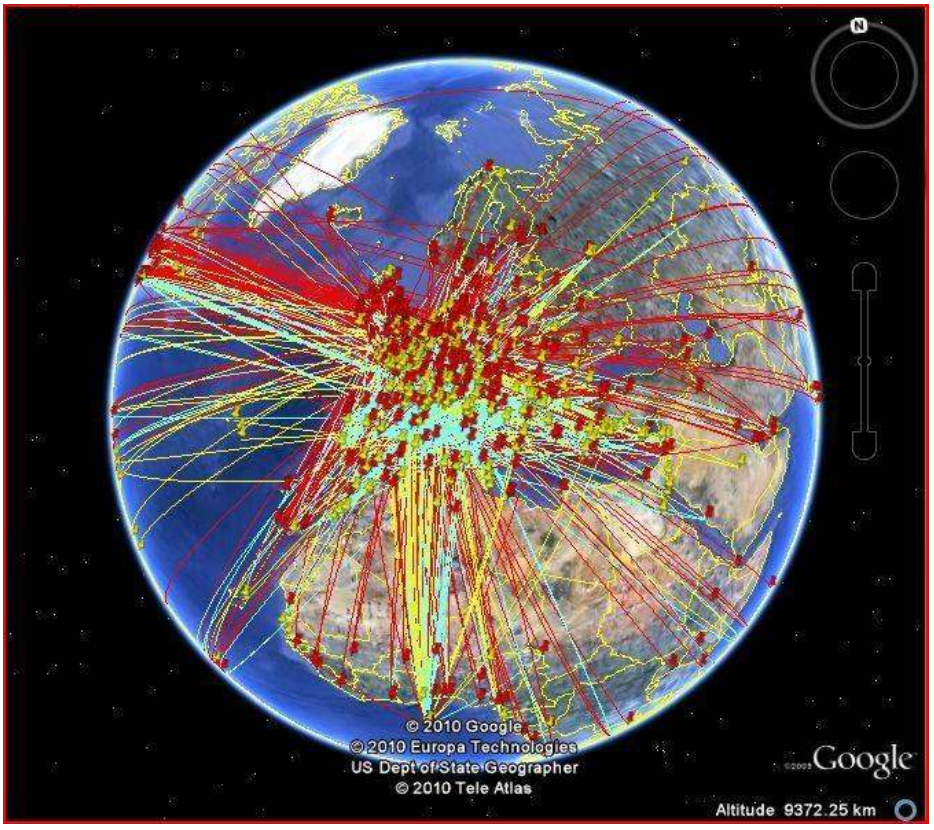

© 2010 Google
© 2010 Europa Technologies
US Dept of State Geographer
© 2010 Tele Atlas
Google
Altitude 9372.25 km

FIGURE 3.3

Conclusions and Perspectives

We presented the main research topics and contributions in the framework of three main themes in optimization, from local to global mixed-integer and continuous nonlinear optimization, passing through combinatorial optimization with a focus on graph clustering. A common thread can be identified in the interest for mathematical programming-based modeling, in a special attention for deterministic methods and in the constant numerical validation by development of suitable software tools. ATM applications also constitute a common point towards which different research themes are converging. This path is the result of a personal interest for various topics in optimization, but also of interesting and fructuous collaborations and exchanges with colleagues and research groups from different countries. The next sections highlight tracks for future research.

Perspectives in (mixed-integer and continuous) nonlinear optimization

The perspectives of the research topics described in Chapters 1 and 2 relate specially to the development expected in the context of a 3-years research project of which I am responsible and that has just been selected to be funded by the French National Agency of Research (ANR - Agence Nationale de Recherche). This project, *ATOMIC: Air Traffic Optimization via Mixed-Integer Computation*, summarizes already in its title its vocation to propose advancements in a precise applicative context, that of ATM and specifically of aircraft conflict avoidance and conflict-free trajectory planning, by means of Mixed-Integer Optimization tools. One of the aims of the project is to put together a small team of specialists, the most of which are young researchers, who work on complementary sub-domains of optimization and operations research and can convey their expertise around the project theme. The effort will concern, on the one hand, devising suitable modeling and, on the other hand, conceiving appropriate solution methods and algorithms. As discussed in Chap. 1 Sect. 1.4, the main difficulties for mathematical optimization in the considered context are related to the size of the addressed problems, and, even more challengingly, to the need of simultaneously considering combinatorial decisions and modeling nonlinear processes (e.g., aircraft separation

conditions). The focus will be on the deterministic solution of the proposed mathematical programming models, a guaranteed global optimal solution being interesting in an ATM operational context. The complexity of the addressed problems leading to very difficult, large-scale MINLPs, the idea is to explore suitable reformulation techniques to obtain formulations that are more convenient from the point of view of the application of algorithms, and a special attention will be devoted to devising MINLP cuts and symmetry-breaking reformulation constraints [88, 87]. Cuts are constraints valid for the feasible region of the original problem that are designed to cut off a part of the relaxed feasible region, thereby improving MINLP relaxations and hence speeding up Branch-and-Bound algorithms. Aircraft conflict resolution usually yields mathematical programming formulations with a high degree of symmetry, which greatly slows down the solution process by a BB-based algorithm. Symmetry-breaking narrowing reformulations will be investigated to cope with this issue. The interval-Branch-and-Bound variant, based on interval-arithmetic techniques, which has been already successfully employed for the solution of several nonlinear nonconvex continuous and mixed-integer problems arising from real-life applications, will also constitute a point of investigation. The two BB variants, spatial and interval-based, have to be applied to the conceived models including suitable strategies tailored on the problem at hand to enhance efficiency. It is however very hard to solve real-life large-scale problems by means of these approaches. Other strategies will be implemented to deal with the computational difficulty of the problem. Devising optimization algorithms based on the hybridization of mathematical programming techniques and (meta)heuristics [95] seems to offer a viable alternative, although it does not guarantee the global optimality of the solution. In this "matheuristic" framework (an example of which has been already presented in Sect. 1.4), an essential feature is the exploitation of the characteristics of the conceived mathematical programming models of the addressed problem. The aim will be to obtain efficiently, in reasonable computational time, "good" bounds on the optimal solution value.

Perspectives in network clustering

The research activity on network clustering will be pursued addressing methodological aspects as well as applications to real-life problems.

An application to air transportation networks has already been mentioned in Chap. 3, Sect. 3.4. The idea is to apply the developed approaches to networks built on real air traffic data to fully analyze the structural properties of the networks, and thereby to identify properties of the underlying complex air transportation system. This may be a good complement to the studies that will be undertaken on ATM problems in the context of nonlinear optimization (continuous and mixed-integer). Real data corresponding

to large-scale problems, exact methods (see Chap. 3, Sect. 3.2.1) are unlikely to be successful. Suitable heuristics, or hybrid methods, will be specially devised.

Concerning clustering criteria and clustering methods, the research will be carried out trying, on the one hand, to propose efficient solution approaches, specially driven by applications, and, on the other hand, to provide advancements on the open question of finding the "best" clustering criterion. More precisely, it will be further pursued along the following directions.

First, we intend to develop specialized algorithms for some classes of graphs. This is currently being undertaken for graph trees, based on a dynamical programming approach for modularity maximization (joint work with P. Hansen).

Second, we shall devise efficient methods for large-scale problems. For these problems, one has in general to resort to heuristics or hybrid methods. The acquired expertise, specially on exact algorithms, can be of help. Comparing the exact solution with those obtained by some heuristic can allow to identify parts of the problem that are badly solved and can suggest how to improve the heuristic. Conversely, heuristics can be used in order to accelerate one or several steps of the exact algorithm, e.g., by providing a hot start.

Third, we plan to analyze thoroughly different conditions for a community to be considered as such. To that effect, mathematical programming formulations and suitable solution approaches for the corresponding optimization problem will be developed (this has been already started out concerning the strong condition, see Chap. 3, Sect. 3.3).

Fourth, we shall combine a criterion for evaluating community partitions, like modularity, with conditions on each community, like the strong and weak conditions. This has been pioneered by Medus and Dorso [98] and can be pursued in two ways: (i) modifying the models and the corresponding algorithms or heuristics to incorporate the constraints on the communities from the outset, or (ii) using a standard algorithm or heuristic and then checking if the obtained communities satisfy the given constraints; if not, applying some specific heuristic to restore feasibility.

Finally, we will try to deduce from the results of investigation planned in previous points, a set of conditions, or axioms, for clustering on networks, to identify a "good" clustering criterion.

General Bibliography

[1] W.P. Adams and H.D. Sherali. A tight linearization and an algorithm for 0-1 quadratic programming problems. *Management Science*, 32(10):1274–1290, 1986.

[2] C.S. Adjiman. *Global Optimization Techniques for Process Systems Engineering*. PhD thesis, Princeton University, 1998.

[3] C.S. Adjiman, I.P. Androulakis, and C.A. Floudas. Global optimization of MINLP problems in process synthesis and design. *Computers and Chemical Engineering*, 21:445–450, 1997.

[4] C.S. Adjiman, I.P. Androulakis, and C.A. Floudas. A global optimization method,α-BB, for general twice-differentiable constrained NLPs: Ii. Implementation and computational results. *Computers and Chemical Engineering*, 22:1159–1179, 1998.

[5] C.S. Adjiman, S. Dallwig, C.A. Floudas, and A. Neumaier. A global optimization method,α-BB, for general twice-differentiable constrained NLPs: I. Theoretical advances. *Computers and Chemical Engineering*, 22:1137–1158, 1998.

