XIII Global Optimization Workshop

GOW'16

PROCEEDINGS OF THE

XIII GLOBAL OPTIMIZATION WORKSHOP

GOW'16

4-8 September 2016

University of Minho, Braga, Portugal

ISBN 978-989-20-6764-3

PROCEEDINGS OF THE XIII GLOBAL OPTIMIZATION WORKSHOP

GOW'16

4-8 September 2016

Edited by

Ana Maria A. C. Rocha
University of Minho/Algoritmi Research Centre

M. Fernanda P. Costa
University of Minho/Centre of Mathematics

Edite M. G. P. Fernandes
Algoritmi Research Centre

ISBN: 978-989-20-6764-3
University of Minho, Braga, Portugal

Preface

Past Global Optimization Workshops have been held in Sopron (1985 and 1990), Szeged (WGO, 1995), Florence (GO'99, 1999), Hanmer Springs (Let's GO, 2001), Santorini (Frontiers in GO, 2003), San José (Go'05, 2005), Mykonos (AGO'07, 2007), Skukuza (SAGO'08, 2008), Toulouse (TOGO'10, 2010), Natal (NAGO'12, 2012) and Málaga (MAGO'14, 2014) with the aim of stimulating discussion between senior and junior researchers on the topic of Global Optimization.

In 2016, the XIII Global Optimization Workshop (GOW'16) takes place in Braga and is organized by three researchers from the University of Minho. Two of them belong to the Systems Engineering and Operational Research Group from the Algoritmi Research Centre and the other to the Statistics, Applied Probability and Operational Research Group from the Centre of Mathematics. The event received more than 50 submissions from 15 countries from Europe, South America and North America.

We want to express our gratitude to the invited speaker Panos Pardalos for accepting the invitation and sharing his expertise, helping us to meet the workshop objectives. GOW'16 would not have been possible without the valuable contribution from the authors and the International Scientific Committee members. We thank you all.

This proceedings book intends to present an overview of the topics that will be addressed in the workshop with the goal of contributing to interesting and fruitful discussions between the authors and participants. After the event, high quality papers can be submitted to a special issue of the Journal of Global Optimization dedicated to the workshop.

Ana Maria A. C. Rocha
M. Fernanda P. Costa
Edite M. G. P. Fernandes

GOW'16 Organizers

Scientific Committee:

- Panos M. Pardalos, University of Florida (USA)
- Christodoulos A. Floudas, Texas A&M University (USA)
- Boris Mordukhovich, Wayne State University (USA)
- Nikolaos V. Sahinidis, Carnegie Mellon University (USA)
- David Gao, Federation University Australia (Australia)
- Ruey-Lin Sheu, National Cheng-Kung University (Taiwan)
- Sergiy Butenko, Texas A&M University (USA)
- Ana Maria A. C. Rocha, University of Minho (Portugal)
- Chrysanthos E. Gounaris, Carnegie Mellon University (USA)
- Ding-zhu Du, University of Texas (USA)
- Nicolas Hadjisavvas, King Fahd University of Petroleum and Minerals (Saudi Arabia)
- Fülöp János, Hungarian Academy of Sciences (Hungary)
- Inmaculada G. Fernandez, University of Malaga (Spain)
- Alexander Mitsos, RWTH Aachen University (Germany)
- Dumitru Motreanu, University of Perpignan (France)
- Steffen Rebennack, Colorado School of Mines (USA)
- Mauricio G. C. Resende, AT&T Labs Research (USA)
- Leo Liberti, IBM T.J. Watson Research Center (USA)
- Yaroslav D. Sergeyev, University of Calabria (Italy)
- Fabio Schoen, University of Florence (Italy)
- Kaisa Miettinen, University of JyvÃd'skylÃd' (Finland)
- Immanuel Bomze, University of Vienna (Austria)
- M. Fernanda P. Costa, University of Minho (Portugal)
- Angelo Lucia, University of Rhode Island (USA)
- Stratos Pistikopoulos, Texas A&M University (USA)
- Antanas Zilinskas, Vilnius University (Lithuania)
- Leonidas Pitsoulis, Aristotle University of Thessaloniki (Greece)
- Julius Zilinskas, Vilnius University (Lithuania)
- Edite M.G.P. Fernandes, University of Minho (Portugal)
- Igor Konnov, Kazan Federal University (Russia)
- Hoang Tuy, Institute of Mathematics (Vietnam)
- Paul I. Barton, Massachusetts Institute of Technology (USA)
- Michel Théra, University of Limoges (France)
- Erhun Kundakcioglu, Ozyegin University (Turkey)
- Simge Kucukyavuz, Ohio State University (USA)
- Tapio Westerlund, Åbo Akademi University (Finland)
- W. Art Chaovalitwongse, University of Washington (USA)
- Juan Enrique Martínez-Legaz, Universitat Autónoma de Barcelona (Spain)
- Marco Locatelli, University of Parma (Italy)
- Gianni Di Pillo, Sapienza University of Rome (Italy)
- Aris Daniilidis, University of Chile (Chile)
- Sofia Giuffrè, Mediterranea University of Reggio Calabria (Italy)
- Antonino Maugeri, University of Catania (Italy)
- Joaquim Júdice, University of Coimbra (Portugal)
- Anthony Man-Cho So, Chinese University of Hong Kong (China)
- C. Audet, Polytechnique Montreal (Canada)
- Janos D. Pintér, Saint Mary's University (Canada)
- Francesco Zirilli, Sapienza University of Rome (Italy)
- Adil Bagirov, University of Ballarat (Australia)
- Eligius M. T. Hendrix, University of Malaga (Spain)
- J. Valério de Carvalho, University of Minho (Portugal)

- Boris Goldengorin, Ohio University (USA)
- Mirjam Dür, University of Trier (Germany)
- Montaz Ali, University of the Witwatersrand (South Africa)
- Dmitri E. Kvasov, University of Calabria (Italy)

Organizing Committee:

- Ana Maria A. C. Rocha
- M. Fernanda P. Costa
- Edite M. G. P. Fernandes

Sponsors:

- School of Engineering
- Department of Production and Systems
- Algoritmi Research Centre
- School of Sciences
- Department of Mathematics and Applications
- Centre of Mathematics
- University of Minho
- Câmara Municipal de Braga
- Libware

Contents

INVITED TALK

Proceedings of GOW'16, pp. 3 – 5.

On the Passage from Local to Global in Optimization: New Challenges in Theory and Practice

Panos M. Pardalos [1]

[1] *Center for Applied Optimization, ISE Department,*
303 Weil Hall, University of Florida, Gainesville, FL 32611,
http://www.ise.ufl.edu/pardalos, pardalos@ise.ufl.edu

Abstract Large scale problems in the design of networks and energy systems, the biomedical field, finance, and engineering are modeled as optimization problems. Humans and nature are constantly optimizing to minimize costs or maximize profits, to maximize the flow in a network, or to minimize the probability of a blackout in the smart grid.

Due to new algorithmic developments and the computational power of computers, optimization algorithms have been used to solve problems in a wide spectrum of applications in science and engineering. In this talk I am going to address new challenges in the theory and practice of optimization. First, we have to reflect back a few decades to see what has been achieved and then address the new research challenges and directions.

Keywords: Global optimization,Local optimization, Complexity issues, Challenging problems

1. Global Optimization Problem

$$f^* = f(x^*) = global\ min_{x \in D} f(x)\ (or\ max_{x \in D} f(x)) \tag{1}$$

For these global optimization problem we are going to address the following tasks:

- Compute a globally optimal solution.
- Compute "good" locally optimal solutions (or feasible points that satisfy the optimality conditions).
- Compute "better" solutions than "known" solutions.
- Check feasibility of the constraints.

For the first case we will discuss complexity issues and the need of certificates of optimality. In addition, we are going to discuss classes of problems where only globally optimal solutions are needed (or make sense). Furthermore, we are going to discuss recent progress and open questions regarding exact algorithms.

The next two cases are of great practical significance. We can use approximation algorithms or heuristics. We are going to discuss issues regarding the evaluation of the performance of heuristics and the complexity of approximation algorithms.

For the last case, checking feasibility of the constraints is an important problem. In the case of infeasibility, we may need to make a minimum data perturbation so that the problem becomes feasible.

2. Why are Optimization Problems Difficult?

Different complexity theories have tried to classify problems as easy or hard. The main focus of computational complexity is to analyze the intrinsic difficulty of optimization problems and

to decide which of these problems are likely to be tractable. The pursuit of developing efficient algorithms also leads to elegant general approaches for solving optimization problems, and reveals surprising connections among problems and their solutions.

What do we know about the phase transition from easy to hard problems? In addition, what are the classes of problems between the easy and hard instances? The general optimization problem is **NP-hard**. Furthermore, checking if a feasible point is a local optimum is also an **NP-hard** problem. Is it the "hardness of checking convexity", or the "exponential number of local mimina" that make a global optimization problem difficult to solve? We are going to discuss recent attempts to answer some of these questions.

3. Classes of Global Optimization Problems

There is a huge literature that deals with many important classes of global optimization problems such as multi-level (or hierarchical optimization), problems with equilibrium constraints, Lipschitz optimization, DC (different of convex functions) and DM (difference of monotonically increasing functions) optimization.

In addition, "black box" optimization is an important class of hard optimization problems that appear very often in practice. Black box optimization is connected with machine learning and we are going to discuss some challenging issues regarding black box optimization.

4. Global Optimization Software

Some of the first optimization books (for continuous and discrete optimization) have been written by chemical engineers because of the significance of optimization in solving problems in the oil industry. There is a state of the art software for problems that can be expressed or approximated by mixed linear zero-one models. Such approaches are in particularly very practical for separable optimization.

In addition, there are several optimization packages that are very efficient for problems with a special structure. However, there is still a need for general purpose optimization software. We are going to address several issues regarding testing, automatic parameter identification, and evaluation of global optimization software.

5. Research Directions and New Challenges in Optimization

In many practical cases, uncertainty of the data is a key problem in optimization. Over the years parametric optimization and stochastic programming approaches have been used to address issues of uncertainty. In the last few years, robust optimization has been a very promising alternative. In particular, in data sciences robust optimization algorithms can have a great impact in several applications.

Optimization with massive data sets remains a very challenging area of research. For example, new algorithms and new computer environments are needed to solve optimization problems with massive networks External memory algorithms and new data structures have been developed only for a few such optimization problems.

"Multi-objective optimization" is the final frontier. The search for a Pareto optimal solution remains a challenge, although several sophisticated heuristics have been developed in the last few decades. In nature, "cooperative systems" manage to optimize in certain ways. We may need the synergy of the two fields to lead us to new paths for developing novel approaches in multi-objective optimization.

References

[1] P.M. Pardalos and J.B. Rosen (Eds.) *Constrained Global Optimization: Algorithms and Applications*. Lecture Notes in Computer Science 268, Springer-Verlag, Berlin, 1987.

[2] P.M. Pardalos. *Continuous Approaches to Discrete Optimization Problems*. In: Pillo, G.D. and Giannessi, F. (Eds.) Nonlinear Optimization and Applications, Plenum, pp. 313-328, 1996.

[3] D.Z. Du and P.M. Pardalos. *Global Minimax Approaches for Solving Discrete Problems*. Lecture Notes in Economics and Mathematical Systems 452:34–48, 1997.

[4] J. Abello, P.M. Pardalos and M.G.C. Resende. *On maximum clique problems in very large graphs*. In DIMACS Vol. 50, American Mathematical Society, pp. 119-130, 1999.

[5] J. Abello, P.M. Pardalos and M.G.C. Resende (Eds.), *Handbook of Massive Data Sets*, Kluwer Academic Publishers, 2000.

[6] P.M. Pardalos, D. Shalloway and G. Xue (Eds.) *Global Minimization of Nonconvex Energy Functions: Molecular Conformation and Protein Folding*. DIMACS Series, American Mathematical Society, 1996.

[7] R. Horst and P.M. Pardalos (Eds.), *Handbook of Global Optimization*, Kluwer Academic Publishers, 1995.

[8] R. Horst, P.M. Pardalos and N.V. Thoai. *Introduction to Global Optimization*, Kluwer Academic Publishers, 1995 (Second Edition (2000)).

[9] P.M. Pardalos (Ed.) *Approximation and Complexity in Numerical Optimization*, Nonconvex Optimization and Its Applications Series, Kluwer Academic Publishers, 2000.

[10] B. Goldengorin, D. Krushinsky and P.M. Pardalos. *Cell Formation in Industrial Engineering*. Springer Optimization and Its Applications, Springer, 2013.

[11] B. Goldengorin and P.M. Pardalos. *Data Correcting Approaches in Combinatorial Optimization*. Springer, New York, 2012.

[12] A. Sorokin and P.M. Pardalos. *Dynamics of Information Systems: Algorithmic Approaches.* Springer, 2013.

[13] P.M. Pardalos, D.-Z Du and R. Graham (Eds.) *Handbook of Combinatorial Optimization*. 5 volumes, 2-nd edition, Springer, 2013.

[14] S.D. Eksioglu, S. Rebennack and P.M. Pardalos (Eds.) *Handbook of Bioenergy: Bioenergy Supply Chain - Models and Applications*. Springer, 2015.

[15] P.M. Pardalos and E. Romeijn (Eds.) *Handbook of Global Optimization, Volume 2: Heuristic Approaches.* Kluwer Academic Publishers, 2002.

[16] A. Chinchuluun, A. Migdalas, P.M. Pardalos and L. Pitsoulis. *Pareto Optimality, Game Theory and Equilibria.*, Springer, 2008.

EXTENDED ABSTRACTS

Proceedings of GOW'16, pp. 9 – 12.

The Cluster Problem in Constrained Global Optimization*

Rohit Kannan and Paul I. Barton

Process Systems Engineering Laboratory,
Massachusetts Institute of Technology, Cambridge, MA, USA {rohitk, pib}@mit.edu

Abstract One of the key issues in continuous deterministic global optimization is the cluster problem wherein a large number of boxes may be visited in the neighborhood of a global minimizer [3, 4, 5, 6]. It is well-known in the unconstrained global optimization literature that, in the worst case, at least second-order Hausdorff convergence of lower bounding schemes is necessary to avoid the cluster problem when the minimizer sits at a point of differentiability of the objective function. In this work, a definition of convergence order for lower bounding schemes for constrained problems is proposed. Based on the proposed definition, the cluster problem for constrained global optimization is analyzed and sufficient conditions for first-order convergent lower bounding schemes to eliminate the cluster problem are provided.

Keywords: Cluster problem, Convergence-order, Branch-and-bound, Constrained global optimization

1. Introduction

Consider the problem

$$\begin{aligned} \underset{x\in X}{min}\ & f(\mathbf{x}) \\ \text{s.t. } & \mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \\ & \mathbf{h}(\mathbf{x}) = \mathbf{0}, \end{aligned} \tag{P}$$

where $X \subset \mathbb{R}^{n_x}$ is a nonempty open bounded convex set, and the functions $f : X \to \mathbb{R}$, $\mathbf{g} : X \to \mathbb{R}^{m_I}$, and $\mathbf{h} : X \to \mathbb{R}^{m_E}$ are continuous on X. We make the following assumptions.

Assumption 1. *The functions f, $\mathbf{g}$, and $\mathbf{h}$ are twice continuously differentiable on X, and the constraints $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$ and $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ define a nonempty compact set contained in X.*

Assumption 2. *Let $\mathbf{x}^* \in X$ be a global minimum for Problem* (P)*, and assume that the branch-and-bound algorithm has found the upper bound $UBD = f(\mathbf{x}^*)$ sufficiently early on. Let $\varepsilon > 0$ be the termination tolerance for the branch-and-bound algorithm, and suppose the algorithm fathoms node k when $UBD - LBD_k \leq \varepsilon$, where LBD_k is the lower bound on node k.*

Denote by $\mathbb{I}Z$ the set of nonempty bounded interval subsets of $Z \subset \mathbb{R}^n$, by $\bar{\mathbf{f}}(Z)$ the image of $Z \subset X$ under the function $\mathbf{f} : X \to \mathbb{R}^m$, and by $\mathcal{N}_\alpha^p(\mathbf{x})$ the set $\{\mathbf{z} \in X : \|\mathbf{z} - \mathbf{x}\|_p < \alpha\}$ corresponding to the α-neighborhood of $\mathbf{x}$ in X with respect to the p-norm.

Definition 3 (Width of an Interval)**.** *Let $Z = [z_1^{\mathrm{L}}, z_1^{\mathrm{U}}] \times \cdots \times [z_n^{\mathrm{L}}, z_n^{\mathrm{U}}]$ be an element of $\mathbb{IR}^n$. The width of Z, denoted by $w(Z)$, is given by $w(Z) = \max_{i=1,\cdots,n} (z_i^{\mathrm{U}} - z_i^{\mathrm{L}})$.*

Definition 4 (Distance Between Two Sets)**.** *Let $Y, Z \subset \mathbb{R}^n$. The distance between Y and Z, denoted by $d(Y, Z)$, is defined as $d(Y, Z) = \inf_{(\mathbf{y},\mathbf{z})\in Y\times Z} \|\mathbf{y} - \mathbf{z}\|_2$.*

The reader is directed to the work of Mitsos and coworkers [1, 2] for the definition of schemes of relaxations, and the notions of Hausdorff and pointwise convergence of such

*The authors gratefully acknowledge financial support from BP. This work was conducted as a part of the BP-MIT conversion research program.

schemes. The following definition extends the notion of convergence order [1, 2, 6] to lower bounding schemes for constrained problems.

Definition 5 (Convergence Order of a Lower Bounding Scheme)**.** *Consider Problem* (P) *satisfying Assumption 1. For any $Z \in \mathbb{I}X$, let $\mathcal{F}(Z) = \{\mathbf{x} \in Z : \mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \mathbf{h}(\mathbf{x}) = \mathbf{0}\}$ denote the feasible set of Problem* (P) *with $\mathbf{x}$ restricted to Z. Let $(f_Z^{\text{cv}})_{Z\in\mathbb{I}X}$ and $(\mathbf{g}_Z^{\text{cv}})_{Z\in\mathbb{I}X}$ denote continuous schemes of convex relaxations of f and $\mathbf{g}$, respectively, in X, and let $(\mathbf{h}_Z^{\text{cv}}, \mathbf{h}_Z^{\text{cc}})_{Z\in\mathbb{I}X}$ denote a continuous scheme of relaxations of $\mathbf{h}$ in X. For any $Z \in \mathbb{I}X$, let $\mathcal{F}^{\text{CV}}(Z) = \left\{\mathbf{x} \in Z : \mathbf{g}_Z^{\text{CV}}(\mathbf{x}) \leq \mathbf{0}, \mathbf{h}_Z^{\text{CV}}(\mathbf{x}) \leq \mathbf{0}, \mathbf{h}_Z^{\text{CC}}(\mathbf{x}) \geq \mathbf{0}\right\}$ denote the feasible set of the convex relaxation-based lower bounding scheme. The convex relaxation-based lower bounding scheme is said to have convergence of order $\beta > 0$ at*

1. *a feasible point $\mathbf{x} \in X$ if there exists $\tau \geq 0$ such that for every $Z \in \mathbb{I}X$ with $\mathbf{x} \in Z$,*

$$\min_{\mathbf{z}\in\mathcal{F}(Z)} f(\mathbf{z}) - \min_{\mathbf{z}\in\mathcal{F}^{\text{CV}}(Z)} f_Z^{\text{CV}}(\mathbf{z}) \leq \tau w(Z)^{\beta}.$$

2. *an infeasible point $\mathbf{x} \in X$ if there exists $\bar{\tau} \geq 0$ such that for every $Z \in \mathbb{I}X$ with $\mathbf{x} \in Z$,*

$$d(\bar{\mathbf{g}}(Z), \mathbb{R}_-^{m_I}) - d(\bar{\mathbf{g}}_Z^{\text{cv}}(Z), \mathbb{R}_-^{m_I}) \leq \bar{\tau} w(Z)^{\beta}, \text{ and}$$
$$d(\bar{\mathbf{h}}(Z), \{\mathbf{0}\}) - d(\mathcal{I}_E(Z), \{\mathbf{0}\}) \leq \bar{\tau} w(Z)^{\beta},$$

where $\mathcal{I}_E(Z)$ is defined by

$$(\mathcal{I}_E(Z))_{Z\in\mathbb{I}X} := (\{\mathsf{w} \in \mathbb{R}^{m_E} : \mathbf{h}_Z^{\text{cv}}(\mathbf{x}) \leq \mathsf{w} \leq \mathbf{h}_Z^{\text{cc}}(\mathbf{x}) \text{ for some } \mathbf{x} \in Z\})_{Z\in\mathbb{I}X}.$$

The scheme of lower bounding problems is said to have convergence of order $\beta > 0$ on X if it has convergence of order (at least) β at each $\mathbf{x} \in X$, with the constants τ and $\bar{\tau}$ independent of $\mathbf{x}$.