[6] G. Agarwal and D. Kempe. Modularity-maximizing graph communities via mathematical programming. *The European Physical Journal B*, 66(3):409–418, 2008.

[7] F.A. Al-Khayyal and J.E. Falk. Jointly constrained biconvex programming. *Mathematics of Operations Research*, 8(2):273–286, 1983.

[8] A. Alonso-Ayuso, L.F. Escudero, and F.J. Martín-Campo. Collision Avoidance in Air Traffic Management: a Mixed-Integer Linear Optimization Approach. *IEEE Transactions on Intelligent Transportation Systems*, 12(1):47–57, 2011.

[9] A. Alonso-Ayuso, L.F. Escudero, and F.J. Martín-Campo. A mixed 0-1 nonlinear optimization model and algorithmic approach for the collision avoidance in ATM: Velocity changes through a time horizon. *Computers and Operations Research*, 2012. in press.

[10] E.D. Andersen, J. Gondzio, C. Meszaros, and X. Xu. Implementation of interior point methods for large scale linear programming. In T. Terlaky, editor, *Interior Point Methods in Mathematical Programming*, pages 189–252. Kluwer Academic Publisher, 1996.

[11] I.P. Androulakis, C.D. Maranas, and C.A. Floudas. α-BB: a global optimization method for general constrained nonconvex problems. *Journal of Global Optimization*, 7:337–363, 1995.

[12] MOSEK ApS. The mosek optimization tools version 3.1 (revision 28). Users manual and reference, 2002. `http://www.mosek.com/`.

[13] S. Balram. Crude transshipment via floating, production, storage and offloading platforms. Masters thesis, Dept. of Chemical and Biomolecular Engineering, National University of Singapore, 2010.

[14] A.-L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[15] 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.

[16] M. Benzi, G. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2005.

[17] M. Benzi and V. Simoncini. On the eigenvalues of a class of saddle point matrices. *Numerische Mathematik*, 103:173–196, 2006.

[18] L. Bergamaschi, J. Gondzio, M. Venturin, and G. Zilli. Inexact constraint preconditioners for linear systems arising in interior point methods. *Computational Optimization and Applications*, 36(2-3):137–147, 2007.

[19] L. Bergamaschi, J. Gondzio, and G. Zilli. Preconditioning indefinite systems in interior point methods for optimization. *Computational Optimization and Applications*, 28(2):149–171, 2004.

[20] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[21] A. Bicchi, A. Marigo, G. Pappas, M. Pardini, G. Parlangeli, C. Tomlin, and S.S. Sastry. Decentralized air traffic management systems: performance and fault tolerance. In *Proceedings of IFAC International Workshop on Motion Control, Grenoble*, pages 279–284, 1998.

[22] A. Billionnet, S. Elloumi, and A. Lambert. Extending the QCR method to general mixed-integer programs. *Mathematical Programming*, 131:381–401, 2012.

[23] V.D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics*, P10008, 2008.

[24] D. Bonini, C. Dupré, and G. Granger. How ERASMUS can support an increase in capacity in 2020. In *Proceedings of the 7th International Conference on Computing, Communications and Control Technologies: CCCT 2009, Orlando, Florida*, 2009.

[25] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner. On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):172–188, 2008.

[26] A. Brook, D. Kendrick, and A. Meeraus. GAMS, a user's guide. *ACM SIGNUM Newsletter*, 23(3-4):10–11, 1988.

[27] S. Cafieri. *On the application of iterative solvers to KKT systems in Interior Point methods for Large-Scale Quadratic Programming problems.* PhD thesis, University of Naples "Federico II", 2006.

[28] S. Cafieri. Aircraft conflict avoidance: a mixed-integer nonlinear optimization approach. In *Proceedings of the Global Optimization Workshop (GOW'12)*, pages 43–46, Natal, 2012.

[29] S. Cafieri and N. Durand. A mixed-integer optimization model for air traffic deconfliction. In *Proceedings of the Toulouse Global Optimization workshop*, 2010.

[30] A. Clauset, M.E.J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70:066111, 2004.

[31] A. Costa and L. Liberti. Relaxations of multilinear convex envelopes: dual is better than primal. In *Experimental Algorithms (Proceedings of SEA 2012)*, volume 7276 of *Lecture Notes in Computer Science*, pages 87–98. Springer, 2012.

[32] L. Danon, A. Diaz-Guilera, and A. Arenas. The effect of size heterogeneity on community identification in complex networks. *Journal of Statistical Mechanics*, P11010, 2006.

[33] H. Dollar and A. Wathen. Incomplete factorization constraint preconditioners for saddle-point matrices. *Numerical Analysis Group, Oxford University, Research Rep. NA-04/01*, 2004.

[34] H.S. Dollar and A.J. Wathen. *SIAM Journal of Scientific Computing*, 27(Approximate factorization constraint preconditioners for saddle-point matrices):1555–1572, 2006.

[35] O. du Merle, D. Villeneuve, J. Desrosiers, and P. Hansen. Stabilized column generation. *Discrete Mathematics*, 194:229–237, 1999.

[36] N. Durand and J.M. Alliot. Optimal resolution of en-route conflict. In *Proceedings of the Eurocontrol/FAA ATM Seminar*. Eurocontrol/FAA, 1997.

[37] N. Durand and J.M. Alliot. Ant-colony optimization for air traffic conflict resolution. In *Proceedings of the Eighth USA/Europe Air Traffic Management Research and Development Seminar*. Eurocontrol/FAA, 2009.

[38] N. Durand, J.M. Alliot, and F. Medioni. Neural nets trained by genetic algorithms for collision avoidance. *Applied Artificial Intelligence*, 13(3):205–213, 2000.

[39] N. Durand, J.M. Alliot, and J. Noailles. Automatic aircraft conflict resolution using genetic algorithms. In *Proceedings of the Symposium on Applied Computing, Philadelphia*. ACM, 1996.

[40] C. Durazzi and V. Ruggiero. Indefinitely preconditioned conjugate gradient method for large sparse equality and inequality constrained quadratic problems. *Numerical Linear Algebra with Applications*, 10(8):673–688, 2003.

[41] N. Etoundi, T. Labarre, C. Martin, and V. Steyaert. Analyse d'un réseau de transport aérien. Rapport final, Projet de synthése ENAC, Toulouse, 2010.

[42] C.A. Floudas. *Deterministic Global Optimization: Theory, Methods and Applications*. Nonconvex Optimization and Its Applications. Kluwer Academic Publishers, 1999.

[43] C.A. Floudas. Global optimization in design and control of chemical process systems. *Journal of Process Control*, 10:125–134, 2001.

[44] A. Forsgren, P.E. Gill, and J.D. Griffin. Iterative solution of augmented systems arising in interior methods. *SIAM Journal on Optimization*, 18(2):666–690, 2007.

[45] A. Forsgren, P.E. Gill, and J.R. Shinnerl. Stability of symmetric ill-conditioned systems arising in interior methods for constrained optimization. *SIAM Journal on Matrix Analysis and Applications*, 17:187–211, 1996.

[46] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, 2010.

[47] S. Fortunato and M. Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences, USA*, 104(1):36–41, 2007.

[48] R. Fourer and D. Gay. *The AMPL Book*. Duxbury Press, Pacific Grove, 2002.

[49] A. Fügenschuh, H. Homfeld, H. Schülldorf, and S. Vigerske. Mixed-integer nonlinear problems in transportation applications. In H. Rodrigues et al., editor, *Proceedings of the 2nd International Conference on Engineering Optimization*, 2010.

[50] K. Fukuda. `cdd`. `www.ifor.math.ethz.ch/~fukuda/cdd_home/cdd.html`.

[51] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.

[52] E.M. Gertz and S.J. Wright. Object-Oriented Software for Quadratic Programming. *ACM Transactions on Mathematical Software*, 29:58–81, 2003. `http://www.cs.wisc.edu/~swright/ooqp/`.

[53] M. Girvan and M. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences, USA*, 99(12):7821–7826, 2002.

[54] G.H. Golub and A.J. Wathen. An iteration for indefinite systems and its application to the Navier-Stokes equation. *SIAM Journal on Scientific Computing*, 19:530–539, 1998.

[55] J. Gondzio. Interior Point methods 25 years later. *European Journal of Operations Research*, 218:587–601, 2011.

[56] B. H. Good, Y.-A. de Montjoye, and A. Clauset. The performance of modularity maximization in practical contexts. *Physical Review E*, 81:046106, 2010.

[57] C.E. Gounaris and C.A. Floudas. Tight convex underestimators for C^2-continuous problems: I. Multivariate functions. *Journal of Global Optimization*, 42(1):69–89, 2008.

[58] C.E. Gounaris and C.A. Floudas. Tight convex underestimators for C^2-continuous problems: I. Univariate functions. *Journal of Global Optimization*, 42(1):51–67, 2008.

[59] G. Granger and N. Durand. A traffic complexity approach through cluster analysis. In *Proceedings of the 5th ATM R&D Seminar, Budapest*, 2003.

[60] M. Grötschel and Y. Wakabayashi. A cutting plane algorithm for a clustering problem. *Mathematical Programming*, 45:59–96, 1989.

[61] M. Grötschel and Y. Wakabayashi. Facets of the clique partitioning polytope. *Mathematical Programming*, 47:367–387, 1990.