Suppose the convex relaxation-based lower bounding scheme has convergence of order $\beta^* > 0$ on $\mathcal{F}(X)$ with prefactor $\tau^* > 0$, and convergence of order $\beta^I > 0$ at infeasible points with prefactor $\tau^I > 0$. Furthermore, suppose the scheme $(f_Z^{\text{cv}})_{Z\in\mathbb{I}X}$ has convergence of order $\beta^f > 0$ at infeasible points with prefactor $\tau^f > 0$. Define δ, tolerances ε^I and ε^f such that $\left(\frac{\varepsilon^I}{\tau^I}\right)^{\frac{1}{\beta^I}} = \left(\frac{\varepsilon^f}{\tau^f}\right)^{\frac{1}{\beta^f}} = \left(\frac{\varepsilon}{\tau^*}\right)^{\frac{1}{\beta^*}} = \delta$, and consider the following partition of X:

$$X_1 = \left\{\mathbf{x} \in X : \max\left\{d(\{\mathbf{g}(\mathbf{x})\}, \mathbb{R}_-^{m_I}), d(\{\mathbf{h}(\mathbf{x})\}, \{\mathbf{0}\})\right\} > \varepsilon^I\right\},$$
$$X_2 = \left\{\mathbf{x} \in X : \max\left\{d(\{\mathbf{g}(\mathbf{x})\}, \mathbb{R}_-^{m_I}), d(\{\mathbf{h}(\mathbf{x})\}, \{\mathbf{0}\})\right\} \in (0, \varepsilon^I] \text{ and } f(\mathbf{x}) - f(\mathbf{x}^*) > \varepsilon^f\right\},$$
$$X_3 = \left\{\mathbf{x} \in X : \max\left\{d(\{\mathbf{g}(\mathbf{x})\}, \mathbb{R}_-^{m_I}), d(\{\mathbf{h}(\mathbf{x})\}, \{\mathbf{0}\})\right\} \in (0, \varepsilon^I] \text{ and } f(\mathbf{x}) - f(\mathbf{x}^*) \leq \varepsilon^f\right\},$$
$$X_4 = \left\{\mathbf{x} \in X : \max\left\{d(\{\mathbf{g}(\mathbf{x})\}, \mathbb{R}_-^{m_I}), d(\{\mathbf{h}(\mathbf{x})\}, \{\mathbf{0}\})\right\} = 0 \text{ and } f(\mathbf{x}) - f(\mathbf{x}^*) > \varepsilon\right\},$$
$$X_5 = \left\{\mathbf{x} \in X : \max\left\{d(\{\mathbf{g}(\mathbf{x})\}, \mathbb{R}_-^{m_I}), d(\{\mathbf{h}(\mathbf{x})\}, \{\mathbf{0}\})\right\} = 0 \text{ and } f(\mathbf{x}) - f(\mathbf{x}^*) \leq \varepsilon\right\}.$$

By virtue of the definitions of δ, ε^I, and ε^f, nodes with domains $\bar{X}_1 \in \mathbb{I}X_1$, $\bar{X}_2 \in \mathbb{I}X_2$, and $\bar{X}_4 \in \mathbb{I}X_4$ will be fathomed when or before their widths are δ. However, nodes $\bar{X}_5 \in \mathbb{I}X_5$ may, in the worst case, need to be covered by boxes of width δ before they are fathomed. Furthermore, nodes $\bar{X}_3 \in \mathbb{I}X_3$ may also need to be covered by a large number of boxes depending on the convergence properties of the lower bounding scheme on X_3.

2. Analysis of the Cluster Problem

We assume that Problem (P) has a finite number of global minimizers, and ε is small enough that both X_3 and X_5 are contained in neighborhoods of the global minimizers.

Definition 6 (Nonisolated Feasible Point)**.** *A feasible point $\mathbf{x} \in \mathcal{F}(X)$ is said to be nonisolated if $\forall \alpha > 0$, $\exists \mathbf{z} \in \mathcal{N}_\alpha^1(\mathbf{x}) \cap \mathcal{F}(X)$ such that $\mathbf{z} \neq \mathbf{x}$.*

Definition 7 (Set of Active Inequality Constraints)**.** *Let $\mathbf{x} \in \mathcal{F}(X)$ be a feasible point for Problem* (P)*. The set of active inequality constraints at $\mathbf{x}$, denoted by $\mathcal{A}(\mathbf{x})$, is given by*

$$\mathcal{A}(\mathbf{x}) = \{j \in \{1, \cdots, m_I\} : g_j(\mathbf{x}) = 0\}.$$

2.1 Estimates for the number of boxes required to cover X_5

An estimate for the number of boxes required to cover some α-neighborhood, $\mathcal{N}_\alpha^1(\mathbf{x}^*)$, of $\mathbf{x}^*$ which contains the subset of X_5 around $\mathbf{x}^*$ is provided under suitable assumptions. We assume that $\mathbf{x}^*$ is a nonisolated feasible point; otherwise $\exists \alpha > 0$ such that $\mathcal{N}_\alpha^1(\mathbf{x}^*) \cap X_5 = \{\mathbf{x}^*\}$ which can be covered using a single box of width δ.

Lemma 8. *Suppose $\mathbf{x}^*$ is a nonisolated feasible point for Problem* (P) *and $\exists \alpha > 0$ such that $L := \inf_{\{d:\|d\|_1=1,\, \exists t>0 \text{ s.t. } (x^*+td)\in N_\alpha^1(x^*)\cap F(X)\}} \nabla f(\mathbf{x}^*)^{\mathsf{T}}\mathbf{d} > 0$. Then, $\exists \hat{\alpha} \in (0, \alpha]$ such that $\mathcal{N}_{\hat{\alpha}}^1(\mathbf{x}^*) \cap X_5$ is overestimated by $\hat{X}_5 = \left\{\mathbf{x} \in N_{\hat{\alpha}}^1(\mathbf{x}^*) : L\|\mathbf{x} - \mathbf{x}^*\|_1 \leq 2\varepsilon\right\}$.*

Theorem 9. *Suppose the assumptions of Lemma 8 hold. Define $r = \frac{2\varepsilon}{L}$ and recall $\delta = \left(\frac{\varepsilon}{\tau^*}\right)^{\frac{1}{\beta^*}}$.*

1. *If $\delta \geq 2r$, let $N = 1$.*
2. *If $\frac{2r}{m-1} > \delta \geq \frac{2r}{m}$ for some $m \in \mathbb{N}$ with $m \leq n_x$ and $2 \leq m \leq 6$, then let $N = \sum_{i=0}^{m-1} 2^i \binom{n_x}{i} + 2n_x \left\lceil \frac{m-3}{3} \right\rceil$.*
3. *Otherwise, let $N = \left\lceil \frac{2\varepsilon}{L\delta} \right\rceil^{n_x - 1} \left(\left\lceil \frac{2\varepsilon}{L\delta} \right\rceil + 2n_x \left\lceil \frac{\varepsilon}{L\delta} \right\rceil \right)$.*

Then, N is an upper bound on the number of boxes with width δ required to cover $\hat{X}_5$.

Remark 10. *Under the assumptions of Lemma 8, the dependence of N on ε disappears when the lower bounding scheme has first-order convergence on X_5, i.e., $\beta^* = 1$. Therefore, the cluster problem may be eliminated even using first-order convergent lower bounding schemes with sufficiently small prefactors. This is in contrast to unconstrained global optimization where at least second-order convergent lower bounding schemes are required to eliminate the cluster problem.*

2.2 Estimates for the number of boxes required to cover $X_3 \backslash \hat{X}_5$

An estimate for the number of boxes required to cover some α-neighborhood, $\mathcal{N}_\alpha^1(\mathbf{x}^*)$, of $\mathbf{x}^*$ which contains the subset of X_3 around $\mathbf{x}^*$ is provided under suitable assumptions. We assume that $\mathbf{x}^*$ is a constrained global minimizer; otherwise $\exists \alpha > 0$ such that $\mathcal{N}_\alpha^1(\mathbf{x}^*) \cap X_3 = \emptyset$. Furthermore, we assume $\mathbf{x}^*$ is at the center of a box of width δ placed while covering $\hat{X}_5$.

Lemma 11. *Consider Problem* (P) *satisfying Assumption 1. Suppose $\mathbf{x}^*$ is a constrained minimizer, and $\exists \alpha > 0$ and a set $\mathcal{D}_1$ such that $L_f := \inf_{d \in D_1 \cap D_I} \nabla f(\mathbf{x}^*)^{\mathsf{T}}\mathbf{d} > 0$, where $\mathcal{D}_I$ is defined as $\mathcal{D}_I := \left\{\mathbf{d} : \|\mathbf{d}\|_1 = 1,\ \exists t > 0 \text{ s.t. } (\mathbf{x}^* + t\mathbf{d}) \in \mathcal{N}_\alpha^1(\mathbf{x}^*) \cap \mathcal{F}^{\mathsf{C}}(X)\right\}$ and S^{C} denotes the complement of S in X, $L_I := \inf_{d \in D_I \backslash D_1} \max\left\{ \max_{j \in A(x^*)} \left\{\nabla g_j(\mathbf{x}^*)^{\mathsf{T}}\mathbf{d}\right\}, \max_{k \in \{1,\cdots,m_E\}} \left\{\left|\nabla h_k(\mathbf{x}^*)^{\mathsf{T}}\mathbf{d}\right|\right\} \right\} > 0$. Then, $\exists \hat{\alpha} \in (0, \alpha]$ such that the region $\mathcal{N}_{\hat{\alpha}}^1(\mathbf{x}^*) \cap X_3 \cap \left\{(\mathbf{x}^* + t\mathbf{d}) \in \mathcal{N}_{\hat{\alpha}}^1(\mathbf{x}^*) \cap \mathcal{F}^{\mathsf{C}}(X) : \mathbf{d} \in \mathcal{D}_1 \cap \mathcal{D}_I, t > 0\right\}$ is overestimated by $\hat{X}_3^1 = \left\{\mathbf{x} \in \mathcal{N}_{\hat{\alpha}}^1(\mathbf{x}^*) : L_f\|\mathbf{x} - \mathbf{x}^*\|_1 \leq 2\varepsilon^f\right\}$, and*
$\mathcal{N}_{\hat{\alpha}}^1(\mathbf{x}^*) \cap X_3 \cap \left\{\mathbf{x} = (\mathbf{x}^* + t\mathbf{d}) \in \mathcal{N}_{\hat{\alpha}}^1(\mathbf{x}^*) \cap \mathcal{F}^{\mathsf{C}}(X) : \mathbf{d} \in \mathcal{D}_I \backslash \mathcal{D}_1, t > 0\right\}$ is overestimated by $\hat{X}_3^2 = \left\{\mathbf{x} \in \mathcal{N}_{\hat{\alpha}}^1(\mathbf{x}^*) : L_I\|\mathbf{x} - \mathbf{x}^*\|_1 \leq 2\varepsilon^I\right\}$.

Furthermore, suppose $\mathbf{x}^*$ *is at the center of a box,* B_δ*, of width* δ *placed while covering* $\hat{X}_5$*. Then for* ε *small enough, the region* $\mathcal{N}^1_{\hat{\alpha}}(\mathbf{x}^*) \cap X_3 \cap \left\{(\mathbf{x}^* + t\mathbf{d}) \in \mathcal{N}^1_{\hat{\alpha}}(\mathbf{x}^*) \cap \mathcal{F}^{\mathsf{C}}(X) : \mathbf{d} \in \mathcal{D}_I \backslash \mathcal{D}_1, t > 0\right\} \backslash B_\delta$ *is overestimated by* $\left\{\mathbf{x} \in \mathcal{N}^1_{\hat{\alpha}}(\mathbf{x}^*) : \max\left\{d(\{\mathbf{g}(\mathbf{x})\}, \mathbb{R}^{m_I}_-), d(\{\mathbf{h}(\mathbf{x})\}, \{\mathbf{0}\})\right\} \in \left(\frac{L_I}{4}\delta, \varepsilon^I\right]\right\}$ *whenever* $L_I\delta < 4\varepsilon^I$*.*

Theorem 12. *Suppose the assumptions of Lemma 11 hold. Define* $\delta_f = \delta$*,* $\delta_I = \left(\frac{L_I\delta}{4\tau^I}\right)^{\frac{1}{\beta^I}}$*,* $r_I = \frac{2\varepsilon^I}{L_I}$*, and* $r_f = \frac{2\varepsilon^f}{L_f}$*. Then for* $j \in \{I, f\}$

1. *If* $\delta_j \geq 2r_j$*, let* $N_j = 1$*.*
2. *If* $\frac{2r_j}{m_j - 1} > \delta_j \geq \frac{2r_j}{m_j}$ *for some* $m_j \in \mathbb{N}$ *with* $m_j \leq n_x$ *and* $2 \leq m_j \leq 6$*, then let* $N_j = \sum_{i=0}^{m_j-1} 2^i \binom{n_x}{i} + 2n_x \left\lceil \frac{m_j - 3}{3} \right\rceil$*.*
3. *Otherwise, let* $N_j = \left\lceil \frac{2\varepsilon^j}{L_j\delta_j} \right\rceil^{n_x - 1} \left(\left\lceil \frac{2\varepsilon^j}{L_j\delta_j} \right\rceil + 2n_x \left\lceil \frac{\varepsilon^j}{L_j\delta_j} \right\rceil \right)$*.*

Then, N_I *is an upper bound on the number of boxes with width* δ_I *required to cover* $\hat{X}^2_3 \backslash \hat{X}_5$*, and* N_f *is an upper bound on the number of boxes with width* δ_f *required to cover* $\hat{X}^1_3$*.*

Remark 13. *Under the assumptions of Lemma 11, the dependence of* N_I *on* ε^I *disappears when the lower bounding scheme has first-order convergence on* X_3*, i.e.,* $\beta^I = 1$*, and the dependence of* N_f *on* ε^f *disappears when the scheme* $(f^{\mathrm{CV}}_Z)_{Z \in \mathbb{I}X}$ *has first-order convergence on* X*, i.e.,* $\beta^f = 1$*. Therefore, the cluster problem on* X_3 *can be eliminated even using first-order convergent lower bounding schemes with sufficiently small prefactors.*

3. Summary

A definition of convergence order for lower bounding schemes for constrained problems has been proposed, and an analysis of the cluster problem for constrained global optimization has been presented. It has been shown that first-order convergence of lower bounding schemes may be sufficient to eliminate the cluster problem under certain conditions. Conditions under which second-order convergence may be sufficient to avoid clustering can be similarly developed.

References

[1] Agustín Bompadre and Alexander Mitsos. Convergence rate of McCormick relaxations. *Journal of Global Optimization*, 52(1):1–28, 2012.

[2] Agustín Bompadre, Alexander Mitsos, and Benoît Chachuat. Convergence analysis of Taylor models and McCormick-Taylor models. *Journal of Global Optimization*, 57(1):75–114, 2013.

[3] Kaisheng Du and R Baker Kearfott. The cluster problem in multivariate global optimization. *Journal of Global Optimization*, 5(3):253–265, 1994.

[4] Arnold Neumaier. Complete search in continuous global optimization and constraint satisfaction. *Acta Numerica*, 13:271–369, 2004.

[5] Achim Wechsung. *Global optimization in reduced space*. PhD thesis, Massachusetts Institute of Technology, 2014.

[6] Achim Wechsung, Spencer D Schaber, and Paul I Barton. The cluster problem revisited. *Journal of Global Optimization*, 58(3):429–438, 2014.

Proceedings of GOW'16, pp. 13 – 16.

Distance Geometry: Too Much is Never Enough

Gustavo Dias[1] and Leo Liberti[1]

[1]*CNRS LIX, Ecole Polytechnique, 91128 Palaiseau, France.*
{dias,liberti}@lix.polytechnique.fr

Abstract Two years after presenting the distance geometry problem (DGP) as "the most beautiful problem I know" at the last Global Optimization Workshop in Malaga, one of the authors of this abstract (LL) confirms his DGP-mania by proposing lots of fun, weird, innovative, elegant and sometimes also practically useful methods for solving this problem, while drawing an unsuspecting Ph.D. student (GD) in the addiction. We present counterintuitive results which only make sense in very high dimensional spaces, adapt the celebrated Isomap heuristic to the DGP setting, and apply some recent techniques for finding feasible solutions of semidefinite programs using a linear programming solver. In short, we do all we can to solve very large DGP instances, albeit approximately.

Keywords: Random projections, Principal component analysis, Diagonally dominant matrix, Smoothing

1. Introduction

The Distance Geometry Problem (DGP) consists of "drawing" a weighted graph in a Euclidean space of given dimension, so that a drawn edge is as long as its weight. More precisely, given an integer $K > 0$ and a simple undirected weighted graph $G = (V, E, d)$, where $d : E \to \mathbb{R}_+$, the DGP asks whether there exists a *realization* $x : V \to \mathbb{R}^K$ such that:

$$\forall \{i, j\} \in E \quad \|x_i - x_j\|_2^2 = d_{ij}^2. \tag{1}$$

This problem is **NP**-hard [13] but is not known to be in **NP** [2] for $K > 1$.

A deceptively similar problem called Euclidean Distance Matrix Completion Problem (EDMCP), where K is not given, and the problem asks whether there exists a $K > 0$ such that Eq. (1) holds, is currently not known to be in **P** nor **NP**-hard.

The DGP arises in all applications where one can measure the distances but not the positions of entities: clock synchronization protocols (where $K = 1$ represents the timeline, and one is given time differences but needs to compute absolute clock times), localization of wireless sensors (where $K = 2$ represents e.g. a city block, or an office floor, and pairwise distances are estimated by the amount of battery power consumed in communication), protein conformation (where $K = 3$, and distances are estimated using Nuclear Magnetic Resonance experiments, and the protein binds to a site according to the relative position of its atoms), control of unmanned underwater vehicles (where again $K = 3$, distances are estimated by sonar, and the position cannot be verified directly since GPS signal does not reach underwater). See [8] for more information.

Our favorite method for solving DGPs is Branch-and-Prune (BP) [7]. It scales up to huge sizes [12], is blazingly fast, incredibly accurate [5], polynomial-time "on proteins" [9], and potentially finds all incongruent solutions. But it does not gracefully adapt to distance errors [3] and, most importantly, only works on graphs with a special structure [4]. And so we turn to approximate methods, heuristics, and relaxations.

In this abstract we summarize some of the recent efforts in solving very large DGP instances approximately. We accept approximate solutions because (a) applications usually provide us distances with some errors, and (b) because exact methods do not necessarily scale up to large sizes.

2. Random projections

High dimensional spaces are host to some weird, counterintuitive and somewhat magical-looking phenomena [6]. The one we are specifically interested in is the Johnson-Lindenstrauss Lemma (JLL), which states that if you have a realization x of n points in $\mathbb{R}^K$ and some $\varepsilon \in (0,1)$, then there exists a $k = O((1/\varepsilon^2)\log n)$ and a $k \times K$ matrix T such that:

$$\forall i,j \in V \quad (1-\varepsilon)\|x_i - x_j\|_2^2 \leq \|Tx_i - Tx_j\|_2^2 \leq (1+\varepsilon)\|x_i - x_j\|_2^2. \tag{2}$$

In fact, if you sample each component of T from $N(0, \sqrt{1/k})$, Eq. (2) holds with probability which approaches 1 exponentially fast as k grows. If you try this out in small dimensional spaces, you will soon see that this is hopeless, which adds a touch of magic to the JLL. We find it even more surprising that the target dimension k is independent of the original dimension K.

Note that the JLL provides a dimensionality reduction mechanism, rather than a solution method for the DGP. Finding a DGP solution in a high dimensional space, however, is easier than finding one with fewer degrees of freedom. So we can project high-dimensional solutions to lower dimensions while keeping the pairwise distances approximately equal. Note that the target dimension k cannot be given: so the JLL applies to the EDMCP rather than the DGP.

Other types of random projections exist, such as Matoušek's, which we also consider.

3. Isomap

The Isomap method [14] is a heuristic method best known for dimensionality reduction, much like the JLL. It works as follows: from a set of n points $X \subseteq \mathbb{R}^K$ we derive a weighted graph $G = (V, E, d)$ from all distances smaller than a given threshold (chosen so as to make the graph connected and reasonably sparse). Note that every edge is weighted with the corresponding Euclidean distance. Next, we complete G to a clique $\bar{G}$ by computing the missing distances using an all (weighted) shortest path algorithm such as Floyd-Warshall. The complete graph $\bar{G}$ is encoded in a symmetric matrix $\bar{D}$ which is an approximation of the (squared) Euclidean Distance Matrix of X. Then we perform classic Multi-Dimensional Scaling (MDS) on $\bar{D}$:

$$\mathcal{G} = -\frac{1}{2} J \bar{D} J, \tag{3}$$

where $\mathcal{G}$ is an approximation of the Gram matrix of X, $J = I - \mathbf{1}/n$, and $\mathbf{1}$ is the all-one $n \times n$ matrix. Since Gram matrices are positive semidefinite (PSD), their eigenvalue matrix Λ has non-negative diagonal, and they can be factored into $YY^\top$ where $Y = P\sqrt{\Lambda}$. $\mathcal{G}$ is not a Gram matrix, however, but only an approximation: so we zero all the negative eigenvalues in Λ (so $\sqrt{\Lambda}$ is real). Finally, we perform a Principal Component Analysis (PCA) step, and discard all but the first K largest eigenvalues of Λ. This yields a set Y of n points in $\mathbb{R}^K$.

Note that Isomap is *almost* a method for solving the DGP. Our "adaptation" consists in a simpe remark: just start Isomap from the weighted graph G.

4. Diagonally dominant programming

In a ground-breaking result, Ahmadi and Hall [1] showed that it is possible to find feasible Semidefinite Programming (SDP) solutions using a Linear Programming (LP) solver. Since SDP solution technology has a considerable computational bottleneck, this result has the potential for unlocking more SDP power. This result is based on the observation that a diagonally dominant (DD) $n \times n$ matrix $X = (X_{ij})$, namely one such that

$$\forall i \leq n \quad X_{ii} \geq \sum_{j \neq i} |X_{ij}|, \tag{4}$$

is also PSD. Note that Eq. (4) can be written linearly by introducing a matrix Y and the constraints:

$$\begin{array}{lrcl} \forall i \leq n & \sum_{j \neq i} Y_{ij} & \leq & X_{ii} \\ & -Y \leq & X & \leq Y. \end{array}$$

This means that the PSD constraint $X \succeq 0$ in any SDP can be replaced by the LP constraints above. Programming over those constraints is known as Diagonally Dominant Programming (DDP).