[62] R. Guimerà and A.N. Amaral. Functional cartography of complex metabolic networks. *Nature*, 433:895–900, 2005.

[63] R. Guimera, S. Mossa, A. Turtschi, and A.N. Amaral. The worldwide air transportation network: anomalous centrality, community structure, and cities' global roles. *Proc. Natl. Acad. Sci. USA*, 102(22):7794–7799, 2005.

[64] P. Hansen, B. Jaumard, and C. Meyer. A simple enumerative algorithm for unconstrained 0–1 quadratic programming. *Les Cahiers du GERAD*, G-2000-59, 2000.

[65] P. Hansen and N. Mladenovic. Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130(3):449–467, 2001.

[66] S. Huang, M. Aydin, and T.A. Lipo. A direct approach to electrical machine performance evaluation: Torque density assessment and sizing optimisation. In *International Conference on Electrical Machines*, 2002. Art. 235.

[67] V. Hugo. *Les Misérables*. Gallimard, Bibliothéque de la Pleiade, Paris, 1951.

[68] ILOG. *ILOG CPLEX 11.0 User's Manual*. ILOG S.A., Gentilly, France, 2008.

[69] B. Jeannet and F. Messine. Extensions of interval branch-and-bound algorithms for mixed global optimization problems with real and categorical variables. In S. Cafieri, B.G. Toth, E.M.T. Hendrix, L. Liberti, and F. Messine, editors, *Toulouse Global Optimization Workshop*, pages 67–70, Toulouse, 2010.

[70] Y. Jiao, F.H. Stillinger, and S. Torquato. Geometrical ambiguity of pair statistics I. Point configurations. Technical Report 0908.1366v1, arXiv, 2009.

[71] J. Kallrath. Solving planning and design problems in the process industry using mixed integer and global optimization. *Annals of Operations Research*, 140(1):339–373, 2005.

[72] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[73] C. Keller, N. Gould, and A. Wathen. Constraint preconditioning for indefinite linear systems. *SIAM Journal on Matrix Analysis and Application*, 21(4):1300–1317, 2000.

[74] B.W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, pages 291–307, 1970.

[75] D.E. Knuth. *The Stanford GraphBase: A Platform for Combinatorial Computing*. Addison-Wesley, Reading, MA, 1993.

[76] V. Krebs. `http://www.orgnet.com/` (unpublished).

[77] J. Kuchar and L. Yang. A review of conflict detection and resolution modeling methods. *IEEE Trans. on Intelligent Transportation Systems*, 1(4):179–189, 2000.

[78] A. Lancichinetti and S. Fortunato. Limits of modularity maximization in community detection. *Physical Review E*, 84:066122, 2011.

[79] C. Lavor, L. Liberti, N. Maculan, and M.A. Chaer Nascimento. Solving Hartree-Fock systems with global optimization metohds. *Europhysics Letters*, 77(5):50006, 2007.

[80] R.D. Lazarov and P.S. Vassilevski. Preconditioning saddle-point problems arising from mixed finite element discretizations of elliptic equations. *Numerical Linear Algebra with Applications*, 3(1):1–20, 1996.

[81] J. Lee and W.D. Morris, Jr. Geometric comparison of combinatorial polytopes. *Discrete Applied Mathematics*, 55(2):163–182, 1994.

[82] A. Letchford and S. Burer. Non-convex mixed-integer nonlinear programming: a survey. *Surveys in Operations Research and Management Science*, 2012. In press.

[83] L. Liberti. Linearity embedded in nonconvex programs. *Journal of Global Optimization*, 33(2):157–196, 2005.

[84] 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.

[85] L. Liberti. Reformulation techniques in mathematical programming, November 2007. Thèse d'Habilitation à Diriger des Recherches, Université de Paris-Dauphine.

[86] L. Liberti. Reformulations in mathematical programming: Definitions and systematics. *RAIRO-RO*, 43(1):55–86, 2009.

[87] L. Liberti. Reformulations in mathematical programming: Automatic symmetry detection and exploitation. *Mathematical Programming A*, 131:273–304, 2012.

[88] L. Liberti. Symmetry in mathematical programming. In J. Lee and S. Leyffer, editors, *Mixed Integer Nonlinear Programming*, IMA Series. Springer, to appear.

[89] L. Liberti, C. Lavor, and N. Maculan. A branch-and-prune algorithm for the molecular distance geometry problem. *International Transactions in Operational Research*, 15:1–17, 2008.

[90] 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.

[91] L. Liberti and C. Pantelides. An exact reformulation algorithm for large nonconvex NLPs involving bilinear terms. *Journal of Global Optimization*, 36:161–189, 2006.

[92] L. Liberti and C.C. Pantelides. Convex envelopes of monomials of odd degree. *Journal of Global Optimization*, 25:157–168, 2003.

[93] R. Lougee-Heimer. The common optimization interface for operations research: Promoting open-source software in the operations research community. *IBM Journal of Research and Development*, 47(1):57–66, 2003.

[94] L. Luksan and J. Vlcek. Indefinitely preconditioned inexact Newton method for large sparse equality constrained nonlinear programming problems. *Numerical Linear Algebra with Applications*, 5:219–247, 1998.

[95] V. Maniezzo, T. Stutzle, and S. Voss, editors. *Hybridizing metaheuristics and mathematical programming*. Springer, New York, 2009.

[96] C.P. Massen and J.P.K. Doye. Identifying communities within energy landscapes. *Physical Review E*, 71:046101, 2005.

[97] G.P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I — Convex underestimating problems. *Mathematical Programming*, 10:146–175, 1976.

[98] A. D. Medus and C. O. Dorso. Alternative approach to community detection in networks. *Physical Review E*, 79(6):066111, 2009.

[99] F. Messine. Deterministic global optimization using interval constraint propagation techniques. *RAIRO-Operations Research*, 38:277–294, 2004.

[100] F. Messine. *A Deterministic Global Optimization Algorithm for Design Problems*, pages 267–292. Springer, 2005.

[101] F. Messine. A deterministic global optimization algorithm for design problems. In C. Audet, P. Hansen, and G. Savard, editors, *Essays and Surveys in Global Optimization*, pages 267–294. Kluwer, 2005.

[102] F. Messine and B. Nogaréde. Optimal design of a multi-airgap electrical machines: An unknown size mixed constrained global optimization formulation. *IEEE Transactions on Magnetics*, 40(3):3847–3853, 2006.

[103] F. Messine, B. Nogaréde, and J.L. Lagouanelle. Optimal design of electromechanical actuators: a new method based on global optimization. *IEEE Transactions on Magnetics*, 34(1):299–307, 1998.

[104] 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.

[105] 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.

[106] C.A. Meyer and C.A. Floudas. Convex envelopes for edge-concave functions. *Mathematical Programming*, 103:207–224, 2005.

[107] R. Misener and C.A. Floudas. Global optimization of mixed-integer quadratically-constrained quadratic programs (MIQCQP) through piecewise-linear and edge-concave relaxations. *Mathematical Programming B*, DOI: 10.1007/s10107-012-0555-6, in press.

[108] S. Mizuno, M. Kojima, and M.J. Todd. Infeasible-interior-point primal-dual potential-reduction algorithms for linear programming. *SIAM Journal on Optimization*, 5:52–67, 1995.

[109] N. Mladenovic and P. Hansen. Variable neighborhood search. *Computers and Operations Research*, 24(11):1097–1100, 1997.

[110] P. Mosses. Denotational semantics. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 575–631. 1990.

[111] W. Murray and U. Shanbhag. A local relaxation approach for the siting of electrical substations. *Computational Optimization and Applications*, 33:7–49, 2006.

[112] P. Neittaanmki, M. Rudnicki, and A. Savini. *Inverse Problems and Optimal Design in Electricity and Magnetism*. Clarendon Press, Oxford, 1996.

[113] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley and Sons, 1988.

[114] M. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences, USA*, 103(23):8577–8582, 2006.

[115] M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026133, 2004.

[116] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74:036104, 2006.

[117] M. E. J. Newman. *Networks: an introduction*. Oxford University Press, Oxford, 2010.

[118] M.E.J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69:066133, 2004.

[119] J. Ninin and F. Messine. A mixed affine reformulation method for global optimization. In S. Cafieri, B.G. Toth, E.M.T. Hendrix, L. Liberti, and F. Messine, editors, *Toulouse Global Optimization Workshop*, pages 101–104, Toulouse, 2010.

[120] A. Noack and R. Rotta. Multi-level algorithms for modularity clustering. *Lecture Notes in Computer Science*, 5526:257–268, 2009.

[121] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, 1999.

[122] B. Nogaréde, A.D. Kone, and M. Lajoie-Mazenc. Optimal design of permanent-magnet machines using analytical field modeling. *Electromotion*, 2(1):25–34, 1995.

[123] D.P. O'Leary. Symbiosis between linear algebra and optimization. *Journal of Computational and Applied Mathematics*, 123:447–465, 2000.

[124] L. Pallottino, E. Feron, and A. Bicchi. Conflict resolution problems for air traffic management systems solved with mixed integer programming. *IEEE Transactions on Intelligent Transportation Systems*, 3(1):3–11, 2002.

[125] P.M. Pardalos and M.G.C. Resende, editors. *Handbook of Applied Optimization*. Oxford University Press, New York, 2002.

[126] P.M. Pardalos and H. Edwin Romeijn, editors. *Handbook of Global Optimization - Volume 2*. Nonconvex Optimization and Its Applications. Kluwer Academic Publishers, 2002.

[127] I. Perugia and V. Simoncini. Block-diagonal and indefinite symmetric preconditioners for mixed finite element formulations. *Numerical Linear Algebra with Applications*, 7:585–616, 2000.

[128] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences, USA*, 101(9):2658–2663, 2004.

[129] L. Rao, W. Yan, and R. He. Mean Field Annealing (MFA) and optimal design of electromagnetic devices. *IEEE Transactions on Magnetics*, 32:1218–1221, 1996.

[130] D. Rey, S. Constans, R. Fondacci, and C. Rapine. A mixed integer linear model for potential conflict minimization by speed modulations. In *Proceedings of the International Conference on Research in Air Transportation, Budapest*, 2010.

[131] D. Rey, C. Rapine, R. Fondacci, and N-E. El Faouzi. Potential air conflicts minimization through speed regulation. *Transportation Research Board*, 01362476, 2012.

[132] A. Rikun. A convex envelope formula for multilinear functions. *Journal of Global Optimization*, 10(4):425–437, 1997.

[133] M. Rozloznik and V. Simoncini. Krylov subspace methods for saddle point problems with indefinite preconditioning. *SIAM Journal on Matrix Analysis and Applications*, 24(2):368–391, 2002.

[134] H.S. Ryoo and N.V. Sahinidis. Global optimization of nonconvex NLPs and MINLPs with applications in process design. *Computers and Chemical Engineering*, 19:551–566, 1995.

[135] H.S. Ryoo and N.V. Sahinidis. A branch-and-reduce approach to global optimization. *Journal of Global Optimization*, 8:107–138, 1996.

[136] P. Schuetz and A. Caflisch. Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement. *Physical Review E*, 77:046112, 2008.

[137] SESAR Consortium: The European ATM Master Plan. Technical Report 1, European Commission and Eurocontrol, 2009.

[138] H. Sherali, E. Dalkiran, and L. Liberti. Reduced RLT representations for nonconvex polynomial programming problems. *Journal of Global Optimization*, 52:447–469, 2012.

[139] H. Sherali and C.H. Tuncbilek. A global optimization algorithm for polynomial programming problems using a reformulation-linearization technique. *Journal of Global Optimization*, 2:101–112, 1991.

[140] H. Sherali and C.H. Tuncbilek. New reformulation linearization/convexification relaxations for univariate and multivariate polynomial programming problems. *Operations Research Letters*, 21:1–9, 1997.

[141] H.D. Sherali. Convex envelopes of multilinear functions over a unit hypercube and over special discrete sets. *Acta Mathematica Vietnamica*, 22:245–270, 1997.

[142] H.D. Sherali. Global optimization of nonconvex polynomial programming problems having rational exponents. *Journal of Global Optimization*, 12:267–283, 1998.

[143] H.D. Sherali. Tight relaxations for nonconvex optimization problems using the reformulation-linearization/convexification technique (RLT). In P.M. Pardalos and H.E. Romeijn, editors, *Handbook of Global Optimization*, volume 2, pages 1–63. Kluwer Academic Publishers, 2002.

[144] H.D. Sherali and W.P. Adams. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*. Kluwer Academic Publishers, Dodrecht, 1999.

[145] H.D. Sherali and A. Alameddine. A new reformulation-linearization technique for bilinear programming problems. *Journal of Global Optimization*, 2:379–410, 1992.

[146] H.D. Sherali and H. Wang. Global optimization of nonconvex factorable programming problems. *Mathematical Programming*, 89:459–478, 2001.

[147] G.R. Slemon and X. Liu. Modeling and design optimization of permanent magnet motors. *Electric Machines and Power Systems*, 20:71–92, 1992.

[148] E.M.B. Smith. *On the Optimal Design of Continuous Processes*. PhD thesis, Imperial College of Science, Technology and Medicine, University of London, October 1996.

[149] E.M.B. Smith and C.C. Pantelides. Global optimization of nonconvex MINLPs. *Computers and Chemical Engineering Supplement*, 21:S791–S796, 1997.

[150] 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.

[151] Eurocontrol long-term forecast : IFR Flight Movements 2010-2030. Technical report, Eurocontrol - Air Traffic Statistics and Forecasts, 2010.

[152] K. Tanabe. Centered Newton method for mathematical programming. In M. Iri and K. Yajima, editors, *System Modeling and Optimization: Proceedings of the 13th IFIP Conference*, Lecture Notes in Control and Information Systems, pages 197–206. Springer-Verlag, 1988.

[153] F. Tardella. Existence and sum decomposition of vertex polyhedral convex envelopes. *Optimization Letters*, 2:363–375, 2008.

[154] 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 Publishers, Amsterdam, 2008.

[155] M. Tawarmalani and N.V. Sahinidis. Convex extensions and envelopes of lower semi-continuous functions. *Mathematical Programming*, 93(2):247–263, 2002.

[156] M. Tawarmalani and N.V. Sahinidis. Global optimization of mixed integer nonlinear programs: A theoretical and computational study. *Mathematical Programming*, 99:563–591, 2004.

[157] M.J. Todd. Potential-Reduction Methods in Mathematical Programming. *Mathematical Programming*, 76:3–45, 1996.

[158] M.J. Todd and Y. Ye. A centered projective algorithm for linear programming. *Mathematics of Operations Research*, 15:508–529, 1990.

[159] C. Tomlin, G.J. Pappas, and S.S. Sastry. Conflict resolution for air traffic management : a case study in multi-agent hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):509–521, 1998.

[160] H. Tuy. D.C. optimization: Theory, methods and algorithms. In R. Horst and P.M. Pardalos, editors, *Handbook of Global Optimization*, volume 1, pages 149–216. Kluwer Academic Publishers, Dordrecht, 1995.

[161] F. Vanderbeck and M.W.P. Savelsbergh. A generic view of Dantzig-Wolfe decomposition in mixed integer programming. *Operations Research Letters*, 34(3):296 – 306, 2006.

[162] R.J. Vanderbei. Symmetric quasi-definite matrices. *SIAM Journal on Optimization*, 5:100–113, 1995.

[163] R.J. Vanderbei. LOQO User's Manual- Version 3.10, 1997. Technical Report SOR-97-08, `http://www.orfe.princeton.edu/~loqo/`.

[164] A. Vela, S. Solak, W. Singhose, and J-P. Clarke. A mixed-integer program for flight-level assignment and speed control for conflict resolution. In *Proceedings of the joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference, Shanghai*. IEEE, 2009.

[165] K. Wakita and T. Tsurumi. Finding community structure in mega-scale social networks. Technical Report 0702048v1, arXiv, 2007.

[166] J. Wang, Y. Qui, R. Wang, and X. Zhang. Remarks on network community properties. *Journal of Systems Science and Complexity*, 21(4):637–644, 2008.

[167] D.S. Watts and S.H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):409–410, 1998.

[168] M.H. Wright. Ill-conditioning and computational error in primal-dual interior methods for nonlinear programming. *SIAM Journal on Optimization*, 9:84–111, 1998.

[169] S.J. Wright. *Primal-Dual Interior-Point Methods*. SIAM, 1997.

[170] S.J. Wright. Stability of augmented system factorizations in interior point methods. *SIAM Journal on Matrix Analysis and Applications*, 18:191–222, 1997.

[171] F. Wurtz, J. Bigeon, and C. Poirson. A methodology and a tool for the computer-aided design with constraints of electrical devices. *IEEE Transactions on Magnetics*, 32:1429–1432, 1996.

[172] G. Xu, S. Tsoka, and L.G. Papageorgiou. Finding community structures in complex networks using mixed integer optimization. *European Physical Journal B*, 60:231–239, 2007.

[173] W.W. Zachary. An information flow model for conflict and fission in small group. *Journal of Anthropological Research*, 33:452–473, 1977.

Other academic activities

Cursus

Education

- *Jan 2006*: **Ph.D. in Mathematical Sciences**
  Thesis title: *On the application of iterative solvers to KKT systems in Interior Point methods for Large-Scale Quadratic Programming problems.*
  University of Naples "Federico II", Italy.

- *Oct 2001*: **Laurea (Master) in Mathematics**
  110/110 cum Laude
  Thesis title: *Quadratic Optimization: Algorithms and Software for sparse problems.*
  Second University of Naples, Italy.

Employment history

- *Dec 2009 - present*
  Assistant Professor [1]
  École Nationale de l'Aviation Civile (ENAC),
  Departement of Mathematics and Computer Science,
  Laboratory of Applied Mathematics, Computer Science and Automation for Aeronautics (MAIAA: Mathématiques Appliquées, Informatique et Automatique pour l'Aérien), Toulouse, France.

- *Mar 2008 - Nov 2009*
  Post-doctoral Researcher
  Laboratoire d'Informatique (LIX), École Polytechnique, France.
  Topic: Reformulation techniques in Mathematical Programming.
  Research advisor: Prof. L. Liberti.

[1] *Enseignant-Chercheur*, equivalent to *Maître de Conférences* in France.

- *Jun 2007 - Feb 2008*
  Post-doctoral Researcher
  Interdipartimental Research Center Bioagromed, University of Foggia, Italy.
  Topic: PDE-based mathematical models for food science applications.
  Research advisor: Prof. M.A. Del Nobile.