Note that DD implies PSD but not vice-versa. Hence a DDP formulation provides an inner approximation of the SDP feasible region. If the original SDP is used to compute bounds, the guarantee is lost; but since SDP has strong duality, it suffices to apply DDP to the SDP dual. Moreover, the DDP might be infeasible even if the original SDP is feasible. To overcome this issue, Ahmadi and Hall provide an iterative improvement algorithm which enlarges the feasible region of the DDP at each step.

We provide and test DDP formulations for the DGP and EDMCP.

5. The DGSol algorithm

This algorithm was proposed around 20 years ago [10], but it is still very competitive in terms of speed (also thanks to a very good implementation). For smaller scale instances the accuracy of the solutions is not impressive. What is impressive, however, is how well DGSol scales with size in both speed and accuracy. In this sense, DGSol is a truly "big data" kind of method.

The algorithm behind DGSol has an outer and an inner iteration. The outer iteration starts from a smoothed convexified version of the penalty objective function,

$$f(x) = \sum_{\{i,j\} \in E} \left(\|x_i - x_j\|_2^2 - d_{ij}^2 \right)^2$$

obtained via a Gaussian transform

$$\langle f \rangle_\lambda(x) = \frac{1}{\pi^{Kn/2} \lambda^{Kn}} \int\limits_{\mathbb{R}^{Kn}} f(y) \exp(-\|y - x\|_2^2 / \lambda^2) \mathsf{d}y,$$

which tends to $f(x)$ as $\lambda \to 0$.

For each fixed value of λ in the outer iteration, the inner iteration is based on the recursion

$$x^{\ell+1} = x^\ell - \alpha_\ell H_\ell \nabla f(x^\ell),$$

for $\ell \in \mathbb{N}$, where α_ℓ is a step size, and H_ℓ is an approximation of the inverse Hessian matrix of f. In other words, the inner iteration implements a local NLP solution method which uses the optimum at the previous value of λ as a starting point.

Overall, this yields a homotopy method which traces a trajectory depending on $\lambda \to 0$, where a unique (global) optimum of the convex smoothed function $\langle f \rangle_\lambda$ for a high enough value of λ (hopefully) follows the trajectory to the global minimum of the multimodal, non-convex function $\langle f \rangle_0 = f$.

We use DGSol as a benchmark for comparison. We also borrow its local NLP subsolver for efficiently improving the approximate methods discussed above in a post-processing phase.

6. Conclusion

Our investigations in alternative methods for the DGP are focused towards identifying the best methods for solving very large scale instances of the DGP and EDMCP. Aside from being

interesting in their own right, we eventually plan to use them within the BP algorithm in order to provide a better extension for dealing with imprecise data.

Acknowledgments

The first author is financially supported by a CNPq Ph.D. thesis award. The second is partially sponsored by the Bip:Bip ANR project under contract ANR-10-BINF-0003.

References

[1] A. Ahmadi and G. Hall. Sum of squares basis pursuit with linear and second order cone programming. Technical Report 1510.01597v1, arXiv, 2015.

[2] N. Beeker, S. Gaubert, C. Glusa, and L. Liberti. Is the distance geometry problem in **NP**? In Mucherino et al. [11].

[3] A. Cassioli, B. Bordeaux, G. Bouvier, A. Mucherino, R. Alves, L. Liberti, M. Nilges, C. Lavor, and T. Malliavin. An algorithm to enumerate all possible protein conformations verifying a set of distance constraints. *BMC Bioinformatics*, page 16:23, 2015.

[4] C. Lavor, J. Lee, A. Lee-St. John, L. Liberti, A. Mucherino, and M. Sviridenko. Discretization orders for distance geometry problems. *Optimization Letters*, 6:783–796, 2012.

[5] C. Lavor, L. Liberti, N. Maculan, and A. Mucherino. The discretizable molecular distance geometry problem. *Computational Optimization and Applications*, 52:115–146, 2012.

[6] M. Ledoux. *The concentration of measure phenomenon*. Number 89 in Mathematical Surveys and Monographs. American Mathematical Society, Providence, 2005.

[7] 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.

[8] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino. Euclidean distance geometry and applications. *SIAM Review*, 56(1):3–69, 2014.

[9] L. Liberti, C. Lavor, and A. Mucherino. The discretizable molecular distance geometry problem seems easier on proteins. In Mucherino et al. [11].

[10] J. Moré and Z. Wu. Global continuation for distance geometry problems. *SIAM Journal of Optimization*, 7(3):814–846, 1997.

[11] A. Mucherino, C. Lavor, L. Liberti, and N. Maculan, editors. *Distance Geometry: Theory, Methods, and Applications*. Springer, New York, 2013.

[12] A. Mucherino, C. Lavor, L. Liberti, and E-G. Talbi. A parallel version of the branch & prune algorithm for the molecular distance geometry problem. In *ACS/IEEE International Conference on Computer Systems and Applications (AICCSA10)*, pages 1–6, Hammamet, Tunisia, 2010. IEEE.

[13] J. Saxe. Embeddability of weighted graphs in k-space is strongly **NP**-hard. *Proceedings of 17th Allerton Conference in Communications, Control and Computing*, pages 480–489, 1979.

[14] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2322, 2000.

Proceedings of GOW'16, pp. 17 – 20.

Testing Pseudoconvexity via Interval Computation

Milan Hladík[1]

[1]*Charles University, Faculty of Mathematics and Physics, Department of Applied Mathematics, Malostranské nám. 25, 11800, Prague, Czech Republic,* milan.hladik@matfyz.cz

Abstract We study the problem of checking pseudoconvexity of a twice differentiable function on an interval domain. Based on several characterizations of pseudoconvexity of a real function, we propose sufficient conditions for verifying pseudoconvexity on a domain formed by a Cartesian product of real intervals. In the sequel, we will carry out numerical experiments to show which methods perform well from two perspectives – the computational complexity and effectiveness of recognizing pseudoconvexity.

Keywords: Interval computation, Pseudoconvexity

1. Introduction

Some methods in deterministic global optimization [6, 7, 8] are based on a branch-and-bound scheme and utilizing interval methods for rigorous inner and outer approximations, among others. In particular, αBB method and its variants [6, 9] create convex underestimators of nonconvex functions on interval regions. Convexity is very convenient property in the context of optimization, however, not always it is achieved. That is why diverse generalized concepts of convexity were thoroughly investigated in the past. In particular, quasiconvexity and pseudoconvexity are among the most commonly used generalizations.

Pseudoconvex objective functions have some nice properties in the context of optimization: On the convex feasible set, each stationary point is a global minimum, each local minimum is a global minimum, and the optimal solution set is convex.

The aim of this paper is to develop methods for checking pseudoconvexity on an interval domain.

1.1 Interval computation

Interval notation. An interval matrix is defined as

$$\boldsymbol{A} := \{A \in \mathbb{R}^{m\times n};\ \underline{A} \le A \le \overline{A}\},$$

where $\underline{A}$ and $\overline{A}$, $\underline{A} \le \overline{A}$, are given matrices and the inequality is understood entrywise. The midpoint and radius matrices are defined as

$$A_c := \frac{1}{2}(\underline{A} + \overline{A}), \quad A_\Delta := \frac{1}{2}(\overline{A} - \underline{A}).$$

The set of all interval $m \times n$ matrices is denoted by $\mathbb{IR}^{m\times n}$. Interval vectors are defined as one-column interval matrices. For interval arithmetic see, e.g., [7].

Other notation and definitions. The diagonal matrix with entries $s_1, \ldots, s_n$ is denoted by $\mathrm{diag}(s)$, and the spectral radius of $A \in \mathbb{R}^{n\times n}$ by $\rho(A)$. For a symmetric $A \in \mathbb{R}^{n\times n}$, we sort its eigenvalues as $\lambda_1(A) \ge \ldots \ge \lambda_n(A)$.

Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be twice differentiable and $\mathcal{S} \subset \mathbb{R}^n$ an open convex set. Then $f(x)$ is *pseudoconvex* on $\mathcal{S}$ if for every $x, y \in \mathcal{S}$ we have

$$\nabla f(x)^T(y - x) \ge 0 \ \Rightarrow \ f(y) \ge f(x).$$

1.2 Problem formulation

Throughout this paper, $\boldsymbol{x} \in \mathbb{IR}^n$ is a given box with nonempty interior, and $f\colon \mathbb{R}^n \to \mathbb{R}$ is a differentiable function on an open set containing $\boldsymbol{x}$. The question studied is whether $f(x)$ is quasiconvex on $\boldsymbol{x}$.

Even though $\boldsymbol{x}$ is not an open set, by using one-sided derivatives, we can extend characterization of quasiconvexity and pseudoconvexity to $\boldsymbol{x}$.

Ahmadi et al. [1] showed that deciding pseudoconvexity is NP-hardness on a class of quartic polynomials. This result indicates that our problem considering any differentiable function is also difficult.

1.3 Characterizations of pseudoconvexity

We review some known [2, 3, 4, 5, 10] second order characterizations of pseudoconvexity that seem to be convenient for interval-based methods for recognizing pseudoconvexity. The statements below are adapted to our problem.

Using the characterization by Mereau and Paquet [10], we have:

Theorem 1 (Mereau and Paquet, 1974)**.** *The function $f(x)$ is pseudoconvex on $\boldsymbol{x}$ if there is $\alpha \geq 0$ such that*

$$M_\alpha(x) := \nabla^2 f(x) + \alpha \nabla f(x) \nabla f(x)^T$$

is positive semidefinite for all $x \in \boldsymbol{x}$.

Denote

$$D(x) := \begin{pmatrix} 0 & \nabla f(x)^T \\ \nabla f(x) & \nabla^2 f(x) \end{pmatrix},$$

and by $D(x)_r$ we denote the principal leading submatrix (i.e., left top submatrix) of size r. The condition by Ferland [4, 5] follows.

Theorem 2 (Ferland, 1972)**.** *The function $f(x)$ is pseudoconvex on $\boldsymbol{x}$ if $\det(D(x)_r) < 0$ for every $r = 2, \ldots, n+1$ and for all $x \in \boldsymbol{x}$.*

Another condition is by Crouzeix and Ferland [3].

Theorem 3 (Crouzeix and Ferland, 1982)**.** *The function $f(x)$ is pseudoconvex on $\boldsymbol{x}$ if for each $x \in \boldsymbol{x}$ either $\nabla^2 f(x)$ is positive semidefinite, or $\nabla^2 f(x)$ has one simple negative eigenvalue and there is $b \in \mathbb{R}^n$ such that $\nabla^2 f(x) b = \nabla f(x)$ and $\nabla f(x)^T b < 0$.*

Eventually, we mention a condition by Crouzeix [2].

Theorem 4 (Crouzeix, 1998)**.** *The function $f(x)$ is pseudoconvex on $\boldsymbol{x}$ if for each $x \in \boldsymbol{x}$ the matrix $D(x)$ is nonsingular and has exactly one simple negative eigenvalue.*

2. Interval methods for testing pseudoconvexity

Let $\boldsymbol{H} \in \mathbb{IR}^{n \times n}$ and $\boldsymbol{g} \in \mathbb{IR}^n$ such that

$$\begin{aligned} &\nabla^2 f(x) \in \boldsymbol{H} \ \ \forall x \in \boldsymbol{x}, \\ &\nabla f(x) \in \boldsymbol{g} \ \ \forall x \in \boldsymbol{x}. \end{aligned}$$

Such interval enclosures of the Hessian matrix and the gradient can be computed, e.g., by interval arithmetic using automatic differentiation. Of course, the tighter enclosure used the better, however, computing a tight enclosure is a computationally hard problem in general.

If every $H \in \boldsymbol{H}$ is positive semidefinite, then $f(x)$ is convex and we are done. Therefore, we focus on problems such that not every $H \in \boldsymbol{H}$ is positive semidefinite.

We will also use the symmetric interval matrix

$$\boldsymbol{D} := \begin{pmatrix} 0 & \boldsymbol{g}^T \\ \boldsymbol{g} & \boldsymbol{H} \end{pmatrix}.$$

2.1 Method based on Theorem 1

Theorem 1 suggests that pseudoconvexity of $f(x)$ can be checked by verifying positive semidefiniteness of matrices

$$M_\alpha(H, g) := H + \alpha g g^T, \quad H \in \boldsymbol{H},\ g \in \boldsymbol{g}$$

for a suitable $\alpha \geq 0$.

The direct approach is to evaluate

$$\boldsymbol{M}(\alpha) := \boldsymbol{H} + \alpha \boldsymbol{g}\boldsymbol{g}^T$$

by interval arithmetic and for a suitable $\alpha \geq 0$. Then we check whether $\boldsymbol{M}(\alpha)$ is positive semidefinite, i.e., whether every $M_\alpha \in \boldsymbol{M}(\alpha)$ is positive semidefinite. It was proved by [11] that checking this property is co-NP-hard. Sufficient and necessary condition is that all matrices of the form

$$M(\alpha)_c - \operatorname{diag}(z) M(\alpha)_\Delta \operatorname{diag}(z) \tag{1}$$

where $z \in \{\pm 1\}^{n-1} \times \{1\}$, are positive semidefinite. An easy to verify sufficient condition is $\lambda_n(M(\alpha)_c) \geq \rho(M(\alpha)_\Delta)$ (see, e.g., [12]).

2.2 Method based on Theorem 2

By Theorem 2, for pseudoconvexity of $f(x)$ on $\boldsymbol{x}$, it is sufficient to check that for each symmetric $D \in \boldsymbol{D}$ and for each $r = 2, \ldots, n+1$ we have $\det(D_r) < 0$. This is, however, a co-NP-hard problem.

Theorem 5. *It is co-NP-hard to check whether* $\det(D) < 0$ *for every symmetric* $D \in \boldsymbol{D}$.

Due to co-NP-hardness, the problem might be computationally expensive in the worst case, so we propose an efficient sufficient condition instead. Let $r \in \{2, \ldots, n+1\}$. Then the condition that $\det(D_r) < 0$ for each symmetric $D_r \in \boldsymbol{D}_r$ can be checked by showing $\det((D_r)_c) < 0$ and $\rho(|(D_r)_c^{-1}|(D_r)_\Delta) < 1$. The former says that determinant of the midpoint matrix is negative, and the second one guarantees nonsingularity. If every symmetric $D_r \in \boldsymbol{D}_r$ is nonsingular, then D_r has constant number of positive and negative eigenvalues, and so has constant sign of the determinant.

2.3 Method based on Theorem 3

After a small modification, the condition for pseudoconvexity based on Theorem 3 can be expressed as follows.

Theorem 6. *The function* $f(x)$ *is pseudoconvex on* $\boldsymbol{x}$ *if for each symmetric* $D \in \boldsymbol{D}$ *we have* $\det(D) < 0$, *and each symmetric* $H \in \boldsymbol{H}$ *is nonsingular and has at most one simple negative eigenvalue.*

In view of Theorem 5, the above condition is hard to verify exactly, so we show a sufficient condition as well.

Theorem 7. *The function* $f(x)$ *is pseudoconvex on* $\boldsymbol{x}$ *if*

$$\det(D_c) < 0, \ \ \rho(|D_c^{-1}|D_\Delta) < 1, \ \ \textit{and} \ \ 0 < \lambda_{n-1}(H_c) - \rho(H_\Delta).$$

2.4 Method based on Theorem 4

Since $D(x)$ has at least one negative eigenvalue by [2], it is sufficient to check that the $(n-1)$th largest eigenvalue is positive. In the interval context, we have to check that the $(n-1)$th largest eigenvalue of every symmetric matrix $D \in \boldsymbol{D}$ is positive. A sufficient condition is:

Theorem 8. *The function $f(x)$ is pseudoconvex on $\boldsymbol{x}$ if $0 \notin \boldsymbol{g}$ and $\lambda_n(D_c) > \rho(D_\Delta)$.*

3. Conclusion

We considered the problem of testing pseudoconvexity of a general differentiable function on an interval domain. We utilized various second order characterizations of pseudoconvexity, and proposed several methods.

We will make a thorough numerical comparisons to find out which methods are the best from the computational point of view and which methods have the highest success rate for checking pseudoconvexity.

Acknowledgments

The author was supported by the Czech Science Foundation Grant P402/13-10660S.

References

[1] Amir Ali Ahmadi, Alex Olshevsky, Pablo A. Parrilo, and John N. Tsitsiklis. NP-hardness of deciding convexity of quartic polynomials and related problems. *Mathematical Programming*, 137(1-2):453–476, 2013.

[2] Jean-Pierre Crouzeix. Characterizations of Generalized Convexity and Generalized Monotonicity, A Survey. In J.-P. Crouzeix, J.-E. Martinez-Legaz, and M. Volle, editors, *Generalized Convexity, Generalized Monotonicity: Recent Results*, volume 27 of *Nonconvex Optimization and Its Applications*, pages 237–256. Springer, 1998.

[3] Jean-Pierre Crouzeix and Jacques A. Ferland. Criteria for quasi-convexity and pseudo-convexity: Relationships and comparisons. *Mathematical Programming*, 23(1):193–205, 1982.

[4] Jacques A. Ferland. Mathematical programming problems with quasi-convex objective functions. *Mathematical Programming*, 3(1):296–301, 1972.

[5] Jacques A. Ferland. Matrix criteria for pseudo-convex functions in the class C^2. *Linear Algebra and its Applications*, 21(1):47–57, 1978.

[6] Christodoulos A. Floudas. *Deterministic Global Optimization. Theory, Methods and Applications*, volume 37 of *Nonconvex Optimization and its Applications*. Kluwer, Dordrecht, 2000.

[7] Eldon R. Hansen and G. William Walster. *Global optimization using interval analysis*. Marcel Dekker, New York, second edition, 2004.

[8] Eligius M. T. Hendrix and Boglárka Gazdag-Tóth. *Introduction to nonlinear and global optimization*, volume 37 of *Optimization and Its Applications*. Springer, New York, 2010.

[9] Milan Hladík. An extension of the αBB-type underestimation to linear parametric Hessian matrices. *Journal of Global Optimization*, 64(2):217–231, 2016.

[10] Pierre Mereau and Jean-Guy Paquet. Second order conditions for pseudo-convex functions. *SIAM Journal on Applied Mathematics*, 27:131–137, 1974.

[11] A. Nemirovskii. Several NP-hard problems arising in robust stability analysis. *MCSS. Mathematics of Control, Signals, and Systems*, 6(2):99–105, 1993.

[12] Jiří Rohn. A handbook of results on interval linear problems. Technical Report 1163, Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague, 2012. `http://uivtx.cs.cas.cz/~rohn/publist/!aahandbook.pdf`.

Proceedings of GOW'16, pp. 21 – 24.

A Preference-Based Multi-Objective Evolutionary Algorithm for Approximating a Region of Interest*

Ernestas Filatovas[1], Olga Kurasova[2], Juana López Redondo[3] and José Fernández[4]

[1] *Vilnius Gediminas Technical University, Sauletekio av. 11, Vilnius, Lithuania* ernest.filatov@gmail.com

[2] *Vilnius University, Universiteto str. 3, Vilnius, Lithuania,* olga.kurasova@mii.vu.lt

[3] *University of Almería, Ctra. Sacramento s/n. La Cañada de San Urbano, Almería, Spain,* jlredondo@ual.es

[4] *University of Murcia, Campus de Espinardo, Murcia, Spain,* josefdez@um.es

Abstract Multi-objective preference-based evolutionary algorithms approximate the part of the Pareto front that meets the preference information expressed by the Decision Maker. However, only a few of such algorithms are able to obtain well-distributed solutions covering the complete "region of interest". In this work a preference-based evolutionary algorithm for approximating the region of interest of multi-objective optimization problems is proposed. The efficiency of the proposed algorithm has been experimentally evaluated and compared to another state-of-the-art multi-objective preference-based evolutionary algorithm by solving a set of multi-objective optimization benchmark problems.

Keywords: Multi-objective Optimization, Evolutionary Algorithms, Preference-based Algorithms

1. Introduction

Many real-world optimization problems are multi-objective, where several conflicting objectives should be optimized. Let us have $k \geq 2$ conflicting objectives, described by the functions $f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})$, where $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is a vector of variables (*decision vector*), and n is the number of variables. A multi-objective minimization problem is formulated as follows [8]:

$$\text{minimize} \quad \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})], \tag{1}$$

$$\text{subject to} \quad \mathbf{x} \in \mathbf{S}. \tag{2}$$

where $\mathbf{z} = \mathbf{f}(\mathbf{x}) \in \mathbb{R}^k$ is the *objective vector* function, and $\mathbf{S} \subset \mathbb{R}^n$ is called the *feasible region*.

A decision vector $\mathbf{x'} \in \mathbf{S}$ is a *Pareto optimal solution* if there no exists another $\mathbf{x} \in \mathbf{S}$ such that $f_i(\mathbf{x}) \leqslant f_i(\mathbf{x'})$ for all i and $f_j(\mathbf{x}) < f_j(\mathbf{x'})$ for at least one j. Objective vectors are defined as Pareto optimal if none of their elements can be improved without worsening at least one of the other elements. An objective vector $\mathbf{f}(\mathbf{x'})$ is Pareto optimal if the corresponding decision vector $\mathbf{x'}$ is Pareto optimal. The set of all Pareto optimal decision vectors is called the *Pareto set*. The region defined by all the objective function values of the points of the Pareto set is called the *Pareto front*. For two objective vectors $\mathbf{z}, \mathbf{z'} \in \mathbb{R}^k$, $\mathbf{z'}$ *dominates* $\mathbf{z}$ (or $\mathbf{z'} \succ \mathbf{z}$) if $z'_i \leqslant z_i$ for all $i = 1, \ldots, k$, and there exists at least one j such that $z'_j < z_j$.