- *Sep 2006 - May 2007*
  Research Fellow
  Department of Mathematics, Second University of Naples, Italy.
  Topic: Development of software tools for Quadratic Programming.
  Research advisor: Prof. M. D'Apuzzo.

- *Jun 2001 - Sep 2001*
  Stage at Center for Research on Parallel Computing and Supercomputers - CPS/CNR, nowadays *Institute for High Performance Computing and Networking* - ICAR/CNR - Naples branch.
  Topic: Algorithms and software for high performance computing.

Awards

- *Academic year 1998/1999*
  2^{nd} **place in the Annual Second University of Naples Prize**,
  for the best students during the academic year.

- *Academic year 1999/2000*
  1^{st} **place in the Annual Second University of Naples Prize**,
  for the best students during the academic year.

Editorial and organizational work

Editorial responsibilities

- Associate Editor for *International Transactions in Operational Research* (ITOR) since September 2011.
- Guest Editor with L. Liberti and F. Messine of a special issue of *Journal of Global Optimization* dedicated to the Toulouse Global Optimization workshop 2010, to appear.
- Guest Editor with U. Faigle and L. Liberti of a special issue of *Discrete Applied Mathematics* dedicated to the CTW09 conference, Volume 159 (16), pages 1659-1914, September 2011.

Conference organization

- Organizing co-chair of the Mixed-Integer Nonlinear Programming stream at EURO 2012 Conference (European Conference on Operations Research), Vilnius, Lithuania, July 2012.
- Scientific committee member in Global Optimization Workshop GOW12, Natal, Brasil, June 2012.
- Local organizing committee member in JFPC 2012 (French Conference on Constraint Programming), Toulouse, France, May 2012.
- Session co-chair at ROADEF 2012 (French Conference on Operations Research), Angers, France, April 2012. Session title "Transport et Controle Aérien".
- Session chair at OR 2011 (International Conference on Operations Research), Zurich, Switzerland, Sept 2011. Session title "Airline, airport and air traffic management".
- Session co-chair at ROADEF 2011 (French Conference on Operations Research), Saint Etienne, France, March 2011. Session title "Transport et Controle Aérien".
- Scientific and local organizing committee member in Toulouse Global Optimization workshop (TOGO10) in Toulouse, France, August-September 2010. Co-editor of the Conference Proceedings.
- Session chair within the stream "Mixed-Integer Nonlinear Programming" at EURO 2010 Conference (European Conference on Operations Research), Lisbon, Portugal, July 2010.

- Local organizing committee member in CTW09 international workshop on Graphs and Combinatorial Optimization in Paris, France, June 2009. Co-editor of the Conference Proceedings.

Reviewing work

Referee for

- International journals: Journal Of Global Optimization, Optimization Letters, Computational Optimization and Applications, Journal Of Control, Journal of Computer Mathematics, Information Processing Letters, Discrete Applied Mathematics, SIAM Journal on Optimization, TOPR, Annals of Operations Research, Optimization.
- International conferences: CTW09, TOGO10, SEA2012, GOW'12.

Scientific societies membership

- ROADEF (French Operations Research Society) since 2009
- MOS (Mathematical Optimization Society, previously Mathematical Programming Society) since 2010
- member of the research group TORO - Toulouse Operations Research and Optimization - which gathers people working on Operation Research and Optimization in research laboratories and Universities in Toulouse, France, since its foundation in 2011: `www.toro-toulouse.fr`

Grants and projects

- **Funded projects**
  - *ATOMIC: Air Traffic Optimization via Mixed-Integer Computation*, ANR JCJC Project, funded by French "Agence Nationale de la Recherche" (ANR), (Principal Investigator (PI), 189 KEUR, 3 years, started Jan. 2013).
  - *PhD fellowship funding*, awarded by PRES University of Toulouse; at ENAC (PI, 88 KEUR, 3 years, started Oct. 2011 (supervision of a PhD thesis)).
- **Participation in scientific projects**
  - *2012-2013*
    ORGE - Optimisation Robuste de dispositifs magnétiques à Grands Entrefers, BQR project funded by INPT-INSA-ISAE.

- *2009-2012*
  RMNCCO (project on Reformulations in Mathematical Programming), funded by Digiteo.
- *2008-2009*
  ARS - Automatic Reformulations Search, ANR JCJC Project, funded by French "Agence Nationale de la Recherche" (ANR), (post-doctoral research activity).
- *2005- 2007*
  Innovative Problems and Methods in Nonlinear Optimization, PRIN Project, funded by Italian Ministry of University and Research (MIUR).
- *2003- 2006*
  Large Scale Nonlinear Optimization, FIRB Project, funded by Italian MIUR.

List of publications and presentations

Publications

- **International Journals**

1. S. Cafieri, L. Liberti, F. Messine, B. Nogarede, *Optimal Design of Electrical Machines: Mathematical Programming Formulations*, **COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering**, in press, 2012.
2. S. Cafieri, P. Hansen, L. Liberti, *Improving heuristics for network modularity maximization using an exact algorithm*, **Discrete Applied Mathematics**, DOI 10.1016/j.dam.2012.03.030, in press, 2012.
3. S. Cafieri, G. Caporossi, P. Hansen, S. Perron, A. Costa, *Finding communities in networks in the strong and almost-strong sense*, **Physical Review E**, 85(4):046113, 2012.
4. S. Cafieri, P. Hansen, L. Liberti, *Locally optimal heuristic for modularity maximization of networks*, **Physical Review E**, 83(5):056105, 2011.
5. D. Aloise, S. Cafieri, G. Caporossi, P. Hansen, L. Liberti, S. Perron, *Column generation algorithms for exact modularity maximization in networks*, **Physical Review E**, 82(4):046112, 2010.
6. S. Cafieri, P. Hansen, L. Liberti, *Loops and multiple edges in modularity maximization of networks*, **Physical Review E**, 81(4):046102, 2010.

7. S. Cafieri, P. Hansen, L. Liberti, *Edge ratio and community structure in networks*, **Physical Review E**, 81(2):026105, 2010.
8. S. Cafieri, J. Lee, L. Liberti, *On convex relaxations of quadrilinear terms*, **Journal of Global Optimization**, 47:661–685, 2010.
9. S. Cafieri, M. Mastromatteo, S. Chillo, M.A. Del Nobile, *Modeling the mechanical properties of pasta cooked at different times*, **Journal of Food Engineering**, 100: 336–342, 2010.
10. S. Cafieri, S. Chillo, M. Mastromatteo, N. Suriano, M.A. Del Nobile, *A mathematical model to predict the effect of shape on pasta hydration kinetic during cooking and overcooking*, **Journal of Cereal Science**, 48 (3): 857–862, 2008.
11. S. Cafieri, M. D'Apuzzo, V. De Simone, D. di Serafino, G. Toraldo, *Convergence Analysis of an Inexact Potential Reduction Method for Convex Quadratic Programming*, **Journal of Optimization Theory and Applications**, 135: 355–366, 2007.
12. S. Cafieri, M. D'Apuzzo, V. De Simone, D. di Serafino, *Stopping criteria for inner iterations in inexact Potential Reduction methods: a computational study*, **Computational Optimization and Applications**, special issue on Linear Algebra issues arising in Interior Point methods, J. Gondzio and G. Toraldo eds., 36 (2): 165-193, 2007.
13. S. Cafieri, M. D'Apuzzo, V. De Simone, D. di Serafino, *On the Iterative Solution of KKT Systems in Potential Reduction Software for Large Scale Quadratic Problems*, **Computational Optimization and Applications**, special issue on High Performance Algorithms and Software for Nonlinear Optimization, A. Murli and G. Toraldo eds, 38: 27–45, 2007.
14. S. Cafieri, M. D'Apuzzo, M. Marino, A. Mucherino, G. Toraldo, *Interior Point Solver for Large-Scale Quadratic Programming Problems with Bound Constraints*, **Journal of Optimization Theory and Applications**, 129 (1): 55–75, 2006.

- **Edited volumes and journal issues**

15. S. Cafieri, L. Liberti, F. Messine (eds.), *Toulouse Global Optimization Workshop 2010*, **special issue of the Journal of Global Optimization** dedicated to the TOGO10 Conference, to appear.
16. S. Cafieri, U. Faigle, L. Liberti (eds.), *Graphs and Combinatorial Optimization*, **special issue of Discrete Applied Mathematics** dedicated to the CTW09 Conference, Volume 159 (16), pages 1659-1914, 2011.

17. S. Cafieri, B.G. Tóth, E.M.T. Hendrix, L. Liberti, F. Messine (eds.), *Proceedings of the Toulouse Global Optimization workshop* (TOGO10), Toulouse, 2010.
18. S. Cafieri, A. Mucherino, G. Nannicini, F. Tarissan, L. Liberti (eds.), *Proceedings of CTW09 Conference on Graphs and Combinatorial Optimization*, Paris, 2009.

- **Book chapters (refereed)**

19. P. Belotti, S. Cafieri, J. Lee, L. Liberti, A. Miller, *On the composition of convex envelopes for quadrilinear terms*, in A. Chinchuluun, P.M. Pardalos, R. Enkhbat and E.N. Pistikopoulos (eds.), Proceedings of the International Conference on Optimization, Simulation and Control, **Series : Springer Optimization and its Application**, to appear.
20. L. Liberti, S. Cafieri, F. Tarissan, *Reformulations in Mathematical Programming: a Computational Approach*, in A. Abraham, A.-E. Hassanien, P. Siarry, and A. Engelbrecht (eds.), Foundations of Computational Intelligence Vol. 3 (Global Optimization: Theoretical Foundations and Applications), **Series: Studies in Computational Intelligence**, 203:153-234, Springer, Berlin, 2009.

- **International Conference publications (refereed)**

21. L. Cellier, S. Cafieri, F. Messine, *Hybridizing direct and indirect optimal control approaches for aircraft conflict avoidance*, in **Proceedings of ADVCOMP 2012: The Sixth International Conference on Advanced Engineering Computing and Applications in Sciences**, pp. 42-45, Barcelone, 2012.
22. S. Cafieri, *Aircraft conflict avoidance: A mixed-integer nonlinear optimization approach*, in **Proceedings of Global Optimization Workshop (GOW'12)**, pp. 43-46, Natal, 2012.
23. S. Cafieri, P. Hansen, L. Létocart, L. Liberti, F. Messine, *Compact relaxations for polynomial programming problems*, in R. Klasing (eds.), Experimental Algorithms (Proceedings of SEA 2012), **Lecture Notes in Computer Science** 7276:75-86, Springer, Berlin, 2012.
24. P. Belotti, S. Cafieri, L. Liberti, J. Lee, *Feasibility-based bounds tightening via fixed points*, in W. Wu and O. Daescu (eds.), Proceedings of Conference on Combinatorial Optimization and Applications (COCOA 2010), **Lecture Notes in Computer Science**, 6508:65-76, 2010.

25. S. Cafieri, L. Liberti, F. Messine, B. Nogarede, *Discussion about formulations and resolution techniques of electrical machine design problems*, in Proceedings of XIX International Conference on Electrical Machines, **IEEE Xplore**, 2010.
26. S. Cafieri, P. Brisset, N. Durand, *A mixed-integer optimization model for Air Traffic Deconfliction*, in **Proceedings of Toulouse Global Optimization workshop (TOGO) 2010**, pp. 27-30, Toulouse, 2010.
27. L. Liberti, S. Cafieri, D. Savourey, *The Reformulation-Optimization Software Engine*, in Komei Fukuda et al. (eds.), International Congress of Mathematical Software (ICMS), **Lecture Notes in Computer Science**, 6327:303-314, 2010.
28. S. Cafieri, P. Hansen, L. Liberti, *Improving heuristics for network modularity maximization using an exact algorithm*, **MatHeuristics 2010**, pp. 130-139, Vienna, 2010.
29. P. Belotti, S. Cafieri, J. Lee, L. Liberti, *On the convergence of feasibility based bounds tightening*, in U. Faigle, R. Schrader, D. Herrmann (eds.), **Proceedings of CTW 2010**, 21-24, Köln 2010.
30. S. Cafieri, P. Hansen, L. Létocart, L. Liberti, F. Messine, *Reduced RLT constraints for polynomial programming*, in P. Bonami, L. Liberti, A. Miller, A. Sartenaer, **Proceedings of European Workshop on MINLP** 2010, Marseille, 2010.
31. S. Cafieri, J. Lee, L. Liberti, *Comparison of convex relaxations of quadrilinear terms*, in C. Ma, L. Yu, D. Zhang, Z. Zhou (eds.), Global Optimization: Theory, Methods and Applications I, **Lecture Notes in Decision Sciences**, 12(B):999-1005, Global-Link Publishers, Hong Kong 2009.
32. S. Cafieri, M. D'Apuzzo, V. De Simone, D. di Serafino, *On the Use of an Approximate Constraint Preconditioner in a Potential Reduction Algorithm for Quadratic Programming*, in V. Cutello, G. Fotia and L. Puccio (eds.), Applied and Industrial Mathematics in Italy II, **Series on Advances in Mathematics for Applied Sciences** Vol. 75, World Scientific, 2007.

- **National Conference publications (refereed)**

33. L. Cellier, S. Cafieri, F. Messine, *Résolution de conflit aérien par contrôle optimal basé sur la régulation en vitesse*, in Proceedings of ROADEF 2012, Angers, France, 2012.
34. S. Cafieri, A. Gondran, S.U. Ngueveu, *Un algorithme mémétique pour construire des trajectoires d'aéronefs robustes aux aléas météorologiques*, in Proceedings of ROADEF 2012, Angers, France, 2012.

35. S. Cafieri, P. Hansen, *Modularity Clustering on Trees*, in Proceedings of ROADEF 2012, Angers, France, 2012.
36. A. Costa, S. Cafieri, P. Hansen, *Reformulation of a locally optimal heuristic for modularity maximization*, in Proceedings of ROADEF 2012, Angers, France, 2012.
37. S. Cafieri, P. Hansen, L. Liberti, *Hierarchical clustering for the identification of communities in networks.* Proceedings of ROADEF 2011, Saint Etienne, France, 2011.
38. D. Aloise, S. Cafieri, G. Caporossi, P. Hansen, L. Liberti, S. Perron, *Algorithms for network modularity maximization*, Proceedings of ROADEF 2010, Toulouse, France, 2010.
39. L. Liberti, S. Cafieri, J. Lee, *Range reduction using fixed points*, Proceedings of ROADEF 2010, Toulouse, France, 2010.
40. S. Cafieri, J. Lee, L. Liberti, *Convex relaxations for quadrilinear terms*, Proceedings of ROADEF 09, Nancy, France, 2009.
41. S. Cafieri, P. Hansen, L. Liberti, *Reformulations between structured global optimization problems and algorithms*, Proceedings of ROADEF 09, Nancy, France, 2009.

- **Technical Reports and submitted papers**

42. S. Cafieri, N. Durand, *Aircraft deconfliction with speed regulation: new models from mixed-integer optimization*, Optimization Online preprint n.3496, June 2012, submitted.
43. P. Belotti, S. Cafieri, J. Lee, L. Liberti, *On feasibility based bounds tightening*, Optimization Online preprint n.3325, January 2012, submitted.
44. S. Cafieri, A. Costa, P. Hansen, *Reformulation of a model for hierarchical divisive graph modularity maximization*, Optimization Online preprint n.3334, December 2011, submitted.
45. A. Mucherino, S. Cafieri, *A New Heuristic for Feature Selection by Consistent Biclustering*, arXiv e-print, arXiv:1003.3279v1, March 2010.

- **Theses**

46. S. Cafieri, *On the application of iterative solvers to KKT systems in Interior Point methods for Large-Scale Quadratic Programming problems*, Ph.D. Thesis, University of Naples "Federico II", 2006.
47. S. Cafieri, *Ottimizzazione quadratica: algoritmi e software per problemi sparsi* (in Italian), Tesi di Laurea, Second University of Naples, 2001.

Conferences and workshops presentations

- EUROmC-VNS - *EURO Mini Conference XXVIII on Variable Neighbourhood Search*, Herceg Novi, Montenegro, Oct. 4-7, 2012.
  "Variable Neighborhood Search for edge-ratio network clustering".

- EURO XXV - *European conference on Operations Research*, Vilnius, Lithuania, July 8-11, 2012.
  "Aircraft conflict avoidance: a mixed-integer nonlinear optimization approach".

- GOW'12 - *Global Optimization Workshop*, Natal, Brazil, June 25-29, 2012.
  "Aircraft conflict avoidance: a mixed-integer nonlinear optimization approach".

- ROADEF'12, Angers, France, avril 2012.
  "Modularity Clustering on Trees".

- AFG'11 - *15th Austrian-French-German conference on Optimization*, Toulouse, France, Sept. 19-23, 2011.
  "Reduced RLT compact relaxations for polynomial programming".

- OR 2011 - *International Conference on Operations Research*, Zurich, Switzerland, Aug 30-Sept 2, 2011.
  "Aircraft deconfliction: a heuristic based on local exact solutions".

- IFORS 2011 - *Conference of the International Federation of Operations Research Societies*, Melbourne, Australia, July 10-15, 2011.
  "Hierarchical Network Clustering".

- ROADEF'11, Saint Etienne, France, March 2011.
  "Hierarchical clustering for the identification of communities in networks".

- ICEM 2010 - *XIX International Conference on Electrical Machines*, Rome, Italy, Sept 6-8, 2010.
  "Discussion about formulations and resolution techniques of electrical machine design problems".

- TOGO10 - *TOulouse Global Optimization workshop*, Toulouse, France, Aug 31-Sept 3, 2010.
  "A mixed-integer optimization model for Air Traffic Deconfliction".

- COSC10 - *International Conference on Optimization, Simulation and Control*, Ulan Baatar, July 25-28, 2010.
  "On the composition of convex envelopes for quadrilinear terms".

- EURO XXIV - *European conference on Operations Research*, Lisbon, Portugal, July 11-14, 2010.
  "Reduced Reformulation-Linearization Technique for Polynomial Programs".

- *MatHeuristics 2010*, Vienna, Austria, June 27-30, 2010.
  "Improving heuristics for network modularity maximization using an exact algorithm".

- EWMINLP - *European Workshop on MINLP*, Marseille, France, March 2010.
  "Reduced RLT constraints for polynomial programming".

- ROADEF10, Toulouse, France, February 2010.
  "Algorithms for network modularity maximization".

- Colloque ANR STIC, Paris, France, January 2010.
  "Automatic Reformulation Search".

- ISMP09 - *The 20th International Symposium of Mathematical Programming*, Chicago, USA, August 2009.
  "Comparing convex relaxations of quadrilinear terms".