The majority of multi-objective optimization problems are NP-hard. That is why algorithms that approximate the Pareto front are widely-used. Evolutionary Multi-objective Optimization (EMO) approaches are commonly employed for this task [2, 12]. The set of obtained

*Juana López Redondo is a fellow of the Spanish 'Ramón y Cajal' contract program, co-financed by the European Social Fund.

solutions approximating the entire Pareto front is presented to the Decision Maker (DM). However, such EMO algorithms are computationally expensive and time consuming. Additionally, only a reasonable number of solutions should be given to the DM so that he/she can make an adequate decision avoiding the usually complex analysis of a large amount of information and reducing cognitive burden. Moreover, the DM is commonly interested only in a certain part of the Pareto front and prefers to explore that part deeper. Thus, incorporation of DM's preferences into EMO algorithms has become a relevant trend during the last decade [4, 7, 9, 10, 11].

The DM's preference information is usually expressed as a reference point (RP), therefore a preference-based EMO algorithm, during its execution, emphasises solutions close to the RP. In particular, the so-called "region of interest" is a part of the Pareto front determined by the RP provided by the DM (see Figure 1). However, only a few preference-based EMO approaches are able to obtain well-distributed solutions covering the complete region of interest.

In this paper, we propose a preference-based EMO algorithm called NSGA-NBI that considers the DM's preference information expressed by means of a reference point. The solutions obtained by the algorithm are well-distributed and cover the complete region of interest.

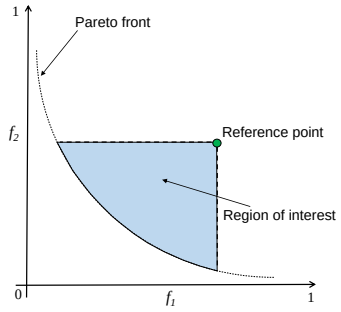


Figure 1. Region of interest.

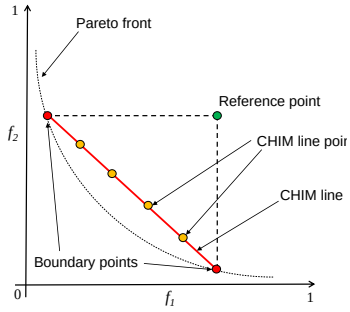


Figure 2. CHIM line of the region of interest.

2. Preference-based EMO algorithm NSGA-NBI

We combine ideas of the well-known EMO algorithm NSGA-II [3] and the NBI method [1]. As we deal with a preference-based approach, before running the NSGA-NBI algorithm, the DM is asked to provide a RP. Then the boundary points (see Figure 2) of the region of interest are obtained by a suitable single-objective optimization algorithm.

The steps of the NSGA-NBI algorithm are as follows:

1. A random initial population P_0 consisting of N decision vectors is randomly generated.
2. At iteration t, a new offspring population Q_t is created by applying genetic operators (crossover and mutation) to the individuals of the parent population.
3. The parent and offspring populations are combined into one joint population P_t .
4. The new population is sorted into different non-domination levels (so-called fronts) by a non-dominated sorting procedure (as in NSGA-II algorithm).
5. The obtained joint population is reduced to the size N of the parent population by leaving the individuals from the best non-domination levels. If not all individuals from the last level can be selected for the next generation, then CHIM line points are generated (as in NBI method) and the individuals closest to those points are selected. The CHIM line

points are evenly distributed between the boundary points (see Figure 2). The number of points on the CHIM line is equal to the number of points to be selected for the new population from the last non-dominated level that cannot be completely selected.

6. If the termination condition is not satisfied, then the process is repeated from Step 2 considering the reduced population as the parent population in the next iteration.

3. Results and conclusions

The proposed algorithm has been experimentally compared with a recently developed WASFGA algorithm [9] also designed to approximate the region of interest. The performance of the preference-based EMO algorithms has been evaluated using the following metrics: *Generational Distance (GD)* [2] for estimating convergence to the true Pareto-front, *Spread* [2] for estimating the distribution evenness of the solutions, *Hypervolume (HV)* [13] for estimating both convergence and distribution. The new *PR metric* [5] is also considered – it evaluates the percentage of solutions that lie into the DM's region of interest. The set of well-known 2-objective test problems ZDT1–ZDT4, ZDT6 with different complexity and characteristics (convexity, concavity, discontinuity, non-uniformity, ...) has been considered [6].

Each algorithm has been run 30 times using different initial populations and average results have been evaluated. We selected a population size of 100 individuals and 100 generations. Those values are enough to obtain a good approximation of a region of the Pareto front in reasonable computational time.

Preference information provided by the DM that is expressed as a RP is required for the evaluated algorithms, therefore various RPs (achievable as well as unachievable) have been selected. The used RPs, the numbers of objectives and variables for each test problem considered are presented in Table 1.

We can see in Tables 2 and 3 that the proposed NSGA-NBI algorithm is superior in most of the cases (better values are marked in bold). In all the analysed cases, and according to the GD metric, the proposed algorithm approximates the region of interest better than the WASFGA algorithm. The values of Spread metric show that the solutions obtained by the NSGA-NBI algorithm are better evenly distributed in most of the cases. The proposed algorithm is superior for all analysed cases according to the HV metric. The PR metric indicates that the proposed algorithm is able to obtain sufficiently high number of solutions in the region of interest for all analysed problems unlike the WASFGA algorithm. When solving the ZDT6 problem with achievable RP the WASFGA algorithm could not obtain any solutions in the region of interest, therefore the mean values of GD, Spread, and HV metrics could not be calculated – we show these values as NaN (see Table 2). In conclusion, the proposed algorithm NSGA-NBI has shown promising results, as it has obtained competitive values for all the quality indicators considered.

Table 1. Test problems and reference points used in the evaluated algorithms

Problem	Number of objectives	Number of variables	Achievable reference point	Unachievable reference point
ZDT1	2	30	(0.80, 0.60)	(0.20, 0.40)
ZDT2	2	30	(0.80, 0.80)	(0.50, 0.30)
ZDT3	2	30	(0.30, 0.80)	(0.05, 0.00)
ZDT4	2	10	(0.80, 0.60)	(0.08, 0.25)
ZDT6	2	10	(0.78, 0.61)	(0.39, 0.21)

Table 2. Values of performance metrics for achievable RPs

	GD		Spread		HV		PR	
Problem	NSGA-NBI	WASFGA	NSGA-NBI	WASFGA	NSGA-NBI	WASFGA	NSGA-NBI	WASFGA
ZDT1	**0.0004**	0.0160	**0.0049**	0.0064	**0.1760**	0.1612	98.47	**100.00**
ZDT2	**0.0018**	0.0344	**0.0065**	0.0331	**0.0687**	0.0513	**98.17**	95.97
ZDT3	**0.0005**	0.0101	**0.0062**	0.0066	**0.0689**	0.0590	99.43	**100.00**
ZDT4	**0.0003**	0.1345	**0.0046**	0.0202	**0.1758**	0.0649	**98.57**	55.50
ZDT6	**0.0004**	NaN	**0.0012**	NaN	**0.0162**	NaN	**98.87**	0.00

Table 3. Values of performance metrics for unachievable RPs

	GD		Spread		HV		PR	
Problem	NSGA-NBI	WASFGA	NSGA-NBI	WASFGA	NSGA-NBI	WASFGA	NSGA-NBI	WASFGA
ZDT1	**0.0003**	0.0107	**0.0047**	0.0078	**0.0127**	0.0101	**98.80**	81.13
ZDT2	**0.0009**	0.0214	**0.0054**	0.0066	**0.0716**	0.0592	**97.44**	80.20
ZDT3	**0.0006**	0.0085	0.0207	**0.0078**	**0.1150**	0.1049	93.77	**93.93**
ZDT4	**0.1048**	0.1938	0.1048	**0.0520**	**0.2167**	0.0411	**88.83**	36.43
ZDT6	**0.0002**	0.1769	**0.0017**	0.0320	**0.1354**	0.0299	**98.90**	28.27

References

[1] Indraneel Das and John E. Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998.

[2] Kalyanmoy Deb. *Multi-objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, 2001.

[3] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[4] Kalyanmoy Deb, J. Sundar, N. Udaya Bhaskara Rao, and Shamik Chaudhuri. Reference point based multi-objective optimization using evolutionary algorithms. *International Journal of Computational Intelligence Research*, 2(3):273–286, 2006.

[5] Ernestas Filatovas, Algirdas Lancinskas, Olga Kurasova, and Julius Zilinskas. A preference-based multi-objective evolutionary algorithm R-NSGA-II with stochastic local search. *Central European Journal of Operations Research*, Accepted.

[6] Simon Huband, Philip Hingston, Luigi Barone, and Lyndon While. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, 2006.

[7] Antonio López-Jaimes and Carlos A. Coello Coello. Including preferences into a multiobjective evolutionary algorithm to deal with many-objective engineering optimization problems. *Information Sciences*, 277:1–20, 2014.

[8] Kaisa Miettinen. *Nonlinear Multiobjective Optimization*. Springer, 1999.

[9] Ana Belén Ruiz, Rubén Saborido, and Mariano Luque. A preference-based evolutionary algorithm for multiobjective optimization: the weighting achievement scalarizing function genetic algorithm. *Journal of Global Optimization*, pages 1–29, 2014.

[10] Florian Siegmund, Amos H.C. Ng, and Kalyanmoy Deb. Finding a preferred diverse set of Pareto-optimal solutions for a limited number of function calls. In *2012 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2012.

[11] Karthik Sindhya, Ana Belen Ruiz, and Kaisa Miettinen. A preference based interactive evolutionary algorithm for multi-objective optimization: PIE. In *6th International Conference on Evolutionary Multi-criterion Optimization, EMO 2011*, pages 212–225. Springer, 2011.

[12] El-Ghazali Talbi. *Metaheuristics: from Design to Implementation*, volume 74. John Wiley & Sons, 2009.

[13] Eckart Zitzler and Lothar Thiele. Multiobjective optimization using evolutionary algorithms - a comparative case study. In *Parallel Problem Solving from Nature - PPSN V*, pages 292–301. Springer, 1998.

Proceedings of GOW'16, pp. 25 – 28.

Energy-Aware Computation of Evolutionary Multi-Objective Optimization*

G. Ortega[1], J.J. Moreno[1], E. Filatovas[2], J.A. Martínez[1] and E.M. Garzón[1]

[1]*Informatics Dpt., Agrifood Campus of International Excellence (ceiA3), University of Almería, Ctra. Sacramento s/n. La Cañada de San Urbano, 04120 Almería, Spain* gloriaortega@ual.es, juanjonrg@gmail.com, jmartine@ual.es, gmartin@ual.es

[2]*Vilnius University, Universiteto str. 3, Vilnius, Lithuania,* Ernestas.Filatovas@mif.vu.lt

Abstract Large-scale multi-objective optimization problems with many criteria have a need for Ultrascale computing to solve them within a reasonable amount of time. Current evolutionary multi-objective optimization algorithms as well as their parallel versions being designed for this goal do not consider energy consumption and savings – only the execution time is treated as main criterion of an algorithm efficiency.

In this research we focus on the most computationally expensive part of the state-of-the-art evolutionary NSGA-II algorithm – non-dominated sorting procedure – which consumes most of the computational burden. The impact of CPU and GPU workloads on power consumption and energy efficiency is experimentally investigated for our recently developed parallel versions of the Fast Non-Dominated Sorting (FNDS) procedure when solving the multi-objective optimization problems. Estimation of the balance between energy consumption and performance is also carried out, and the recommendation of usage of the developed parallel version of non-dominated sorting procedure depending on the specific platform and architecture are provided as well. The results of this research will help to design the NSGA-II based energy aware algorithms for solving large-scale multi-objective optimization problems.

Keywords: Multi-objective Optimization, Evolutionary Algorithms, NSGA-II, Green Computing, High-Performance Computing

1. Introduction

The main goal of Multi-Objective Optimization (MOO) is to provide the set of solutions that determine the Pareto front. Due the complexity of the majority of MOO problems it is impossible to obtain the exact Pareto front, therefore Evolutionary Multi-objective Optimization (EMO) algorithms are commonly-used to approximate the Pareto front [2, 15, 18]. There are many works where EMO algorithms are successfully applied for solving relatively the small multi-objective optimization problems with small number of objectives, where the population size do not exceed 1,000 individuals. It is obvious that when the number of objectives increases very large populations should be used in EMO algorithms to represent the Pareto front informatively. However, is such cases computational load of the EMO algorithms strongly increases. Recently, several parallel versions of EMO algorithms have been developed, which can exploit the resources of Ultrascale computing platforms, with the unique goal of to improve the performance of these methods [9, 13, 14].

Nowadays, the energy efficiency is another target in several computational contexts. Specifically, in Ultrascale computation large data centres the energy consumption has a strong impact in the management cost and reliability. As consequence, an intensive effort is being developed to design approaches for improving the energy/power efficiency of computational

*This work has been partially supported by the Spanish Ministry of Science throughout project TIN15-66680, by J. Andalucía through projects P12-TIC-301 and P11-TIC7176, and by the European Regional Development Fund (ERDF)

devices and platforms [1, 7, 10, 12, 16]. Currently, energy costs represent a relevant share of the total costs of High Performance Computing (HPC) systems. They include several kinds of processing units, such as CPU cores and GPU, whose energy consumption depends on the kind of processing which is performing. The energy consumed by a computational process can be get as the product of its run-time and the average electrical power during its execution. It plays a key role for evaluating the efficiency of the systems in terms of performance and power/energy.

In this work we focus on development of energy-aware EMO algorithms. The majority of the EMO approaches in the literature are based on Pareto dominance ranking, which is computed by a Non-Dominated Sorting (NDS) procedure [3, 4, 5, 8, 19], etc. As shown in [9, 14], it consumes most of the computational burden of the EMO algorithm.

NSGA-II, as a representative EMO algorithm based on NDS procedure, is analysed in this work in terms of energy efficiency. We have developed three parallel versions of NSGA-II which accelerate the NDS procedure on: (1) a multicore processor, (2) a GPU card and (3) both. Our main goal is to define the computational resources (number of cores and/or GPU) what optimize the energy efficiency. Therefore, for every combination platform-resources/problem-size the energy E is evaluated for a analysed test problem. The analysis of the results allows us: (1) to evaluate the energy consumption of NSGA-II when it is computed on several computational resources and test problems and (2) to select appropriate resources of platforms to compute NSGA-II, according to the number of individuals in the populations (N), number of objective functions (M) and number of CPU cores/ GPU cards of the available computational platforms. Therefore, this work results a methodology to optimize the energy consumption of the NSGA-II on platforms with several CPU cores and GPU cards.

2. Multi-objective optimization problems

A multi-objective minimization problem is formulated as follows [11]:

$$\min_{\mathbf{x}\in\mathbf{S}} \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})]^T \tag{1}$$

where $\mathbf{z} = \mathbf{f}(\mathbf{x})$ is an *objective vector*, defining the values for all objective functions $f_1(\mathbf{x})$, $f_2(\mathbf{x})$, $\ldots$, $f_M(\mathbf{x})$, $f_i : \mathbb{R}^V \to \mathbb{R}$, $i \in \{1, 2, \ldots, M\}$, where $M \geq 2$ is the number of objective functions; and $\mathbf{x} = (x_1, x_2, \ldots, x_V)$ is a vector of variables (*decision vector*) and V is the number of variables $\mathbf{S} \subset \mathbb{R}^V$ is *search space*, which defines all feasible decision vectors.

A decision vector $\mathbf{x}' \in \mathbf{S}$ is a *Pareto optimal solution* if $f_i(\mathbf{x}') \leqslant f_i(\mathbf{x})$ for all $\mathbf{x} \in \mathbf{S}$ and $f_j(\mathbf{x}') < f_j(\mathbf{x})$ for at most one j. Objective vectors are defined as optimal if none of their elements can be improved without worsen at least one of the other elements. An objective vector $\mathbf{f}(\mathbf{x}')$ is Pareto optimal if the corresponding decision vector $\mathbf{x}'$ is Pareto optimal. The set of all the Pareto optimal decision vectors is called the *Pareto set.* The region defined by all the objective function values for the Pareto set points is called the *Pareto front*.

For two objective vectors $\mathbf{z}$ and $\mathbf{z}'$, $\mathbf{z}'$ *dominates* $\mathbf{z}$ (or $z' \succ z$) if $z'_i \leqslant z f_i$ for all $i = 1, \ldots, M$ and there exists at most one j such that $z'_j < z_j$. In EMO algorithms, the subset of solutions in a population whose objective vectors are not dominated by any other objective vector is called the *non-dominated set*, and the objective vectors are called the *non-dominated objective vectors*. The main aim of the EMO algorithms is to generate well-distributed non-dominated objective vectors as close as possible to the Pareto front.

NSGA-II [4] is the most widely-used and well-known EMO algorithm for approximating the Pareto front that is based on NDS. Thus, it is selected to analyse the energy efficiency of EMO algorithms when different number of CPU cores and/or GPU cards are exploited.

The steps of NSGA-II are described in Algorithm 1. The Step 2 of the algorithm is devoted to the NDS procedure which is the most computationally expensive in the NSGA-II.

Step 1: Generate a random initial population P_0 of size N.
Step 2: Sort the population to different non-domination levels (fronts) using, and assign each individual a fitness equal to its non-domination level (1 is the best level).
Step 3: Create an offspring population of size N using binary tournament selection, recombination and mutation operations (parents with larger crowding distance are preferred if their non-domination levels are the same).
Step 4: Combine the parent and the offspring populations and create a population R.
Step 5: Reduce the population R to the population P of size N: sort the population R into different non-dominated fronts; fill the population P with individuals from population R starting from the best non-dominated front until the size of P is equal to N; if all the individuals in a front cannot be picked fully, calculate a crowding distance and add individuals with the largest distances into the population P.
Step 6: Check if the termination criterion is satisfied. If yes, go to Step 7, else return to Step 2.
Step 7: Stop.

Algorithm 1: NSGA-II.

3. Energy efficiency evaluation of NSGA-II

In this section, a preliminary study of NSGA-II energy consumption is carried out. The widely-used DTLZ2 benchmark problem [6] has been considered as it suits well for experimental evaluation in this context. The test platform has been a Bullx R421-E4 Intel Xeon E5 2620v2 (12 CPU-cores, 64-GB RAM and 1-TB HDD) with a NVIDIA K80 (Kepler GK210) GPU. The NSGA-II algorithm has been implemented in MATLAB and it can call to several parallel NDS routines (multicore or GPU version). The multicore version is based on C++ and Pthreads and the GPU version is developed on CUDA 7.5. All the test have been executed with a number of generations equal to 100. The size of the population ranges from 200 to 5,000. The power measurements on CPU (or GPU) have been sampled by an external monitor program that queries RAPL register using the PAPI library [17] (or NVML interface).

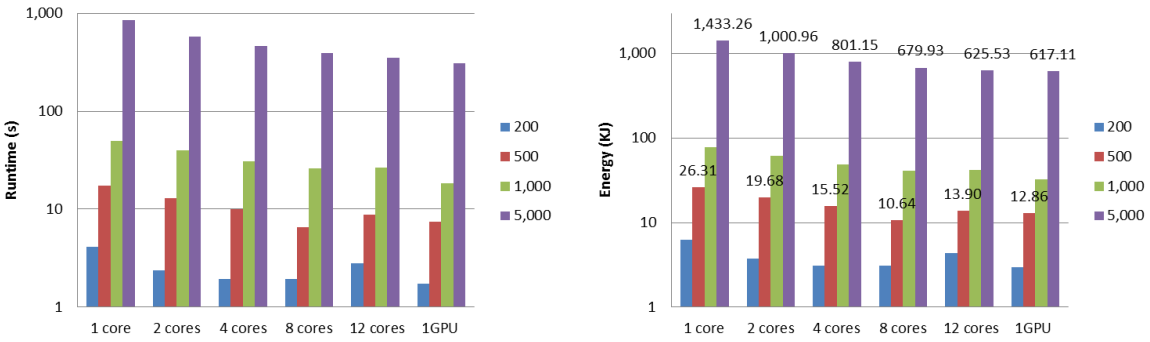


Figure 1. Runtime in seconds (left) and the consumed energy in KJ (right) when the NSGA-II is executed with several populations sizes (200, 500, 1,000, and 5,000) for solving the DTLZ2 problem on several resources of Bullx R421-E4 Intel Xeon E5 2620v2 (12 CPU-cores) with a NVIDIA K80 GPU.

Figure 1 shows the runtime (in seconds) and the consumed energy (KJ) when the NSGA-II is executed with different population sizes solving the DTLZ2 problem with 5 objective functions on several resources of the test computational platform (1-12 cores, 1GPU). As it can be observed, the runtime and energy strongly increase with the size of the population. Moreover, runtime and energy vary with the selection of the resources. For instance, when the population size is 5,000, the optimal execution is on the GPU in terms of energy consumption.

When the problem size is 500, the subset of 8 cores minimizes the energy consumption. So, for a specific platform (with a set of available resources) and several instances of problems of the NSGA-II algorithm, it is necessary to define approaches to identify what is the subset of resources to optimize the energy consumption.

References

[1] J. Carretero, S. Distefano, D. Petcu, D. Pop, T. Rauber, G. Rünger, and D. Singh. Energy-efficient algorithms for ultrascale systems. *Supercomputing frontiers and innovations*, 2(2), 2015.

[2] K. Deb. *Multi-objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, 2001.

[3] K. Deb and H. Jain. An improved NSGA-II procedure for many-objective optimization, Part I: Solving problems with box constraints. *KanGAL Report*, (2012009), 2012.

[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[5] K. Deb, J. Sundar, N. Udaya, and S. Chaudhuri. Reference point based multi-objective optimization using evolutionary algorithms. *International Journal of Computational Intelligence Research*, 2(3):273–286, 2006.

[6] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multi-objective optimization test problems. In *IEEE World Congress on Computational Intelligence*, pages 825–830, 2002.

[7] Y. Deng, Y. Hu, X. Meng, Y. Zhu, Z. Zhang, and J. Han. Predictively booting nodes to minimize performance degradation of a power-aware web cluster. *Cluster Computing*, 17(4):1309–1322, 2014.