- WCGO09 - *1st World Congress on Global Optimization in Engineering and Science*, Hunan, China, June 2009. "Comparison of convex relaxations of quadrilinear terms".

- CIMINLP - *Computational Issues in MINLP*, Bordeaux, France, March 19-20, 2009 (**Invited speaker**).
  "Comparing convex relaxations of quadrilinear terms".

- ROADEF09, Nancy, France, Feb 10-12 2009.
  "Convex relaxations for quadrilinear terms".

- ARS08 - *first ANR Automatic Reformulation Search Project Workshop*, École Polytechnique, Palaiseau, France, Oct 31st 2008.
  "Rose: Reformulation/Optimization Software Engine"; "Convex relaxations for quadrilinear terms".

- *Journeé Optimeo*, Versailles, France, June 11 2008.
  "Linear Algebra issues in Interior Point solvers for Quadratic Programming".

- *IMA Conference on Numerical Linear Algebra and Optimisation*, Birmingham, UK, Sept 13-15 2007.
  "Approximate Constraint Preconditioners for KKT Systems arising in Interior Point Methods".

- *Conference of the Italian MIUR FIRB project "Large Scale Nonlinear Optimization"*, Capri, Italy, Apr 19-20 2007.
  "Sviluppo di software Interior Point per problemi di Ottimizzazione Quadratica".

- SIMAI06 - *8th Congress of the Italian Society for Applied and Industrial Mathematics*, Ragusa, Italy, May 22-26 2006.
  "On the use of Constraint Preconditioners in Potential Reduction methods".

- IFIP TC 7 *Conference on System Modeling and Optimization*, Torino, Italy, July 18-22 2005.
  "A Potential Reduction Solver for Large-Scale Quadratic Programming Problems".

- AIRO04 - *35th Annual Conference of the Italian Operations Research Society*, Lecce, Italy, Sept 7-10 2004.
  "On Linear Algebra Solvers in Potential Reduction Software for Large Scale Quadratic Problems".

- *Large Scale Nonlinear Optimization*, Erice, Italy, June 22-July 1 2004.
  "Linear Algebra Issues in Developing Potential Reduction Software for Large Scale Quadratic Programs".

- *Numerical Methods for Local and Global Optimization: Sequential and Parallel Algorithms*, Cortona, Italy, July 14-20 2003.
  "An Interior Point Solver for Large-Scale Quadratic Programs".

Visiting terms and seminars

- INRA (French Institute for Agricultural Research) Toulouse. March 2012. 1 seminar. Invited by unité de Biométrie et Intelligence Artificielle.
  Seminar: *Clustering dans les réseaux basé sur la maximisation de la modularité.*

- Institut de Mathématiques de Toulouse. May 2011. 1 seminar. Invited by equipe MIP (Mathématiques pour l'Industrie et la Physique).
  Seminar: *Résoudre les conflits aériens par l'optimisation non-linéaire en variables mixtes.*

- GERAD, HEC Montréal. March 2011. Visiting Researcher, 2 weeks. 1 seminar. Invited by P. Hansen.
  Seminar: *Mixed-Integer Optimization for Air Traffic Deconfliction.*

- GERAD, HEC Montréal. July 2009. Visiting Researcher, 1 month. 1 seminar. Invited by P. Hansen.

Seminar: *Convex relaxations in Branch and Bound global optimization methods: quadrilinear terms.*

- LRI, Université Paris XI. 2009, 1 seminar, invited by A. Lisser.
  Seminar: *Convex relaxations in Branch and Bound global optimization methods: quadrilinear terms.*

- Lamsade, Université Paris Dauphine. 2009, 1 seminar, invited by R. Mahjoub.
  Seminar: *Convex relaxations in Branch and Bound global optimization methods: quadrilinear terms.*

- Center for Applied Optimization, University of Florida, Gainesville. March 2007. Visiting Scholar, 3 weeks. 1 seminar. Invited by P.M. Pardalos.
  Seminar: *On the development of Interior Point Software for Quadratic Programming.*

Teaching activity and student supervision

Teaching experience

Note: Some didactic material for the courses can be downloaded from
`http://www.recherche.enac.fr/~cafieri/teaching.html`

- 2011/12, 2012/13: **Differential Computation and Optimization**.
  École Nationale de l'Aviation Civile, 40h lectures.

- 2011/12, 2012/13: **Discrete Optimization**.
  École Nationale Supérieure d'Electrotechnique, Electronique, Informatique, Hydraulique et Télécommunications (ENSEEIHT), 8h lectures + 2h computer labs.

- 2009/10, 2010/11, 2011/12: **Programming and Algorithms**.
  École Nationale de l'Aviation Civile, 50h computer labs (TD).

- 2010/11: **Constraint Programming**.
  École Nationale de l'Aviation Civile, 20h computer labs (TD).

- 2009/10: **Introduction to C++**.
  École Polytechnique, 6h computer labs (TD).

- 2008/09, 2009/10: **Operations Research: Modelling and Software**.
  École Polytechnique, 2h lectures, 16h computer labs.

- 2005/06: **Informatics**.
  Second University of Naples, 40h lectures, 20h computer labs.
- 2002/03 to 2004/05: **Informatics**.
  Second University of Naples, 10h lectures, 50h computer labs.
- 2005/06: **Introduction to Numerical Methods for Optimization**.
  Second University of Naples, 8h lectures.
- 2002/03 to 2005/06: **Parallel Computing**.
  Second University of Naples, 10h lectures, 10h computer labs.
- 2002/03 to 2005/06: **Numerical Computations**.
  Second University of Naples, 10h lectures, 20h computer labs.
- 2002/03 to 2005/06: **Introduction to Programming**.
  Second University of Naples, 10h lectures, 20h computer labs.
- 2002/03 to 2005/06: **Introduction to Informatics**.
  Second University of Naples, 10h lectures, 20h computer labs.
- 2003/04 to 2005/2006: **Introduction to HTML**.
  Second University of Naples, 8h lectures.
- 2002/03, 2003/04: **Computational Mathematics**.
  Second University of Naples, 10h lectures, 10h computer labs.
- 2002/03, 2003/04: **Programming and Numerical Computations**.
  Second University of Naples, 6h lectures, 10h computer labs.
- 2002/03 to 2004/05: **Numerical Analysis**.
  Second University of Naples, 6h lectures, 10h computer labs.

Supervisioning and tutoring

- Since October 2011: **Supervision of the Ph.D. thesis** of Loïc Cellier.
  Topic: Deterministic Global Optimization applied to Air Traffic Management problems (co-supervision with Frédéric Messine).
- 2011: Supervision of 2 mini-projects on development of C and Caml code by 4 students (each project is developed by 2 students) 1st year engineers ENAC.
- 2010: Supervision of a mini-project (implementation in Java) by 4 students second year engineers ENAC. Topic: Analysis of air traffic network.

- 2010: Supervision of 4 mini-projects on development of C code by 8 students (each project is developed by 2 students) 1st year engineers ENAC.
- 2006: Co-supervision (with M. D'Apuzzo) of the M.Sc. thesis of A. Aldanese. Second University of Naples. Topic: Software for linear programming problems.
- 2006: Co-supervision (with D. di Serafino) of the M.Sc. thesis of E. Giannelevigna. Second University of Naples. Topic: A Potential Reduction method for quadratic optimization.
- 2004: Co-supervision (with M. D'Apuzzo) of the M.Sc. thesis of L. Minicucci. Second University of Naples. Topic: Interior Point methods for quadratic optimization.
- Personal tutor for students of Second University of Naples every academic year from 2003/2004 to 2005/2006.

Software

- PRQP (*Potential Reduction for Quadratic Programming*)
  solves convex quadratic problems with linear constraints
  - primal-dual infeasible PR method, feasible whenever possible;
  - different solvers for the KKT system: direct, CG, SQMR;
  - exact and reused constraint preconditioner, limited-memory ICF for bound constrained problems;
  - MA27 routine by the HSL library for sparse LDL^T computation;
  - custom SQMR and sparse matrix-vector products;
  - AMPL and SIF interfaces;
  - Fortran77, C drivers.
- PR-BCQP
  solves convex quadratic problems with only bounds on the variables; it is currently part of PRQP, but can be used as stand-alone software.
- ROSE (*Reformulation/Optimization Software Engine*)
  software framework for automatic reformulations of mathematical programming problems. Co-developer.

  - reformulators able to automatically provide convex relaxations of non-convex nonlinear problems.
  - data format translators.

- Contribution to COUENNE, in COIN-OR.

Selected Publications

[Belotti et al., 2012a] Belotti, P., Cafieri, S., Lee, J., and Liberti, L. (2012a). On feasibility based bounds tightening. *Optimization Online preprint n.3325*. submitted.

[Belotti et al., 2012b] Belotti, P., Cafieri, S., Lee, J., Liberti, L., and Miller, A. (2012b). On the composition of convex envelopes for quadrilinear terms. In Chinchuluun, A., Pardalos, P., Enkhbat, R., and Pistikopoulos, E., editors, *Proceedings of the International Conference on Optimization, Simulation and Control*, Springer Optimization and its Applications. Springer. (in press).

[Cafieri et al., 2012a] Cafieri, S., Caporossi, G., Hansen, P., Perron, S., and Costa, A. (2012a). Finding communities in networks in the strong and almost-strong sense. *Physical Review E*, 85(4):046113.