[8] E. Filatovas, O. Kurasova, and K. Sindhya. Synchronous R-NSGA-II: An Extended Preference-Based Evolutionary Algorithm for Multi-Objective Optimization. *Informatica*, 26(1):33–50, 2015.

[9] S. Gupta and G. Tan. A scalable parallel implementation of Evolutionary Algorithms for Multi-Objective optimization on GPUs. In *IEEE Congress on Evolutionary Computation*, pages 1567–1574. IEEE, 2015.

[10] Y. Kang, W. Choi, B. Kim, and J. Kim. On tradeoff between the two compromise factors in assigning tasks on a cluster computing. *Cluster Computing*, 17(3):861–870, 2014.

[11] K. Miettinen. *Nonlinear Multiobjective Optimization*, volume 12 of *International Series in Operations Research and Management Science*. Kluwer Academic Publishers, Dordrecht, 1999.

[12] S. Mittal and J.S. Vetter. A Survey of Methods for Analyzing and Improving GPU Energy Efficiency. *ACM Computing Surveys*, 47(2):19:1–19:23, August 2014.

[13] D. Sharma and P. Collet. Implementation techniques for massively parallel multi-objective optimization. In *Massively Parallel Evolutionary Computation on GPGPUs*, pages 267–286. Springer, 2013.

[14] C. Smutnicki, J. Rudy, and D. Żelazny. Very fast non-dominated sorting. *Decision Making in Manufacturing and Services*, 8(1-2):13–23, 2014.

[15] E. Talbi. *Metaheuristics: from Design to Implementation*, volume 74. John Wiley & Sons, 2009.

[16] Y. Tian, C. Lin, and K. Li. Managing performance and power consumption tradeoff for multiple heterogeneous servers in cloud computing. *Cluster Computing*, 17(3):943–955, 2014.

[17] V.M. Weaver, M. Johnson, K. Kasichayanula, J. Ralph, P. Luszczek, D. Terpstra, and S. Moore. Measuring Energy and Power with PAPI. In *Proceedings of the 2012 41st International Conference on Parallel Processing Workshops*, ICPPW '12, pages 262–268, Washington, DC, USA, 2012. IEEE Computer Society.

[18] Q. Zhang and H. Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

[19] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Zurich, Switzerland, 2001.

Proceedings of GOW'16, pp. 29 – 32.

Locating a Facility with the Partially Probabilistic Choice Rule*

José Fernández[1], Boglárka G.-Tóth[2], Juana López Redondo[3] and Pilar Martínez Ortigosa[3]

[1] *University of Murcia, Murcia, Spain*
josefdez@um.es

[2] *Budapest University of Technology and Economics, Budapest, Hungary*
bog@math.bme.hu

[3] *University of Almería, Agrifood Campus of International Excellence, ceiA3, Almería, Spain*
{jlredondo,ortigosa}@ual.es

Abstract Locating a facility is a strategic decision which usually requires a massive investment. In a competitive environment, where other facilities offering the same goods or products exist, the location and quality of a facility may determine its success or failure. The profit that a facility obtains largely depends on the market share that it captures, and this depends on the attraction that the customers feel for the facility and on the patronizing behavior of the customers. In this work we assume that the former can be appropriately computed, and we concentrate on the latter. In this paper we consider that a customer only patronizes those facilities for which he/she feels an attraction greater than or equal to a threshold value. Implicitly, this implies that there may be some unmet demand. We apply this new rule to the problem of locating a single new facility in the plane, where the quality of the new facility to be located is also considered as a variable of the problem to be determined. The threshold value provokes discontinuities in the objective function of the problem. Both exact and heuristic methods are proposed to solve the location problem with the new choice rule.

Keywords: Continuous location, Competition, Customer choice rule, Interval branch-and-bound method, Patronizing behavior, Evolutionary Algorithms, Computational study

1. Introduction

The patronizing behavior of customers is one of the core inputs in many business and economic indicators. This is the case for the estimation of the market share captured by the facilities in a competitive environment. When there exist several facilities offering the same product, the way in which customers decide to spend their buying power among them may determine the success or the failure of the facilities. Consumer behavior is a function of the attraction that consumers feel for the facilities. This attraction is the result of several factors, but the two most important forces are the location of the facilities and their quality: the closer the facility to the customer and the higher its quality, the greater the attraction of the customer towards the facility. In fact, when a company wants to enter a market or to expand its presence by opening new facilities, both the location and the quality of the new facilities are chosen so as to maximize the revenues obtained by the chain, which largely depend on the market share that it captures.

*This research has been supported by grants from the Spanish Ministry of Economy and Competitiveness (MTM2015-70260-P, TIN2015-66680-C2-1-R), Fundación Séneca (The Agency of Science and Technology of the Region of Murcia, 19241/PI/14), Junta de Andalucía (P11-TIC7176 and P12-TIC301), in part financed by the European Regional Development Fund (ERDF). This work is also funded by the Hungarian National Research, Development and Innovation Office âĂŞ NKFIH, OTKA grant PD115554 and also by the project ICT COST Action TD1207 (EU). Juana López Redondo is a fellow of the Spanish 'Ramón y Cajal' contract program.

The new customer choice rule introduced here is an extension of the classical probabilistic choice rule. According to the new rule, called partially probabilistic choice rule, a customer, in order to satisfy his/her demand, only patronizes those facilities for which he/she feels an attraction greater than or equal to a threshold value, and the demand is split among them proportionally to their attraction. employed in literature, the threshold value implicitly implies that there may be some unmet demand.

The influence of the choice rule in the location of competitive facilities has been analyzed. In particular, the problem of locating a single new facility in the *plane* has been considered. The corresponding location problem for profit maximization has been formulated, and an exact interval branch-and-bound method (iB&B), as well as a heuristic evolutionary algorithm (UEGO) have been proposed to cope with the problem.

2. The partially probabilistic location problem

We will assume the following particular scenario (see [1] for more details). A single new facility is going to be located in a given region of the plane by a chain. There already exist m facilities around selling the same goods or product; k of those facilities may belong to the locating chain. The demand to be served, known and fixed, is concentrated at some demand points, whose locations are known. The location and quality of the existing facilities is also known. The attraction function of a demand point towards a facility is modeled as perceived quality divided by perceived distance. The maximization of the *profit* obtained by the chain after the location of the new facility is the objective to be achieved, to be understood as the income due to the market share captured by the chain minus its operational costs. The aim is to find both the location and the quality of the new facility to be located.

It is important to have effective methods for solving the location problems. Usually competitive location problems lead to hard-to-solve global optimization problems. The challenge of the location problem with the partially probabilistic choice rule is that it also presents discontinuities as a consequence of the threshold value (see Figure 1). For the problem at hand, the usefulness of the interval branch-and-bound algorithm iB&B [1] and the evolutionary algorithm UEGO [5, 6] has been investigated.

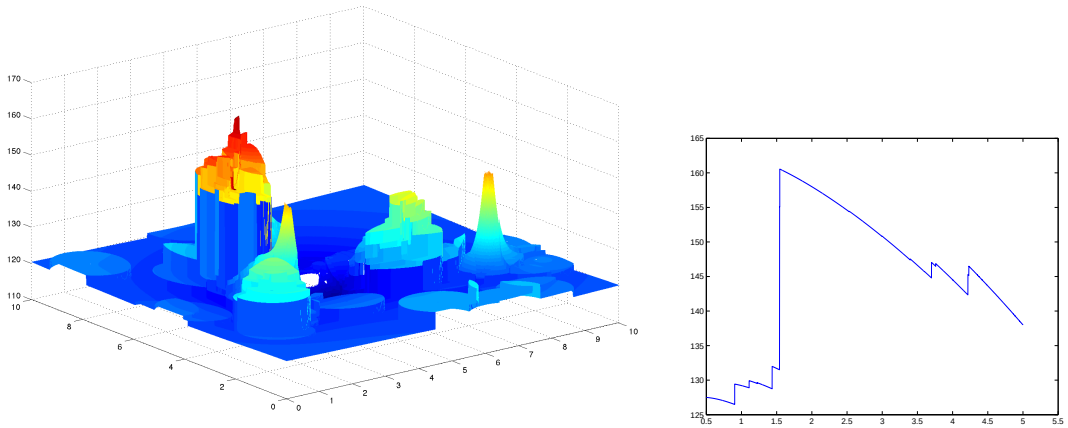


Figure 1. Projections of the objective function of an instance with setting $(i_{max} = 71, j_{max} = 5, k = 2)$ when the threshold value is equal to 2. On the left, when $\alpha = 1.545898$. On the right, when $(x_1 = 3.989257, x_2 = 7.065429)$

Interval B&B methods have been successfully applied to solve other location problems (see for instance [2, 8] and the references therein). In particular, in [1] such a method was applied to solve the corresponding probabilistic problem. A similar method can handle the partially probabilistic model, thanks to the use of the interval tools employed to compute the bounds. However, only the discarding tests described in [1] which do not make use of the differentia-

Table 1. UEGO results for problems with $n = 10000$ demand points.

$(m,k)_{prob}$	Time	Distance	Minimum	Average	Maximum	Deviation
$(100,25)_1$	763.1	0.010	14026.9	14026.9	14026.9	0.006
$(100,25)_2$	631.9	0.006	13059.7	13059.7	13059.7	0.012
$(100,25)_3$	629.2	0.046	14092.3	14092.4	14092.4	0.043
$(100,25)_4$	480.3	0.145	15650.1	15650.4	15650.6	0.165
$(100,25)_5$	534.0	0.002	12688.1	12688.1	12688.1	0.013
average	607.7	0.042	13903.4	13903.5	13903.5	0.048
$(100,50)_1$	729.4	0.002	24672.3	24672.3	24672.3	0.000
$(100,50)_2$	857.6	0.009	27241.9	27241.9	27241.9	0.012
$(100,50)_3$	435.3	0.010	27092.0	27092.1	27092.1	0.038
$(100,50)_4$	678.0	0.081	24517.4	24517.4	24517.4	0.013
$(100,50)_5$	837.9	0.000	29174.6	29174.6	29174.6	0.002
average	707.6	0.020	26539.6	26539.6	26539.7	0.013
$(200,50)_1$	527.3	0.000	12684.2	12684.2	12684.2	0.000
$(200,50)_2$	660.7	0.051	12102.1	12102.3	12102.3	0.074
$(200,50)_3$	679.8	0.108	12447.8	12447.9	12448.0	0.072
$(200,50)_4$	529.6	0.220	12572.6	12572.8	12572.9	0.110
$(200,50)_5$	691.1	0.105	14112.2	14112.2	14112.3	0.051
average	617.7	0.097	12783.8	12783.9	12784.0	0.061
$(200,100)_1$	903.4	0.098	25274.8	25274.8	25274.9	0.007
$(200,100)_2$	912.4	0.128	26845.2	26845.2	26845.3	0.037
$(200,100)_3$	884.9	0.000	24426.4	24426.4	24426.4	0.000
$(200,100)_4$	626.4	0.003	23853.1	23853.1	23853.1	0.001
$(200,100)_5$	594.2	0.059	27100.8	27100.8	27100.9	0.015
average	784.3	0.058	25500.1	25500.1	25500.1	0.012

bility of the objective function can be used for the new problem, namely, the cut-off and the feasibility tests. In addition to this, inclusion functions for piece-wise functions have had to be designed.

UEGO is a general heuristic able to solve many global optimization problems. In fact, it has also been applied to other competitive location problems as well (see for instance [4]). Only the local search procedure used within UEGO needs to be adapted for each particular problem. In view of its success in solving different competitive location problems, UEGO has also been adapted to the problem with the partially probabilistic choice rule. It was first implemented using a Weiszfeld-like algorithm as a local search, following the lines in [5]. However, the obtained results were not as good as expected. It is with the stochastic hill climber SASS [7] that UEGO can solve the problem effectively and efficiently as we will see. The use of SASS as the local optimizer within UEGO has also worked fine for other location problems with discontinuities (see for instance [3]).

3. Computational results

A comprehensive computational study has been carried out. In all, 80 problems have been solved, i.e., 20 instances for 4 different number of demand points n =500, 1000, 5000 and 10000.

The exact iB&B can manage instances with up to 1000 demand points without difficulties, but it starts experiencing problems with instances with 5000 demand points. This clearly shows the difficulty of the problem at hand, and the need for a heuristic method to cope with large-size problems. Regarding UEGO, it is quite robust. It has 100% success in all the problems solved by iB&B, i.e, in all the problems and in all the runs, the solution provided by UEGO was always included in one of the solution boxes provided by iB&B. In fact, the minimum, average and maximum objective function values obtained by UEGO are always

about the same. Not only is the objective function value the same in all the runs, but the solution point offered by the algorithm is also the same. And this using a tiny fraction of the CPU time employed by iB&B.

On the other hand, UEGO is a reliable heuristic method, able to solve problems with up to 10000 demand points without difficulties, solving them in less than 13 minutes. Table 1 shows the results obtained by UEGO for the 20 instances with 1000 demand points and different values for m and k. The actual settings employed can be seen in Table 1. Notice that UEGO is a stochastic algorithm, and hence each run may lead to a different solution. To take this fact into account, each problem has been solved 5 times, and the following values have been computed: the average CPU time employed in solving the problem in the 5 runs, the maximum Euclidean distance between any pair of solutions, the minimum, average and maximum objective function values of the solutions and the corresponding standard deviation.

4. Summary

In this work, a new location problem is studied. It has been solved by means of an interval Branch & Bound method and an evolutionary algorithm. From the computational study, it can be concluded that iB&B can solve problems with up to 1000 demand points exactly. For larger instances, the use of the evolutionary algorithm UEGO is recommended, as it is a reliable and robust method.

References

[1] J. Fernández, B. Pelegrín, F. Plastria, and B. Tóth. Solving a Huff-like competitive location and design model for profit maximization in the plane. *European Journal of Operational Research*, 179(3):1274–1287, 2007.

[2] J. Fernández and B. Pelegrín. Using interval analysis for solving planar single-facility location problems: new discarding tests. *Journal of Global Optimization*, 19(1):61–81, 2001.

[3] J.L. Redondo, J. Fernández, A.G. Arrondo, I. García, and P.M. Ortigosa. A two-level evolutionary algorithm for solving the facility location and design (1|1)-centroid problem on the plane with variable demand. *Journal of Global Optimization*, 56(3):983–1005, 2013.

[4] J.L. Redondo, J. Fernández, A.G. Arrondo, I. García, and P.M. Ortigosa. Fixed or variable demand? Does it matter when locating a facility? *Omega*, 40(1):9–20, 2012.

[5] J.L. Redondo, J. Fernández, I. García, and P.M. Ortigosa. A robust and efficient global optimization algorithm for planar competitive location problems. *Annals of Operations Research*, 167(1):87–106, 2009.

[6] J.L. Redondo, J. Fernández, I. García, and P.M. Ortigosa. Parallel algorithms for continuous competitive location problems. *Optimization Methods & Software*, 23(5):779–791, 2008.

[7] F.J. Solis and R.J.B. Wets. Minimization by random search techniques. *Mathematics of Operations Research*, 6(1):19–30, 1981.

[8] B. Tóth and J. Fernández. *Interval methods for single and bi-objective optimization problems - applied to competitive facility location problems*. Lambert Academic Publishing, Saarbrücken, 2010.

Proceedings of GOW'16, pp. 33 – 36.

On Regular Refinement of Unit Simplex by just Visiting Grid Points*

L.G. Casado[1], J.M.G. Salmerón[1], B.G.-Tóth[2], E.M.T. Hendrix[3] and I. García[3]

[1] *University of Almería, Spain,* leo@ual.es, josemanuel@ual.es

[2] *Budapest University of Technology and Economics, Hungary,* bog@math.bme.hu

[3] *Universidad de Málaga, Spain,* eligius@uma.es, igarciaf@uma.es

Abstract In Branch and Bound (BnB) algorithms, the branching rule plays an important rule in order to reduce the number of evaluated sub-problems and points. Recent studies addressed the unit simplex refinement with regular simplices. When the achieved accuracy is related on the distance among sampled points, a common method is to stop the refinement when the size of a sub-problem, measured as the largest distance among its evaluated vertices, is smaller or equal to a given precision. The worst case was studied: the complete tree is generated, which is determined by the size of its leafs. In such scenario, 2USC (two-uniform-simplex cover) on a mesh grid showed a better performance than LEB (longest edge bisection) in terms of number of simplices and vertices evaluations. Due to the overlapping of sub-simplices, termination criterion in 2USC can be relaxed but the largest precision to cover the mesh grid depends on the dimension. This work focus on the study of the 2MUSC division method, where regular sub-simplices may have different orientation, in order to know if 2MUSC can cover the mesh grid for larger precisions than 2USC does.

Keywords: Unit simplex, Refinement Regular division, Mesh grid.

1. Introduction

The unit simplex is defined as

$$S_1 = \{x \in \mathbb{R}^d \mid \sum_{j=1}^{d} x_j = 1;\ x_j \geq 0,\ \forall j\}. \tag{1}$$

This search region is widely used for mixture design problems, equilibria and fixed point problems with application in economy, game theory and nonlinear systems of equations [10], co-positivity detection of matrices [1], stability in biological population dynamics [9] and others [7]. We studied these problems with quadratic constraints for practical mixture design in lubricant industry [6] and for the design of fat blends in food production [3, 5].

The main question is how to design a refinement strategy for a unit simplex such that the number of evaluated sub-problems and vertices is minimized taking explicitly the equidistant grid with relative mesh size α into account. The size of a sub-problem is defined as $w(S)$, that is the width of the simplex S defined as the length of its longest edge. We assume a user given relative accuracy ϵ. A sub-problem S is not further divided when $w(S) \leq \epsilon w(S_1)$. The BnB algorithm returns the set of final and not rejected sub-problems. In this way, the problem solution is guaranteed to be in one of the final sub-simplices with a size less than $\epsilon w(S_1)$.

Two points x, y on the α-grid over the unit simplex are neighbour if $\exists i, j, i \neq j$ such that, $y_i = x_i + \alpha$, $y_j = x_j - \alpha$ and $y_k = x_k$, for $k \neq i, j$. It is a challenge to find a refinement in

*This work has been funded by grant from the Spanish Ministry (TIN2015-66688), Junta de Andalucía (P11-TIC-7176) and in part financed by the European Regional Development Fund (ERDF).

the BnB algorithm that only generates sample points as vertices of the sub-sets that coincide with grid points on S_1. Given an accuracy ϵ, the largest value for the mesh size is $\alpha = \frac{1}{G}$, with $G = \lceil \frac{1}{\epsilon} \rceil$ parts on each axis and $G+1$ grid point per axis, whereas two of them represent vertices of S_1. For the unit simplex, the number of α-grid points is given by [3]:

$$\sum_{k=1}^{d-1} \left(\begin{array}{c} G+1 \\ k \end{array} \right) \left(\begin{array}{c} d-2 \\ k-1 \end{array} \right). \tag{2}$$

According to [2], one can take the α mesh size to guide the refinement using regular simplices of same size and orientation (2USC) and the improvement in the number of evaluated sub-problems and vertices is significant when is compared to longest edge bisection (LEB). The drawback of the 2USC refinement on the grid is that the smallest G value of the accuracy needed to completely visit the grid depends on the dimension. This work studies the 2MUSC (2 Mixed Unit Simplex Cover) refinement of the unit simplex which uses the so-called 2USC and 2∇USC refinement, see [4]. 2∇USC generates regular sub-simplices of different size and orientation on the grid.

2. 2USC on grid

The two-uniform-simplex-cover refines a regular simplex in d overlapping regular sub-simplices with two sub-simplices per edge (see Figures 1 and 2). A reduction factor $\beta \geq \frac{d-1}{d}$ guarantees the covering of the refined simplex. Considering that sub-simplices overlap, the refinement method only evaluates an overlapped region once, see [4].

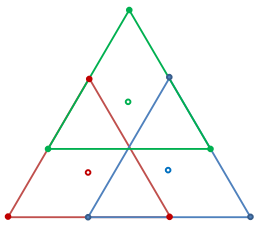


Figure 1. 2USC in $\mathbb{R}^3$. Dots represent the vertices and the centre of a sub-simplex.

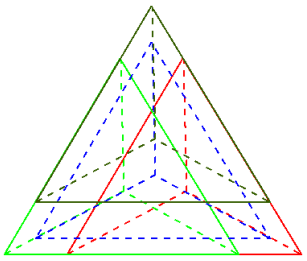


Figure 2. 2USC in $\mathbb{R}^4$.

The main characteristics of the 2USC on α-grid are the following:

- Let S^ℓ be a sub-simplex leaf of the d-ary tree. The general termination criterion $w(S) \leq \epsilon w(S_1)$ can be relaxed to $w(S^\ell) = \alpha(d-1)w(S_1) \leq \epsilon w(S_1)$.

- The sequence of sub-simplex size αk_i on each level i of the tree can be determined taking into account that the size of every sub-simplex should be a multiple of α:

$$\begin{aligned} w(S_1 = S^0) &= \alpha k_0 & k_0 &= G \\ w(S^1) &= \alpha k_1, & k_1 &= \left\lceil k_0 \frac{d-1}{d} \right\rceil \\ w(S^2) &= \alpha k_2, & k_2 &= \left\lceil k_1 \frac{d-1}{d} \right\rceil \\ &\dots \\ w(S^{\ell-2}) &= \alpha k_{\ell-2}, & k_{\ell-2} &= \left\lceil k_{\ell-3} \frac{d-1}{d} \right\rceil \\ w(S^{\ell-1}) &= \alpha k_{\ell-1}, & k_{\ell-1} = d &= \left\lceil k_{\ell-2} \frac{d-1}{d} \right\rceil \\ w(S^{\ell}) &= \alpha k_{\ell}, & k_{\ell} = d-1 &= \left\lceil k_{\ell-1} \frac{d-1}{d} \right\rceil . \end{aligned} \tag{3}$$

- The minimum G value to force covering of the α-grid by the set of generated vertices in 2USC refinement depends on the dimension d: $G \geq (d-1)^2$.