[Cafieri et al., 2012b] Cafieri, S., Costa, A., and Hansen, P. (2012b). Reformulation of a model for hierarchical divisive graph modularity maximization. *Annals of Operations Research*. DOI 10.1007/s10479-012-1286-z (in press).

[Cafieri and Durand, 2012] Cafieri, S. and Durand, N. (2012). Aircraft deconfliction with speed regulation: new models from mixed-integer optimization. *Optimization Online preprint n.3496*. submitted.

[Cafieri et al., 2012c] Cafieri, S., Hansen, P., Létocart, L., Liberti, L., and Messine, F. (2012c). Compact relaxations for polynomial programming problems. In Klasing, R., editor, *Experimental Algorithms (Proceedings of SEA)*, volume 7276 of *Lecture Notes in Computer Science*, pages 75–86, Berlin. Springer.

[Cafieri et al., 2012d] Cafieri, S., Hansen, P., and Liberti, L. (2012d). Improving heuristics for network modularity maximization using an exact algorithm. *Discrete Applied Mathematics*. DOI 10.1016/j.dam.2012.03.030 (in press).

[Cafieri et al., 2012e] Cafieri, S., Liberti, L., Messine, F., and Nogaréde, B. (2012e). Optimal design of electrical machines: Mathematical programming formulations. *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*. (in press).

[Cellier et al., 2012] Cellier, L., Cafieri, S., and Messine, F. (2012). Hybridizing direct and indirect optimal control approaches for aircraft conflict avoidance. In *ADVCOMP 2012: The Sixth International Conference on Advanced Engineering Computing and Applications in Sciences*, pages 42–45, Barcelone. ISBN: 978-1-61208-237-0.

[Cafieri et al., 2011] Cafieri, S., Hansen, P., and Liberti, L. (2011). Locally optimal heuristic for modularity maximization of networks. *Physical Review E*, 83(5):056105.

[Aloise et al., 2010] Aloise, D., Cafieri, S., Caporossi, G., Hansen, P., Liberti, L., and Perron, S. (2010). Column generation algorithms for exact modularity maximization in networks. *Physical Review E*, 82(4):046112.

[Cafieri et al., 2010a] Cafieri, S., Hansen, P., and Liberti, L. (2010a). Edge ratio and community structure in networks. *Physical Review E*, 81(2):026105.

[Cafieri et al., 2010b] Cafieri, S., Hansen, P., and Liberti, L. (2010b). Loops and multiple edges in modularity maximization of networks. *Physical Review E*, 81(4):046102.

[Cafieri et al., 2010c] Cafieri, S., Lee, J., and Liberti, L. (2010c). On convex relaxations of quadrilinear terms. *Journal of Global Optimization*, 47:661–685.

[Liberti et al., 2010] Liberti, L., Cafieri, S., and Savourey, D. (2010). The reformulation-optimization software engine. In Fukuda, K., editor, *Proceedings of ICMS10*, volume 6327 of *Lecture Notes in Computer Science*, pages 303–314. Springer.

[Liberti et al., 2009] Liberti, L., Cafieri, S., and Tarissan, F. (2009). Reformulations in mathematical programming: a computational approach. In Abraham, A., Hassanien, A.-E., Siarry, P., and Engelbrecht, A., editors, *Foundations of Computational Intelligence (Global Optimization: Theoretical Foundations and Applications)*, volume 203 of *Studies in Computational Intelligence*, pages 153–234. Springer.

[Cafieri et al., 2007a] Cafieri, S., D'Apuzzo, M., De-Simone, V., and di Serafino, D. (2007a). On the iterative solution of KKT systems in potential reduction software for large-scale quadratic problems. *Computational Optimization and Applications*, 38(1):27–45.

[Cafieri et al., 2007b] Cafieri, S., D'Apuzzo, M., De-Simone, V., and di Serafino, D. (2007b). On the use of an approximate constraint preconditioner in a potential reduction algorithm for quadratic programming. In Cutello, V., Fotia, G., and Puccio, L., editors, *Applied and Industrial Mathematics in Italy II*, Series on Advances in Mathematics for Applied Sciences Vol. 75.

[Cafieri et al., 2007c] Cafieri, S., D'Apuzzo, M., De-Simone, V., and di Serafino, D. (2007c). Stopping criteria for inner iterations in inexact potential reduction methods: a computational study. *Computational Optimization and Applications*, 36(2):165–193.

[Cafieri et al., 2007d] Cafieri, S., D'Apuzzo, M., De-Simone, V., di Serafino, D., and Toraldo, G. (2007d). Convergence analysis of an inexact potential reduction method for convex quadratic programming. *Journal of Optimization Theory and Applications*, 135(1):355–366.

[Cafieri et al., 2006] Cafieri, S., D'Apuzzo, M., Marino, M., Mucherino, A., and Toraldo, G. (2006). Interior-point solver for large-scale quadratic programming problems with bound constraints. *Journal of Optimization Theory and Applications*, 129(1):55–75.

Appendix A

S. Cafieri, J. Lee, L. Liberti,

On convex relaxations of quadrilinear terms,

Journal of Global Optimization,

47:661–685, 2010.

Appendix B

S. Cafieri, P. Hansen, L. Létocart, L. Liberti, F. Messine,
Compact relaxations for polynomial programming problems,
in R. Klasing (eds.), Experimental Algorithms
(Proceedings of SEA 2012),
Lecture Notes in Computer Science 7276:75-86,
Springer, Berlin, 2012.

Appendix C

L. Liberti, S. Cafieri, F. Tarissan,
Reformulations in Mathematical Programming: a Computational Approach,
in A. Abraham, A.-E. Hassanien, P. Siarry, and A. Engelbrecht (eds.),
Foundations of Computational Intelligence Vol. 3 (Global Optimization: Theoretical Foundations and Applications),
Series: Studies in Computational Intelligence, 203:153-234,
Springer, Berlin, 2009.

Appendix D

S. Cafieri, N. Durand,

Aircraft deconfliction with speed regulation: new models from mixed-integer optimization,

Optimization Online preprint n.3496,

Jun 2012, submitted.

Appendix E

S. Cafieri, L. Liberti, F. Messine, B. Nogarede,
Optimal Design of Electrical Machines: Mathematical Programming Formulations,
COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering,
in press, 2012.

Appendix F

S. Cafieri, M. D'Apuzzo, V. De Simone, D. di Serafino, G. Toraldo,
Convergence Analysis of an Inexact Potential Reduction Method for Convex Quadratic Programming,
Journal of Optimization Theory and Applications,
135: 355–366, 2007.

Appendix G

S. Cafieri, M. D'Apuzzo, V. De Simone, D. di Serafino,
On the Iterative Solution of KKT Systems in Potential Reduction Software for Large Scale Quadratic Problems,
Computational Optimization and Applications,
special issue on High Performance Algorithms and Software for Nonlinear Optimization, A. Murli and G. Toraldo (eds.),
38: 27–45, 2007.

Appendix H

S. Cafieri, M. D'Apuzzo, V. De Simone, D. di Serafino,
Stopping criteria for inner iterations in inexact Potential Reduction methods: a computational study,
Computational Optimization and Applications,
special issue on Linear Algebra issues arising in Interior Point methods, J. Gondzio and G. Toraldo (eds.),
36 (2): 165-193, 2007.

Appendix I

S. Cafieri, M. D'Apuzzo, V. De Simone, D. di Serafino,
On the Use of an Approximate Constraint Preconditioner in a Potential Reduction Algorithm for Quadratic Programming,
in V. Cutello, G. Fotia and L. Puccio (eds.),
Applied and Industrial Mathematics in Italy II,
Series on Advances in Mathematics for Applied Sciences
Vol. 75, World Scientific, 2007.

Appendix J

L. Cellier, S. Cafieri, F. Messine,
Hybridizing Direct and Indirect Optimal Control Approaches for Aircraft Conflict Avoidance,
in Proceedings of *ADVCOMP 2012: The Sixth International Conference on Advanced Engineering Computing and Applications in Sciences*, pp. 42-45, Barcelone, 2012.

Appendix K

S. Cafieri, P. Hansen, L. Liberti,
Loops and multiple edges in modularity maximization of networks,
Physical Review E,
81(4):046102, 2010.

Appendix L

D. Aloise, S. Cafieri, G. Caporossi, P. Hansen, L. Liberti, S. Perron,
Column generation algorithms for exact modularity maximization in networks,
Physical Review E,
82(4):046112, 2010.

Appendix M

S. Cafieri, P. Hansen, L. Liberti,
Locally optimal heuristic for modularity maximization of networks,
Physical Review E,
83(5):056105, 2011.

Appendix N

S. Cafieri, P. Hansen, L. Liberti,
Improving heuristics for network modularity maximization using an exact algorithm,
Discrete Applied Mathematics,
DOI 10.1016/j.dam.2012.03.030,
in press, 2012.

Appendix O

S. Cafieri, P. Hansen, L. Liberti,

Edge ratio and community structure in networks,

Physical Review E,

81(2):026105, 2010.

Appendix P

S. Cafieri, G. Caporossi, P. Hansen, S. Perron, A. Costa,
Finding communities in networks in the strong and almost-strong sense,
Physical Review E,
85(4):046113, 2012.

Appendix Q

S. Cafieri, A. Costa, P. Hansen,

Reformulation of a model for hierarchical divisive graph modularity maximization,

Optimization Online preprint n.3334,

Dec. 2011 (in revision).