3. 2MUSC on grid

2-mixed-uniform-simplex-cover refinement (2MUSC) combining 2USC and 2∇USC in the refinement of the unit simplex. The 2∇-uniform-simplex-cover (2∇USC) introduced in [4] refines a regular simplex in $d+1$ regular sub-simplices. In contrast to 2USC, d of them have the same orientation using a reduction factor $\frac{d-2}{d-1} \leq \rho < \frac{d-1}{d}$ and one non-overlapped inverted sub-simplex has a reduction factor $\varrho = d(1-\rho) - 1$. Figure 3 shows an example of 2∇USC for $d = 3$, where $\rho = \varrho = 1/2$. Figure 4 shows an example for $d = 4$ where $\rho = 2/3$ and $\varrho = 1/3$.

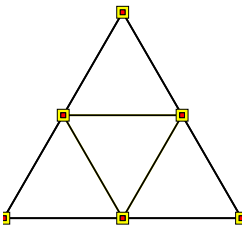


Figure 3. 2∇USC and α-grid with $G = 2$ and $d = 3$. Big boxes are α-grid points and little boxes are vertices of simplices.

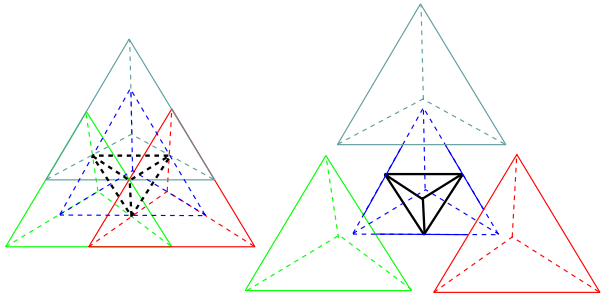


Figure 4. 2∇USC for $d = 4$.

In 2∇USC, the sub-simplices overlap less than in 2USC, but the sub-simplices are not equally oriented and sized. 2∇USC requires a delicate choice of the reduction factor ρ. In order to have a value for ρ being a multiple of α, ρ should be greater or equal to $(d-1)/d$. In that case, the reduction factor of the inverted simplex $\varrho \leq 0$. This means that the inverted sub-simplex does not exist and 2∇USC reduces to 2USC. Therefore, 2MUSC uses 2USC or 2∇USC depending on the current simplex, either a reduction factor ρ or β is applied to force vertices of the generated sub-simplices to be on the α-grid. The main characteristics of the 2MUSC on α-grid are the following:

- Assuming that last refinement is 2∇USC, the termination criterion can be relaxed to $w(S^{\ell}) = \alpha(d-2)w(S_1)$.

- The sequence of sub-simplex size αr_i on each level i of the tree are determined by:

$$
\begin{aligned}
& w(S^0) = \alpha r_0 \qquad r_0 = G \\
& w(S^1) = \alpha r_1, \qquad r_1 = \left\lceil r_0 \frac{d-2}{d-1} \right\rceil \begin{cases} 2\nabla\text{USC, if } \left\lceil r_0 \frac{d-2}{d-1} \right\rceil < r_0 \frac{d-1}{d} \\ 2\text{USC, otherwise.} \end{cases} \\
& w(S^2) = \alpha r_2, \qquad r_2 = \left\lceil r_1 \frac{d-2}{d-1} \right\rceil \begin{cases} 2\nabla\text{USC, if } \left\lceil r_1 \frac{d-2}{d-1} \right\rceil < r_1 \frac{d-1}{d} \\ 2\text{USC, otherwise.} \end{cases} \\
& \quad \dots \\
& w(S^{l-1}) = \alpha r_{l-1}, r_{l-1} = \left\lceil r_{l-2} \frac{d-2}{d-1} \right\rceil \begin{cases} 2\nabla\text{USC, if } \left\lceil r_{l-2} \frac{d-2}{d-1} \right\rceil < r_{l-2} \frac{d-1}{d} \\ 2\text{USC, otherwise.} \end{cases} \\
& w(S^l) = \alpha r_l, \qquad r_l = \left\lceil r_{l-1} \frac{d-2}{d-1} \right\rceil (2\nabla\text{USC}),
\end{aligned}
\tag{4}
$$

  where $r_{l-1} = d-1$ and $r_l = d-2$.
- Due to the existence of different sequences depending on d and G, the minimum G value to force covering of the α-grid has still to be determined.

4. Conclusions

This paper presentes the 2MUSC on grid refinement for the unit simplex, with less overlap than 2USC on grid. Numerical results and the minimum number of segments G per axis to cover the α-grid will be outlined at GOW'16. The evaluation of these refinement methods for a real problem on a BnB algorithm will be also discussed in GOW'16 [8].

References

[1] Immanuel M. Bomze. On standard quadratic optimization problems. *Journal of Global Optimization*, 13(4):369–387, 1998.

[2] L. G. Casado, E. M. T. Hendrix, J.M.G. Salmerón, B. G.-Tóth, and I. García. On grid aware refinement in branch and bound. *Submitted to Journal of Global Optimization*.

[3] Leocadio G. Casado, Eligius M. T. Hendrix, and Inmaculada García. Infeasibility spheres for finding robust solutions of blending problems with quadratic constraints. *Journal of Global Optimization*, 39(2):215–236, 2007.

[4] B. G.-Tóth, E. M. T. Hendrix, L. G. Casado, and I. García. On refinement of the unit simplex using regular simplices. *Journal of Global Optimization*, 64(2):305–323, 2015.

[5] Eligius M. T. Hendrix, Leocadio G. Casado, and Inmaculada García. The semi-continuous quadratic mixture design problem: Description and branch-and-bound approach. *European Journal of Operational Research*, 191(3):803 – 815, 2008.

[6] E.M.T. Hendrix and J. Pintér. An application of Lipschitzian global optimization to product design. *Journal of Global Optimization*, 1:389–401, 1991.

[7] E. de Klerk, D. den Hertog, and G Elabwabi. On the complexity of optimization over the standard simplex. *European Journal of Operational Research*, 191(3):773–785, 2008.

[8] J.M.G. Salmerón, L. G. Casado, E. M. T. Hendrix, and J.F.R. Herrera. On regular simplex division in solving blending problems. In *Proceedings of GOW'16: XIII Global Optimization Workshop*, 2016.

[9] G. J. Still and F. Ahmed. Quadratic maximization on the unit simplex: structure, stability, genericity and application in biology. Memorandum 2034, Department of Applied Mathematics, University of Twente, Enschede, February 2014.

[10] Zaifu Yang. *Computing Equilibria and Fixed Points*, volume 21 of *Theory and Decision Library C*. Springer, 1999.

Proceedings of GOW'16, pp. 37 – 40.

Metabolic Pathway Analysis Using Nash Equilibrium

Angelo Lucia[1] and Peter A. DiMaggio[2]

[1]*Department of Chemical Engineering, University of Rhode Island, Kingston, RI 02881, USA,* lucia@egr.uri.edu

[2]*Department of Chemical Engineering, Imperial College London, London SW7 2AZ, UK,* p.dimaggio@imperial.ac.uk

Abstract A novel approach to metabolic network analysis using a Nash equilibrium formulation is proposed. Enzymes are considered to be players in a multi-player game in which each player attempts to minimize the dimensionless Gibbs free energy associated with the biochemical reaction(s) it catalyzes subject to elemental mass balances. Mathematical formulation of the metabolic network as a set of nonlinear programming (NLP) sub-problems and an appropriate solution methodologies are described. A small example representing part of the production cycle for acetyl-CoA is used to demonstrate the efficacy of the proposed Nash equilibrium framework and that it represents a paradigm shift in metabolic network analysis.

Keywords: Metabolic pathway analysis, Nash equilibrium, Global optimization

1. Introduction

Flux balance analysis (FBA) has been the mainstay for understanding and quantifying metabolic networks for many years. See, for example, [1, 2, 3, 4, 5, 6], and others. The basic idea behind FBA is to represent a given metabolic network at steady-state in the form of a graph with nodes that define specific biochemical reactions and fluxes that connect nodes. The constraints for the network constitute a set of under-determined steady-state linear mass balance equations. To complete the representation, a linear objective function relevant to the particular biological task at hand (e.g., maximizing the output flux [5]; minimizing the cardinality of the flux vector [4]; minimum nutrient intake; minimum ATP production minimal knockout, etc.) is selected, which together with the mass balance constraints, results in a linear programming (LP) formulation. Many variants and extensions to FBA have also been proposed over the years, including Mixed Integer Linear Programming (MILP) formulations [7], the incorporation of linearized thermodynamic constraints [8] or thermodynamic metabolic flux analysis (TMFA), dynamic FBA [9], and others. In addition to FBA, the other main approach to metabolic network analysis involves a time-dependent or dynamic formulation that incorporates chemical reaction kinetics.

Both the kinetic model and FBA approaches suffer from a number of significant modeling and biological limitations. For example, kinetic models require a large number of parameters that are not 'directly' measurable and thus must be determined by model regression. Also, all FBA-based approaches are constrained only by reaction stoichiometry, which often results in degenerate solutions that lie on the user-specified flux bounds. Additional modeling limitations stem from the inability of either approach to accurately capture the inherent complexities within a crowded cellular environment. In particular, natural competition/cooperation among enzymes exists for the pool of continuously produced metabolites and neither approach takes into consideration the population distribution and/or heterogeneity of protein-metabolite interactions that form the basis for these reactions. Moreover, from a biological perspective, these approaches also fail to model the phenotypic consequence of overproducing a given product and the subsequent regulation of overproduction in these organisms. For

instance, it is well known that excess quantities of a given protein or small molecule can lead to 'higher order' interactions, which can be non-specific or the result of adaptive evolutionary pressures that activate alternative pathways to deplete these excess pools.

2. Nash Equilibrium Approach

This paper takes a radically different approach to metabolic network analysis by formulating the problem as a Nash Equilibrium (NE) problem using first principles (i.e., conservation of mass and rigorous reaction thermodynamics). The key idea behind our proposed NE approach is to view enzymes as 'players' in a multi-player game, in which each enzyme pursues a strategy that can be quantified using a payoff or objective function. More specifically, the goal of each enzyme is to minimize the dimensionless Gibbs free energy of the biochemical reaction it catalyzes subject to appropriate elemental mass balances (i.e., conservation of mass of carbon, hydrogen, oxygen, nitrogen, phosphorous, sulfur, etc.). Thus the overall goal of the network is to find the best value of

$$\frac{G(v)}{RT} = \min \sum_{j=1}^{N} \frac{G_j(v_j)}{RT} \tag{1}$$

where $\frac{G_j}{RT}$, the dimensionless Gibbs free energy, is the objective function associated with the appropriate enzymes involved in a particular set (or number) of metabolic reactions at a given node j in the network, v is a vector of unknown metabolic fluxes, R is the gas constant and T is the temperature. The solution, called a NE point, optimizes the 'payoff' of all players simultaneously, is the best solution for all players taken together, and not necessarily best for any one player. This NE approach, in our opinion, is a more accurate representation of the evolutionary-defined competition/cooperation observed in complex metabolic pathways.

Let the unknown variables, v, be partitioned into N subsets, $v = [v_1, v_2, ..., v_N]$. Each variable partition, v_j, has n_j unknown variables and there are N total unknowns. The Nash Equilibrium formulation for an arbitrary metabolic network is different from that for FBA and is given by $j = 1, 2, ..., N$ nonlinear programming (NLP) sub-problems of the form:

$$\begin{array}{ll} \min & \frac{G_j(v_j, v_j^*)}{RT} \\ \text{subject to} & \text{conservation of mass} \\ & v_j^* \end{array} \tag{2}$$

The conservation of mass constraints are elemental mass balances and represent the flow of metabolic material in and out of any node. Finally, the vector, v_j^*, denotes the minima of all other sub-problems, $k = 1, 2, .., j-1, j+1, ..., N$. In this article the words "sub-problem" and "node" mean the same thing.

3. Example

Figure 1 is a simplified metabolic network for the production of acetyl-CoA whose Nash Equilibrium formulation and solution were determined by specifying the input flux to the network (i.e., the pyruvate flux) and breaking the acetate feedback to the acetyl-CoA pool. Initial estimates of the acetate flux and co-enzyme A (CoA) allows the NLP sub-problems associated with the following bio-chemical reaction equilibria to be solved in the order shown below.

1. Acetyl-CoA production from pyruvate and acetate:

$$C_3H_3O_3 \; + \; C_2H_3O_2 \; + \; 2C_{21}H_{32}N_7O_{16}P_3S \; \rightleftharpoons \; 2C_{23}H_{34}N_7O_{17}P_3S \; + \; CO_2 \; + \; H_2O$$

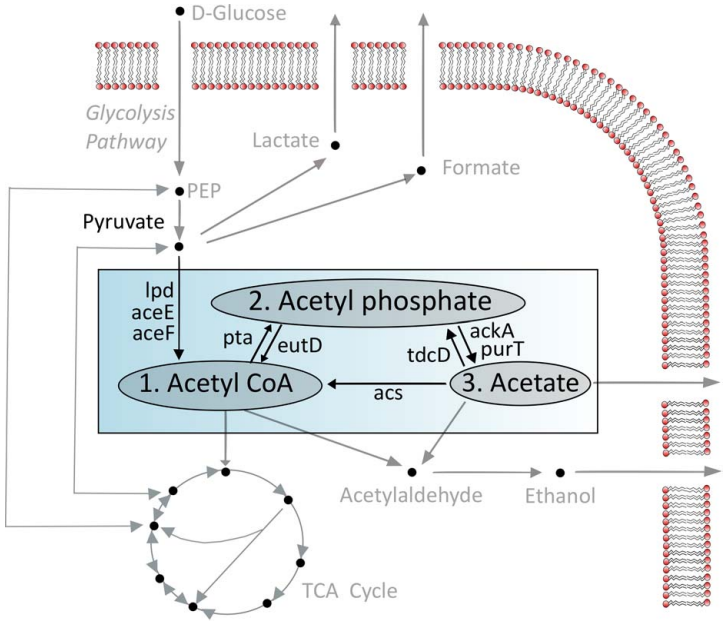


Figure 1. Simplified metabolic network for the production of acetyl-CoA based on the iJO1366 *E. coli* network [10].

2. Acetyl phosphate production from acetyl-CoA:

$$C_{23}H_{34}N_7O_{17}P_3S \ + \ HO_4P \ \rightleftharpoons \ C_2H_3O_5P \ + \ C_{21}H_{32}N_7O_{16}P_3S$$

3. Acetate production from acetyl phosphate:

$$C_2H_3O_5P \ \rightleftharpoons \ C_2H_3O_2^- \ + \ PO_3^{3-}$$

The solution of the acetate biochemical equilibrium provides new values for the acetate and CoA fluxes, which are compared to their initial estimates. If the initial and calculated values do not match, the calculated values are used for the next iteration, and this process is repeated until convergence is achieved.

Table 1 gives a complete summary of the NE iterations and solution per nmol/h of pyruvate starting from an initial estimate of acetate flux of $v_{ac} = 1$ nmol/h and stoichiometric CoA. Note that the NE iterations converge linearly and, after the first iteration, maintain a monotonically decreasing sequence of values of G/RT. In addition, the numerical results clearly reveal that the objective function, G/RT, drives the solution to one that is best for the enzymes lpd, acs, and aldB but not best for the enzyme pta. Table 2 presents a comparison of the NE predicted relative fluxes for the network and shows that the calculated results are in fair agreement with available experimental data [11].

4. Summary

A Nash Equilibrium approach to metabolic pathway analysis was proposed. The key idea behind this approach is to view enzymes as players in a multi-player game, where each player optimizes a specific objective function. A small proof-of-concept example consisting of a simplified network for the production of acetyl-CoA with four players was presented. Predicted fluxes at the NE point were found to compare favorably with existing experimental data.

Table 1. Nash Equilibrium Iterates and Solution to Acetyl-CoA Production.

Iter.	G/RT^*	v_{actp}	v_{ac}	v_{PO_3}	v_{accoa}	v_{coa}	v_{pi}	v_{co2}	v_{h2o}	*Error*
1	-12.48	0.0348	0.0844	0.0844	0.4330	1.5670	0.4330	0.5143	1.0380	0.9155
2	-12.30	0.0952	0.6787	0.1269	0.5889	1.4111	0.3303	0.8707	0.9403	0.5942
3	-14.34	0.1262	0.9593	0.1653	0.7754	1.2246	0.2608	1.1221	0.9449	0.2806
4	-15.46	0.1456	1.1275	0.1897	0.8792	1.1208	0.2171	1.2684	0.9461	0.1682
5	-16.07	0.1564	1.2223	0.2032	0.9380	1.0620	0.1927	1.3509	0.9467	0.0948
6	-16.39	0.1621	1.2752	0.2104	0.9704	1.0296	0.1799	1.3963	0.9465	0.0529
7	-16.57	0.1651	1.3045	0.2140	0.9879	1.0121	0.1732	1.4210	0.9460	0.0293
8	-16.67	0.1667	1.3206	0.2159	0.9972	1.0028	0.1698	1.4342	0.9456	0.0161
9	-16.72	0.1675	1.3294	0.2168	1.0022	0.9978	0.1680	1.4413	0.9452	0.0088
10	-16.74	0.1679	1.3341	0.2173	1.0048	0.9952	0.1671	1.4451	0.9450	0.0047
11	-16.76	0.1681	1.3367	0.2176	1.0063	0.9937	0.1666	1.4471	0.9448	0.0026
12	-16.77	0.1683	1.3381	0.2177	1.0070	0.9930	0.1664	1.4482	0.9448	0.0014
13	-16.77	0.1683	1.3388	0.2178	1.0074	0.9926	0.1662	1.4488	0.9447	0.0007

$^*G/RT = \min \sum_{j=1}^{3} G_j/RT$

Table 2. Comparison of Relative Flux Values Between Experimental Data [11] and Predictions using NE Model (Table 1).

Component	*Experimentally Measured Fluxes*	*NE Predicted Fluxes*
Acetyl-CoA	0.5443	0.4006
Acetyl phosphate	0.0000	0.0669
Acetate	0.4557	0.5324

References

[1] A. Varma and B.O. Palsson. Metabolic flux balancing: basic concepts, scientific and practical use. *Nature Biotechnology*, 12:994–998, 1994.

[2] K.J. Kauffman, P. Prakash, and J.S. Edwards. Advances in flux balance analysis. *Current Opinions in Biotechnology*, 14:491–496, 2003.

[3] H.G. Holzhutter. The principles of flux minimization and its application to estimate stationary fluxes in metabolic networks. *European Journal of Biochemistry*, 271:2905–2922, 2004.

[4] A.A. Julius, M. Imielinski, and G.J. Pappas. Metabolic networks analysis using convex optimization. In *Proceedings of the 47th IEEE Conference on Decision and Control*, page 762, 2008.

[5] K. Smallbone and E. Simeonidis. Flux balance analysis: a geometric perspective. *Journal of Theoretical Biology*, 258:311–315, 2009.

[6] E. Murabito, E. Simeonidis, K. Smallbone, and J. Swinton. Capturing the essence of a metabolic network: a flux balance analysis approach. *Journal of Theoretical Biology*, 260(3):445–452, 2009.

[7] S. Lee, C. Phalakornkule, M.M. Domach, and I.E. Grossmann. Recursive milp model for finding all the alternate optima in lp models for metabolic networks. *Computers & Chemical Engineering*, 24:711–716, 2000.

[8] C.S. Henry, L.J. Broadbelt, and V. Hatzimanikatis. Thermodynamic metabolic flux analysis. *Biophysical Journal*, 92:1792–1805, 2007.

[9] R. Mahadevan, J.S. Edwards, and F.J. Doyle. Dynamic flux balance analysis in diauxic growth in escherichia coli. *Biophysical Journal*, 83:1331–1340, 2002.

[10] J.D. Orth, T.M. Conrad, J. Na, J.A. Lerman, H. Nam, A.M. Feist, and B.O. Palsson. A comprehensive genome-scale reconstruction of escherichia coli metabolism–2011. *Molecular Systems Biology*, 11(7):535, 2011. doi: 10.1038/msb.2011.65.

[11] H. Holms. Flux analysis and control of the central metabolic pathways in escherichia coli. *FEMS Microbiology Review*, 19:85–116, 1996.

Proceedings of GOW'16, pp. 41 – 44.

On Metaheuristics for Longest Edge Bisection of the Unit Simplex to Minimise the Search Tree Size *

J.M.G. Salmerón[1], J.L. Redondo[1], E.M.T. Hendrix[2] and L.G. Casado[1]

[1]*Dept. Informatics, University of Almería (CeiA3), Spain,* {josemanuel,jlredondo,leo}@ual.es

[2]*Dept. Computer Architecture, University of Málaga, Spain,* eligius@uma.es

Abstract In several areas like Global Optimization using branch-and-bound (B&B) methods for mixture design, the unit n-simplex is refined by longest edge bisection. This process provides a binary search tree. For $n > 2$, simplices appearing during the refinement process can have more than one Longest Edge (LE). The size of the resulting binary tree depends on the specific sequence of bisected longest edges. The question is how to choose the LE to bisect. In a previous work, a set of LE indices generating the minimum number tree size was presented for $n = 3$ under the so-called m_k-valid condition by means of a deterministic method that examined the whole set of minimum Binary Trees (BTs). We are interested in find out new sets of rules for dimensions higher than 3. However, full enumeration of all possibilities is memory consuming, which makes it inappropriate for the problem at hand. In view of this situation, the use of a metaheuristic appears to be appropriate. In this work, the initial problem is reformulated as a combinatorial optimization problem, where the objective is to find the optimal subset of rules (among a finite and predefined set) to minimize the number of simplices in the BT. The problem is then solved through a stochastic algorithm based on the principles of the Tabu Search method.

Keywords: Simplex, Branch and bound, Longest edge bisection, Bisection sequence

1. Introduction

Global Optimization deals with finding the minimum or maximum value of an objective function f on a closed set with a non-empty interior. We focus here on the standard n-simplex defined in the $(n+1)$-dimensional space

$$S = \left\{ x \in \mathbb{R}^{n+1} \mid \sum_{j=1}^{n+1} x_j = 1;\ x_j \geq 0 \right\}. \tag{1}$$

We study the binary tree (BT) implicitly generated by the refinement of the n-simplex where the simplex division is defined by the Longest Edge Bisection rule (LEB) [5, 7, 9]. LEB is a popular way of iteratively refining a simplex in the context of the finite element method, since it is very simple and can easily be applied in higher dimensions [3]. It is based on splitting a simplex using the hyperplane that connects the mid point of the longest edge of a simplex with the opposite vertices, as illustrated in Figure 1.

In principle, the refinement process can continue infinitely. Nevertheless, for the problem at hand, a stopping criterion has been defined, i.e., the branching process continues until the size of the simplex is smaller than or equal to the desired accuracy ϵ.

*This work is funded by grants from the Spanish Ministry (TIN2015-66680) and Junta de Andalucía (P11-TIC7176 and P12-TIC301), in part financed by the European Regional Development Fund (ERDF). J.M.G. Salmerón is a fellow of the Spanish FPU program. Juana López Redondo is a fellow of the Spanish 'Ramón y Cajal' contract program, co-financed by the European Social Fund.

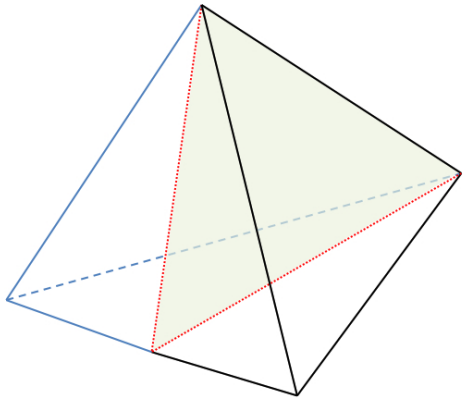


Figure 1. Example of the first Longest Edge Bisection (LEB) on a regular 3-simplex.

During the refinement algorithm, decisions on which longest edge should be bisected have to be carried out. As can be seen in Figure 1, initially, it does not matter which edge is selected, because all generated sub-simplices differ only in orientation. However, notice that after the first subdivision, the generated sub-simplices are irregular and have three (out of six) edges with the longest length.

Calculation of distances between vertices to determine the longest edges is a computationally intensive operation in the partition process. This computation can be avoided by precomputing a set of LE indices to be bisected. Each LE is stored as a pair of vertex indices for a simplex in the BT. It is desirable to find sets with an appropriate repetition of LE indices, because their computational management is more efficient than distance calculation. We look for sets of LEs with a high repetition of longest edge indices per level of any of the smallest BTs.

A set of LE indices generating the minimum number of classes of simplices [4], was presented in [2], for $n = 3$.

This set was validated in [1] as the best one under the so-called m_k-valid condition, by considering a deterministic method that explores the whole set of minimum BTs.

This exact method, aside from being very time consuming, has to be executed in architectures with high memory resources. In fact, the computer can run out of memory for larger instances.

In this work, we are interested in generating a new set of rules for dimensions higher than 3. To this aim, now we formulate the problem as a combinatorial optimization problem, where the objective is to seek the optimal subset of rules (among a finite and predefined set) that minimizes the number of simplices in the BT. To be more specific, the set of rules defining the domain of the problem is composed by all the possible ones based on m_k-valid condition for $k = 2, 4$-valid [1]. The cardinal of this set is not negligible and therefore, facing this new optimization problem with an exhaustive search cannot be feasible. To deal with this hard-to-solve optimization problem, a metaheuristic method has been designed.

2. Minimizing the tree size

To measure how good a subset R of rules is, Algorithm 1 is used. Such an algorithm returns the size of the binary tree, i.e. the number of simplices obtained when the subset R is applied from an initial simplex S_1 and a required precision ϵ is considered as a termination criteria.

Input: S: simplex, ϵ: accuracy, R: Set of rules.
Output: ns
$\Lambda := \{S_1\}$ `// Set of simplices not yet split`
$ns := 1$ `// Number of simplices`
while $\Lambda \neq \emptyset$ **do**
  Extract first simplex S_i from Λ
  if $(w(S_i) > \epsilon)$ `// Final accuracy not reached`
  then
    {j,k}:=SelectLE(S_i, R) `// Select the longest edge from` R
    $\{S_{2i}, S_{2i+1}\}$:= Bisect(S_i, j, k)
    Append simplices $2i$ and $2i+1$ in Λ.
    $ns := ns + 2$.

Algorithm 1: TreeSize(S_1, ϵ, R).

3. The optimization algorithm

The proposed metaheuristic consists of an iterative process, which guides the search toward feasible solutions by using techniques inherited from the Tabu Search (TS) [8]. The Tabu Search (TS) has traditionally been used on combinatorial optimization problems, as the one considered in this work. Many of the applications in literature concern to scheduling, telecommunications, design, production, inventory and investment, location and allocation, routing, logical and artificial intelligence, graph optimization, technology and general combinational optimization.

Similarly to TS, the proposed algorithm exploits an adaptive memory framework as a way of taking advantage of the history of the search. Information is collected through the use of a Tabu List (TL), usually identified as *short term memory* in literature. The TL keeps track of the most recently visited solutions and excludes them from the neighbourhood of the current solution. The use of a short term memory helps the algorithm to escape from local optima and avoids cycles (sequences of moves that constantly repeat themselves) [6].

A comprehensive computational study is carried out. From the results, it can be concluded that the designed heuristic provides useful and practical solutions for the problem at hand.

4. Summary

In this work, a metaheuristic optimization algorithm, based on the principles of the Tabu Search, is presented. Such a method provides a subset of rules with the aim at minimizing the number of simplices in a BT. The preliminary results are promising and place the stochastic method as a good alternative to deal with this problem.

References

[1] Salmerón JMG, Aparicio G, Casado LG, García I, Hendrix EMT and G-Tóth B. Generating a smallest binary tree by proper selection of the longest edges to bisect in a unit simplex refinement. *Journal of Combinatorial Optimization*, 1–14. 2015.

[2] Aparicio G, Casado L, G-Tóth B, Hendrix EMT, García I. On the minimim number of simplex shapes in longest edge bisection refinement of a regular n-simplex. *Informatica*, 26(1):17–32, 2015.

[3] Hannukainen A, Korotov S, Křížek M. On numerical regularity of the face-to-face longest-edge bisection algorithm for tetrahedral partitions. *Science of Computer Programming*, 90:34–41, 2014.

[4] Aparicio G, Casado LG, Hendrix EMT, García I and G-Tóth B. On computational aspects of a regular n-simplex bisection. In: *Proceedings of the Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet*

Computing , IEEE Computer Society, Compiegne, France, pp 513–518, 2013.

[5] Hendrix E, Casado LG, Amaral P. Global optimization simplex bisection revisited based on considerations by Reiner Horst. In: *Computational Science and Its Applications - ICCSA 2012*, Lecture Notes in Computer Science, vol 7335, Springer Berlin Heidelberg, pp 159–173, 2012.

[6] Alba E. Parallel Metaheuristics: A New Class of Algorithms. Wiley-Interscience. 2005.

[7] Horst R. On generalized bisection of n-simplices. *Mathematics of Computation*, 66(218):691-698, 1997.

[8] Glover F and Laguna M. Tabu search. Kluwer Academic Publisher. 1997.

[9] Adler A. On the Bisection Method for Triangles. *Mathematics of Computation*, 40(162):571–574, 1983.

Proceedings of GOW'16, pp. 45 – 48.

On Regular Simplex Division in Solving Blending Problems*

J.M.G. Salmerón[1], L.G. Casado[1], E.M.T. Hendrix[2] and J.F.R Herrera[1]

[1]*Dept. Informatics, University of Almería (CeiA3), Spain,* {josemanuel,leo,juanfrh}@ual.es

[2]*Dept. Computer Architecture, University of Málaga, Spain,* eligius@uma.es

Abstract The blending problem is studied as a problem of finding cheap robust feasible solutions on the unit simplex. Usually longest edge bisection is used in the simplex refinement process, because it is an easy division method that guarantees finding a solution. However, with this method more than one longest edge could appear in the simplex and the behaviour of the search is different depending on the selected longest edge. In this work, an alternative division method using regular simplices is studied for blending problems.

Keywords: Blending, Branch and bound, Simplex, Regular division, Longest edge bisection

1. Introduction

The mixture design (blending) problem consists in identifying mixture products (raw materials), each represented by a vector $x \in \mathbb{R}^n$, which meet certain requirements. The set of possible mixtures is mathematically defined by the unit simplex

$$S = \left\{ x \in \mathbb{R}^d \ : \sum_{j=1}^{d} x_j = 1;\ 0 \leq x_j \leq 1 \right\}, \tag{1}$$

where the variables x_j represent the fraction of the components in a product x. In mixture design problems, the objective is to minimise the cost of the product maintaining quality expectations.

In practical situations, such problems are solved on a daily base in industry where requirements are often modelled by linear inequalities [2, 3, 4]. Due to a large project on product design, a study was done on how to deal with quadratic requirements, with semi-continuity and how to generate robust products [5, 6]. The semi-continuous quadratic mixture design problem (SCQMD) is described as a problem with linear, quadratic and semi-continuity constraints. Moreover, a linear cost objective and an integer valued objective are introduced. The goal is to deal with the SCQMD problem from a branch-and-bound perspective generating robust solutions.

The semi-continuity of the variables is related to a minimum acceptable dose md that the practical problems reveal, i.e. either $x_j = 0$ or $x_j \geq md$. Figure 1 shows a graphical example of the search space in 2D (left hand side) and 3D (right hand side) consisting of unit simplices removing the space where the minimum dose constraint is not satisfied.

In many Branch and Bound (BnB) algorithms, the division of the simplex is usually performed by Longest Edge Bisection (LEB). The consequence of using bisection, are i) there exist evaluated vertices of subsimplices which do not belong to a mesh-grid, ii) the size of

*This work is funded by grants from the Spanish Ministry (TIN2015-66680) and Junta de Andalucía (P11-TIC7176 and P12-TIC301), in part financed by the European Regional Development Fund (ERDF). J.M.G. Salmerón is a fellow of the Spanish FPU program.

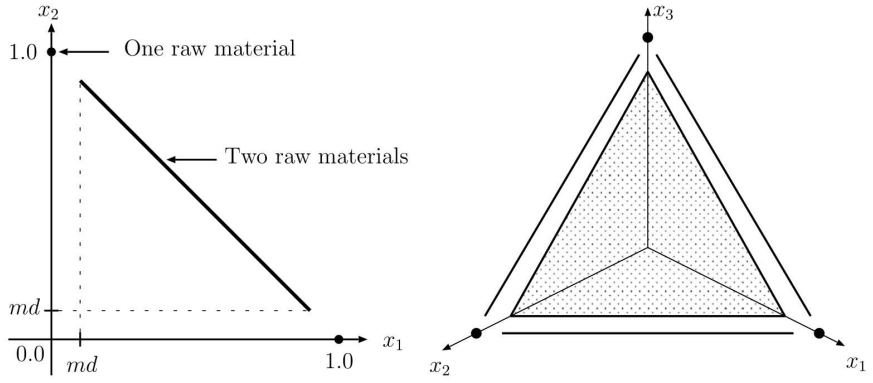


Figure 1. 2D and 3D simplices removing the minimum dose region.

the complete tree is larger than using regular simplices division, and iii) subsimplices storage requirements are larger because irregular simplices appear when bisection is used and all vertices are required to be stored (for regular simplicial division, to store the centre and radius is enough). This paper proposes to apply a division method that splits the initial unit simplex in regular subsimplices to evaluate the improvements of the BnB algorithm in terms of evaluated simplices, vertices and memory requirements.

The basic idea in BnB methods consists of a recursive decomposition of the original problem into smaller disjoint subproblems until the solution is found. However, for our case of study with Regular by Regular (RbR) simplex division, overlapping between subsimplices occurs to completely cover the divided simplex by subsimplices. To avoid re-evaluation of overlapped region, a subsimplex is not generated when it is overlapped by other that need to be divided [1].

2. Longest edge bisection

Reference [6] shows some results of blending problems using LEB as division method in the BnB algorithm. Table 1 shows an example for a hard-to-solve problem, showing the number of simplices and vertices evaluated when LEB is used. We are looking for an alternative simplex division method in order to reduce the evaluations, memory requirements and execution time.

The meaning of table notation is:

- *Problem*: Problem name.
- *NSimplex*: Number of evaluated simplices.
- *NVertex*: Number of evaluated vertices.
- *End NSimplex*: Number of simplices in the solution list.
- *End NVertex*: Number of vertices associated to simplices in the solution list.
- *N.Sol.*: Binary vector which shows if for a given number of raw materials the algorithm found a solution.

We focus on different test problems to benchmark the algorithm. The first one (*RumCoke*) has two linear constraints and two quadratic constraints. The second one (*Case 2*) was taken from an industrial example having five quadratic constraints. Both are three dimensional problems. Additionally, the algorithm was tested with the two seven dimensional problems (*UniSpec1* and *UniSpec5b*) provided by Unilever Research based on similar quadratic functions, but with different requirements. *UniSpec1* has one linear constraint and five quadratic constraints and *UniSpec5b* has four quadratic constraints. Complete specifications about the problems can be found in [6].

Table 1. Numerical results using LEB

Problem	RumCoke	Case2	UniSpec1	UniSpec5b
NSimplex	881	561	72,805	144,929,859
NVertex	340	231	31,841	32,162,536
End NSimplex	201	105	3,104	27,959,718
End NVertex	127	72	953	1,205,184
Time	00:00:00	00:00:00	00:00:00	00:21:03
Memory	4.6 KB	2.9 KB	1.2 MB	1.5GB
N.Sol.	0,0,0	0,0,1	0,1,1,0,0,0,0	0,0,1,1,0,0,0

As Table 1 shows, the number of evaluations is huge for the *UniSpec5b* problem, as well as the memory requirements and execution time. So, the aim of this study is to evaluate RbR division methods in order to know if the performance of the algorithm improves.

3. Regular by regular division

Simplex refinement by regular simplex division was presented in [1]. Now we are interested in applying this new simplex refinement method to a blending problem, comparing the RbR with LEB results [7].

Based on the considerations in [8], a refinement of the unit simplex should use hyperplanes parallel to the initial facets and generate subproblems having edges of size multiple of $\alpha = \frac{1}{G}$, where $G = \lceil \frac{1}{\epsilon} \rceil$ is the number of grid segments per axis.

Regular simplices facilitate the effectiveness of rejection tests based on covering by infeasible spheres because all vertices are at the same distance.

These refinement methods are introduced next and they are described in depth in [9].

3.1 2USC

The two-uniform-simplex-cover refines a regular simplex in d overlapping regular subsimplices with two subsimplices per edge (see Figure 2b). A reduction factor $\beta \geq \frac{d-1}{d}$ guarantees the covering of the refined simplex.

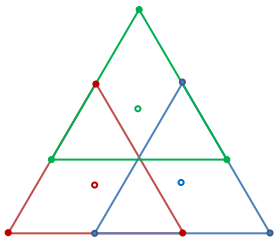


Figure 2a. 2USC in $\mathbb{R}^3$. Dots represent the vertices and the centre of a subsimplex.

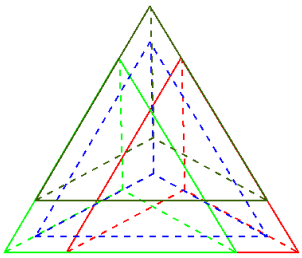


Figure 2b. 2USC in $\mathbb{R}^4$.

3.2 2MUSC

We now introduce 2-mixed-uniform-simplex-cover refinement (2MUSC) which combine 2USC and 2∇USC in the refinement of the unit simplex. The 2∇-uniform-simplex-cover (2∇USC) introduced in [1] refines a regular simplex in $d+1$ regular subsimplices. In contrast to 2USC, d of them have the same orientation using a reduction factor $\frac{d-2}{d-1} \leq \rho < \frac{d-1}{d}$ and one non-overlapped inverted subsimplex has a reduction factor $\varrho = d(1-\rho) - 1$. Figure 3 shows an example for $d = 4$ where $\rho = 2/3$ and $\varrho = 1/3$.

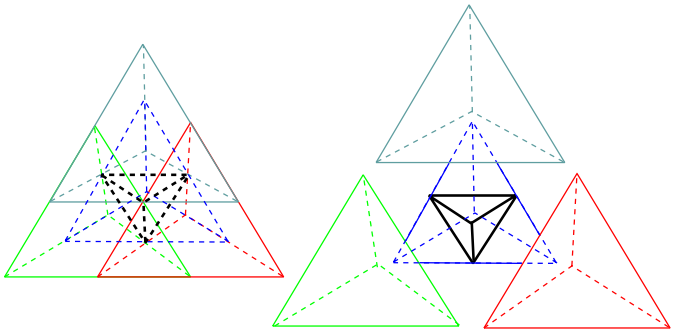


Figure 3. 2∇USC for $d = 4$.

In 2∇USC, the subsimplices overlap less than in 2USC, but the subsimplices are not equally oriented and sized. 2∇USC requires a delicate choice of the reduction factor ρ. In order to have a value for ρ being a multiple of α, ρ may be greater of equal to $(d-1)/d$. In that case, the reduction factor of the inverted simplex $\varrho \leq 0$. This means that the inverted subsimplex does not exist and 2∇USC reduces to 2USC. Consequently, the overlap of the subsimplices is larger. Depending on the current simplex, either a reduction factor ρ or β is applied to divide the simplex using one of both methods presented before.

Results of LEB, 2USC, and 2MUSC for the problems presented in Table 1 will be shown in the talk.

4. Summary

Regular by regular division of the unit simplex seems to be appealing, because only the grid points are evaluated, the number of subsimplices in the complete tree are smaller than using LEB, and memory requirements are also smaller for that case. Previous arguments are valid for the complete tree, and the question is if they are still valid for a BnB algorithm with pruning. Numerical comparison of different division rules will be shown at the talk.

References

[1] G.-Tóth, B., Hendrix, E.M.T., Casado, L.G., García, I. On refinement of the unit simplex using regular simplices. *Journal of Global Optimization*, 64(2):305–323, 2016.

[2] Ashayeri J., van Eijs A.G.M., Nederstigt P. Blending modelling in a process manufacturing: A case study. *European Journal of Operational Research*, 72(3):460–468, 1994.

[3] Bertrand J.W.M., Rutten W.G.M.M. Evaluation of three production planning procedures for the use of recipe flexibility. *European Journal of Operational Research*, 115(1):179–194, 1999.

[4] Williams H.P. Model building in mathematical programming. Wiley, Chichester. 1993.

[5] Casado L.G., Hendrix E.M.T., García I. Infeasibility spheres for finding robust solutions of blending problems with quadratic constraints. *Journal of Global Optimization* 39(4):577–593, 2007.

[6] Hendrix E.M.T., Casado L.G., García I. The semi-continuous quadratic mixture design problem: Description and branch-and-bound approach. *European Journal of Operational Research* 191(3):803–815, 2008.

[7] Herrera J.F.R., Casado L.G., Hendrix E.M.T., García I. Pareto optimality and robustness in bi-blending problems. *TOP*, 22(1):254–273, 2012.

[8] Casado L.G., Hendrix E.M.T., Salmerón J.M.G., G.-Tóth, B., García, I. On grid aware refinement in branch and bound. *Submitted to Journal of Global Optimization*, 2016.

[9] Casado L.G., Salmerón J.M.G., G.-Tóth B., Hendrix E.M.T. and García I. On regular refinement of unit simplex by just visiting grid points. In: *Proceedings of GOW'16: XIII Global Optimization Workshop*, Braga, Portugal, 2016.

Proceedings of GOW'16, pp. 49 – 52.

Strengthening Convex Relaxations of Mixed Integer Non Linear Programming Problems with Separable Non Convexities*

Claudia D'Ambrosio[1], Antonio Frangioni[2] and Claudio Gentile[3]

[1]*LIX UMR 7161, École Polytechnique, Palaiseau, France,* dambrosio@lix.polytechnique.fr

[2]*DI, Università di Pisa, Pisa, Italy,* frangio@di.unipi.it

[3]*IASI, Consiglio Nazionale delle Ricerche, Rome, Italy,* gentile@iasi.cnr.it

Abstract In this work we focus on methods for solving mixed integer non linear programming problems with separable non convexities. In particular, we propose a strengthening of a convex mixed integer non linear programming relaxation based on perspective reformulations. The relaxation is a subproblem of an iterative global optimization algorithm and it is solved at each iteration. Computational results confirm that the perspective reformulation outperforms the standard solution approaches.

Keywords: Global optimization algorithm, Separable functions, Perspective reformulation

1. Introduction and motivation

We consider the following Mixed Integer Non Linear Programming (MINLP) problem:

$$\begin{aligned} \min & \sum_{j \in N} C_j x_j & \\ & f_i(x) + \sum_{k \in H_i} g_{ik}(x_k) \leq 0 & \forall i \in M \\ & L_j \leq x_j \leq U_j & \forall j \in N \\ & x_j \text{ integer} & \forall j \in I \end{aligned}$$

where:

- $f_i : \mathbb{R}^n \to \mathbb{R}$ are convex functions $\forall i \in M$,
- $g_{ik} : \mathbb{R} \to \mathbb{R}$ are non convex univariate function $\forall i \in M, \forall k \in H_i$,
- $H_i \subseteq N \ \ \forall i \in M$, and
- $I \subseteq N$.

For $j \in H_i$ we assume that the bounds are finite $\forall i \in M$. Note also that, if H_i is empty, the corresponding i-th constraint defines a convex set. The problem, that we will call P in the following, is an MINLP problem with separable non convexities represented by terms $\sum_{k \in H_i} g_{ik}(x_k)\ \forall i \in M$ with H_i non-empty.

In D'Ambrosio et al. [2, 3], the authors proposed a global optimization algorithm called Sequential Convex MINLP (`SC-MINLP`) based on the definition of a convex MINLP problem

*The first author acknowledges the partial financial support provided by MINO Initial Training Network (ITN) under the Marie Curie 7th European Framework Programme and the partial financial support under grant ANR 12-JS02-009-01 "ATOMIC".

that provides a lower bound to P and a non convex NLP problem that provides an upper bound to P. Both problems were solved through solvers used as "black-box".

Here we focus on the lower bounding problem and propose alternative solution methods.

2. Convex MINLP relaxation

For simplicity, let us consider a term of the form $g(x_k) := g_{ik}(x_k)$ such that $g : \mathbb{R} \to \mathbb{R}$ is a univariate non convex function of x_k , for some k ($k \in N$).

As the term $g(x_k)$ is univariate, it is possible to automatically detect the concavity/convexity intervals and to compute the following parameters and sets:

- $[P_{p-1}, P_p] :=$ the p-th subinterval of the domain of g ($p \in \{1 \dots \bar{p}\}$);
- $\check{H} :=$ the set of indices of subintervals on which g is convex;
- $\hat{H} :=$ the set of indices of subintervals on which g is concave.

The main idea is to find a relaxation of P that keeps the convex parts of $g(x_k)$ while replaces the concave intervals with a piece-wise linear relaxation, see Figure 1.

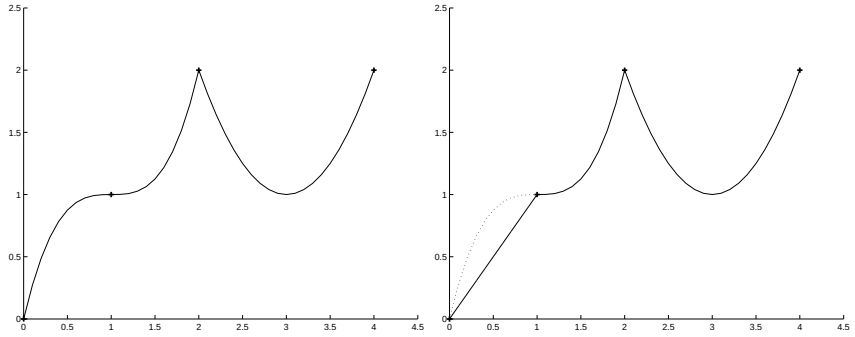


Figure 1. Univariate function $g(x_k)$ on the left and its convex MINLP relaxation on the right

To do this, we replace the term $g(x_k)$ with:

$$\sum_{p\in\check{H}} g(P_{p-1} + \delta_p) + \sum_{p\in\hat{H}} (\alpha_p g(P_p) + (1 - \alpha_p) g(P_{p-1})) - \sum_{p=1}^{\bar{p}-1} g(P_p) \ ,$$

and we include the following set of new variables

- δ_p: a continuous variable assuming a positive value iff $x_k \geq P_{p-1}$ ($p \in \{1, \dots, \bar{p}\}$);
- $z_p \in \{0, 1\}$: binary variable taking value 1 iff $x_k \geq P_p$ ($p \in \{1, \dots, \bar{p} - 1\}$);
- $\alpha_p \in [0, 1]$: the weight of the breakpoints of subinterval p ($p \in \hat{H}$);

and new constraints:

$$\begin{aligned}
x_k &= P_0 + \sum_{p=1}^{\bar{p}} \delta_p \\
\delta_p &\geq (P_p - P_{p-1}) z_p && \forall p \in \check{H} \cup \hat{H} \\
\delta_p &\leq (P_p - P_{p-1}) z_{p-1} && \forall p \in \check{H} \cup \hat{H} \\
\delta_p &= (P_p - P_{p-1}) \alpha_p && \forall p \in \hat{H}
\end{aligned}$$

with two dummy variables $z_0 := 1$, $z_{\bar{p}} := 0$. Note that in [2, 3] the piece-wise linear part of the relaxation is improved iteration by iteration by adding other breakpoints. However, we do not discuss this aspect as it is not the focus of this work.

2.1 Convex MINLP relaxation strengthening

We now strengthen the convex MINLP relaxation thanks to Perspective Reformulation (PR), see [4], that can be applied as the lower bounding problem is a piece-wise convex problem. In particular, we use a PR of each convex piece.

Let us consider a convex piece $p \in \breve{H}$ and let us define $\Delta_p = P_p - P_{p-1}$. The relevant parts of the convex MINLP relaxation are $g(P_{p-1} + \delta_p) - g(P_{p-1})$ and $\Delta_p z_p \leq \delta_p \leq \Delta_p z_{p-1}$ with $z_{p-1} \in \{0, 1\}$ and $z_p \in \{0, 1\}$ that define the non convex function

$$g(\delta_p, z_{p-1}) = \begin{cases} 0 & \text{if } z_{p-1} = 0 \text{ and } \delta_p = 0 \\ g(P_{p-1} + \delta_p) - g(P_{p-1}) & \text{if } z_{p-1} = 1 \text{ and } \delta_p \leq \Delta_p \\ +\infty & \text{otherwise} \end{cases}$$

and its convex envelope

$$\tilde{g}(\delta_p, z_{p-1}) = \begin{cases} 0 & \text{if } z_{p-1} = 0 \text{ and } \delta_p = 0 \\ z_{p-1} g(P_{p-1} + \frac{\delta_p}{z_{p-1}}) - g(P_{p-1}) & \text{if } z_{p-1} \in (0, 1] \text{ and } \delta_p \leq \Delta_p \\ +\infty & \text{otherwise.} \end{cases}$$

We use the function $\tilde{g}(\delta_p, z_{p-1})$ to define the PR of the lower bounding problem that is stronger than the one presented in the previous section.

3. Preliminary computational results

In this section we present preliminary computational results that were performed on instances of the non linear knapsack problem, see [2, 3] for details on this problem and related instances.

In Tables 1 and 2 we show the results on 10 instances with 200 and 10 with 500 objects, respectively. We compare:

- `Bonmin`: the open source solver [1];
- `lin-PR`: the linearization of perspective reformulation of the lower bounding problem, solved via `IBM CPLEX` solver [5];
- `lin-MINLP`: the linearization of original LB problem, solved via `IBM CPLEX`.

The first column represents the instance number. For each method we report the objective function value of the best solution found and the CPU time needed (or the percentage gap if the time limit of 1 hour was reached).

In Table 1 the 3 methods find the same optimal value. However, the CPU time needed by `Bonmin` is of at least 3 orders of magnitude larger than the other two methods. In particular, `Bonmin` takes $1,554.36$ seconds to solve the 10 instances while `lin-PR` and `lin-MINLP` take 1.37 and 6.51 seconds, respectively. On the contrary, instances with 500 objects cannot be solved by `Bonmin` within the time limit (Table 2). Method `lin-PR` outperforms `lin-MINLP` as the same optimal value is found but the CPU time needed is of 1 order of magnitude smaller. In particular, the 10 instances can be solved in 4.65 vs 41.12 seconds.

4. Conclusions and perspectives

We presented an iterative algorithm aimed at solving MINLPs where the non convexity in the objective and constraint functions is manifested as the sum of non convex univariate functions. In particular, we focus on alternative ways to provide a solution of the lower bounding problem that is solved at each iteration, i.e., using perspective reformulation of the lower

	Bonmin		lin-PR		lin-MINLP	
instance n.	f^*	CPU time	f^*	CPU time	f^*	CPU time
0	-6,118.10	32.40	-6,117.78	0.07	-6,118.10	0.63
1	-7,420.28	32.99	-7,420.28	0.23	-7,420.02	0.56
2	-6,514.60	109.47	-6,514.60	0.08	-6,514.60	0.76
3	-6,668.10	55.28	-6,668.10	0.20	-6,668.10	1.22
4	-7,362.95	32.84	-7,362.95	0.09	-7,362.95	0.16
5	-6,387.83	284.17	-6,387.71	0.16	-6,387.83	0.66
6	-7,449.27	837.77	-7,449.12	0.17	-7,449.27	1.20
7	-7,172.44	27.89	-7,172.44	0.13	-7,172.44	0.31
8	-6,513.91	33.18	-6,513.91	0.11	-6,513.91	0.20
9	-6,373.31	108.37	-6,373.31	0.13	-6,373.31	0.81

Table 1. Tests on the non linear knapsack problem instances with 200

	Bonmin		lin-PR		lin-MINLP	
instance n.	f^*	% gap	f^*	CPU time	f^*	CPU time
0	-16,728.73	0.07705940	-16,728.73	0.55	-16,728.73	3.14
1	-17,356.28	0.02726800	-17,356.27	0.20	-17,356.28	1.03
2	-17,609.40	0.14382956	-17,609.40	0.44	-17,608.85	5.02
3	-16,302.00	0.04943574	-16,302.03	0.29	-16,302.00	0.73
4	-17,709.62	0.09751765	-17,692.19	0.55	-17,693.81	4.36
5	-18,785.43	0.16382116	-18,785.72	0.66	-18,785.44	4.38
6	-17,348.88	0.11963388	-17,348.88	0.83	-17,348.88	5.50
7	-16,884.16	0.05327008	-16,884.50	0.20	-16,884.06	5.39
8	-16,796.44	0.16412216	-16,796.44	0.54	-16,796.44	9.20
9	-16,605.13	0.10884773	-16,605.13	0.39	-16,604.69	2.37

Table 2. Tests on the non linear knapsack problem instances with 500 objects (1h time limit)

bounding problem in order to generate cuts. Preliminary computational results show that this approach outperform the standard algorithm implemented in `Bonmin` and the standard linearization of convex MINLPs solved via `Cplex`.

In the future, we plan to take advantage of the fact that we can reuse the PR cuts to solve the lower bounding problem at each iteration. We think also that the relaxation could be strengthened even further by considering the convex hull of a larger set as the structure of the lower bounding problem is well defined. Moreover, we plan to explore how to improve also the concave part of the relaxation by adding breakpoints on the fly.

References

[1] Bonmin. `projects.coin-or.org/Bonmin`, v. trunk.

[2] Claudia D'Ambrosio, Jon Lee, and Andreas Wächter. A global-optimization algorithm for mixed-integer nonlinear programs having separable non-convexity. In *Algorithms—ESA 2009*, volume 5757 of *LNCS*, pages 107–118. Springer, Berlin, 2009.

[3] Claudia D'Ambrosio, Jon Lee, and Andreas Wächter. An algorithmic framework for minlp with separable non-convexity. In Jon Lee and Sven Leyffer, editors, *Mixed Integer Nonlinear Programming*, volume 154 of *The IMA Volumes in Mathematics and its Applications*, pages 315–347. Springer New York, 2012.

[4] Antonio Frangioni and Claudio Gentile. Perspective cuts for a class of convex 0-1 mixed integer programs. *Mathematical Programming*, 106(2):225–236, 2006.

[5] IBM. *ILOG CPLEX 12.2 User's Manual*. IBM, 2010.

Proceedings of GOW'16, pp. 53 – 56.

Statistical Models for Global Optimization: How to Chose an Appropriate One ?*

Antanas Žilinskas[1] and Gražina Gimbutienė[2]

[1]*Institute of Mathematics and Informatics, Vilnius University, 4 Akademijos str., Vilnius, Lithuania,*
antanas.zilinskas@mii.vu.lt

[2]*Institute of Mathematics and Informatics, Vilnius University, 4 Akademijos str., Vilnius, Lithuania,*
grazina.gimbutiene@mii.vu.lt

Abstract The considered algorithms are based on several statistical models and two main optimality criteria: maximum one-step average improvement, and maximum improvement probability. The performance of algorithms of both types is investigated experimentally applying them for the sample functions of the true statistical model and for the sample functions generated according to a different model.

Keywords: Global optimization, Statistical models, Kriging, Rational decision theory

1. Introduction

Global optimization algorithms based on statistical models of objective functions are aimed at the so called class of expensive black box objective functions. Since the information on the properties of objective functions are scarce, and the computation of an objective function value is expensive (i.e. requires much computing time and possibly other resources), a natural idea, for the theoretical substantiation of global optimization algorithms, is to refer to the theory of rational decision under uncertainty. Because of the space limitation, we refer here only to [1] for the review of origins of this approach in global optimization, and to [2] for the discussion on the recent results. In the talk, more detailed review will be presented showing that the problem of selection of an appropriate statistical model is far from a reasonable solution. In this abstract we discuss the model selection limiting ourselves with the case of one-dimensional optimization but in the talk a discussion will be extended to the general case.

The problem $\min_{x\in\mathbf{A}} f(x)$ is considered where $\mathbf{A}$ is a closed interval, and $f(x)$ is a continuous function. An optimization algorithm for planning a current iteration uses as an input $f(x)$ values computed at previous iterations. To substantiate the development of an algorithm, a statistical model $\xi(x)$, $x \in \mathbf{A}$ is used, i.e. the considered objective function is interpreted as a sample function of $\xi(x)$.

2. Selection of a statistical model

The models of functions under uncertainty considered in probability theory are stochastic functions, in particular, one-dimensional case, stochastic processes. In the majority of publications, statistical models were selected from the stochastic processes well researched by the probability theoreticians. Specifically, the selected models were Gaussian stochastic processes, such as Wiener process, stationary Ornstein Uhlenbeck process, and in few cases a

*This work was supported by the Research Council of Lithuania under Grant No. MIP-051/2014.

process with smooth sample functions. However, the available information about an aimed problem and about important for global optimization properties of sample functions of available stochastic functions is not sufficient for a satisfactory substantiation of the selection of an appropriate statistical model. For a heuristic satisfaction of a selection, the visual analysis of graphics of sample functions of the considered stochastic processes can be reasonable; for some examples we refer to Figure 1. Let a type of a stochastic process be selected. Next, the

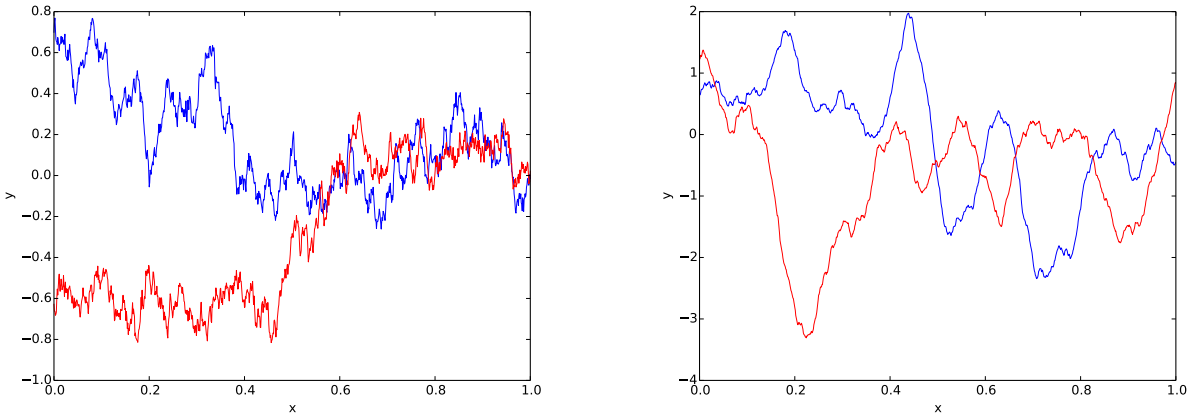


Figure 1. Examples of sample functions of the Wiener process with $\sigma = 1$ (on the left), and examples of sample functions of the Gaussian stationary random process with zero mean and covariance function $\rho(\tau) = \exp(-(\tau/c)^{1.8})$, $c = 0.07$, (on the right).

parameters should be either somehow defined or estimated using statistical techniques. This problem also is not trivial.

3. The considered algorithms

To define the considered algorithms we need the following notation. At the $k+1$ step of search, the objective function values y_j are computed at the points x_j, $j = 1, \dots, k$, and the current minimum computed value is denoted $y_{o,k}$.

The P-algorithm computes the next objective function value at the point

$$x_{k+1} = \arg\max_{x \in \mathbf{A}} \mathbb{P}(\xi(x) \leqq y_{o,k} - \varepsilon_k | x_j, y_j, j = 1, \dots, k), \tag{1}$$

meaning that it is aimed at maximal probability of the improvement of the current estimate of global minimum; here $\mathbb{P}(\cdot|\cdot)$ denotes the conditional probability, $y_{o,k} = min_{1 \leq i \leq k} y_i$, $\varepsilon_k > 0$. For the Gaussian stochastic function, the algorithm (1) is defined by a simpler formula

$$x_{k+1} = \arg\max_{x \in \mathbf{A}} \frac{y_{o,k} - \varepsilon_k - m(x|x_j, y_j, j = 1, \dots, k)}{\sigma(x|x_j, y_j, j = 1, \dots, k)}, \tag{2}$$

where $m(x|x_j, y_j, j = 1, \dots, k)$ and $\sigma(x|x_j, y_j, j = 1, \dots, k)$ denote the conditional mean and conditional standard deviation of the stochastic function at the point x. By the definition of the sequence ε_k, the strategy of search can be regulated, e.g. search is more global with larger ε_k. The formula (2) corresponds to the probability maximization not only for the Gaussian model but also for a large class of probability distributions.

The one-step Bayesian algorithm computes the next objective function where the expected improvement is maximum

$$x_{k+1} = \arg\max_{x \in \mathbf{A}} \Delta Y_{k+1}(x),$$
$$\Delta Y_{k+1}(x) = \mathbb{E}(\max\{y_{o,k} - \xi(x),\ 0\} | x_j, y_j, j = 1, \dots, k), \tag{3}$$

where $\Delta Y_{k+1}(x)$ means expected improvement at $k+1$ step in case of computing function value at the point x, $\mathbb{E}(\cdot|\cdot)$ denotes the conditional expectation. This algorithm in later papers was called also 'kriging' and 'EGO' (efficient global optimization). The reasons for renaming remained without an explanation. We note only, that the term 'kriging' originally was used to call the prediction method by the name of its author D.G.Krige; to our best knowledge Krige has not considered statistical models based global optimization methods. On the other hand, the substantiation for adding the pretentious attribute 'efficient' was needed, at least the theoretical analysis of conditions of the real efficiency and of the limitations of the algorithm; unfortunately such results were not presented by the inventors of the name EGO.

In the implementations of the one-step Bayesian algorithm with a Gaussian stochastic functions for a statistical model, the computations normally are performed according to the following formula

$$x_{k+1} = \arg\max_{x\in\mathbf{A}} \left(v_k(x)\Phi(v_k(x)) + \frac{1}{\sqrt{2\pi}} \exp(-\frac{v_k^2(x)}{2} \right), \tag{4}$$

$$v_k(x) = \frac{y_{o,k} - m(x|x_j, y_j, j=1,\dots,k)}{\sigma(x|x_j, y_j, j=1,\dots,k)},\ \Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} \exp\{-t^2/2\}dt,$$

where the notation $m(x|x_j, y_j, j=1,\dots,k)$ and $\sigma^2(x|x_j, y_j, j=1,\dots,k)$ has the same meaning as in (2). It was observed the rather frequent selection by this algorithm of a point for current computing an objective function value in a vicinity of points of best previous computations. This is related to the very small values of summands in (4) which correspond to the tails of Gaussian density and cumulative distribution functions. In case of a non-Markovian model, such a selection at early stage of search causes the instability of the whole process because the subsequently used covariance matrices become ill-conditioned.

Various modifications of (2), usually with some numerical examples, can be found in numerous publications with 'EGO' and 'kriging' in the titles and lists of key words.

4. Experiments in choosing an appropriate statistical model

For the testing of a global optimization algorithm normally is used a set of known test functions. However, by such a methodology the concrete software implementation is tested. It hardly can explain how much the performance depends on the particular theoretical assumptions, and how much on the implementation. For example, the failure of an algorithm based on a statistical model in the solution of a concrete problem can be caused by the inadequate statistical model, and can be caused by setting inappropriate parameters of the model. On the other hand, possibly, the data collected during the optimization do not significantly disagree with the model, and the failure is caused by numerical problems, e.g. errors caused by inversion of ill conditioned matrix. Therefore we have performed the investigation of performance of algorithms (2) and (4) using as objective functions the sample functions of stochastic processes which were used also as the models for the construction of algorithms.

The detailed description and motivation of the experimentation plan will be commented in the talk. Here we present particular results to illustrate the proposed methodology. Three statistical models were used in the experimentation: Wiener process with the variance parameter equal to 1, and two stationary processes with zero mean, variance equal to 1, and correlation functions $\rho(\tau) = \exp(-\tau/0.06)$ and $\rho(\tau) = \exp(-(\tau/0.07)^2)$ correspondingly; in the tables below they are referred as 'Wien','Exp', and 'Gauss'. In total, 1000 sample paths were generated for each of the considered stochastic functions. Each sample path was represented by its values at the points of the interval $[0,\ 1]$, $T = \{t_j = j/N\}$, $j = 0,\dots,N$, $N = 1000$.

An algorithm was run not only with objective functions which were the sample functions of the stochastic process used as the model for the construction of the algorithm but also with the sample functions of other stochastic functions. In the latter case, to run an optimization algo-

rithm, that assumes a certain statistical model of the objective function ("assumed model") on a sample path of extraneous stochastic function ("actual model"), we estimated the parameters of the assumed models for each generated sample path by the maximum likelihood method. For the estimation the uniformly-spaced function values at $K = \{k_j = j/M\}$, $j = 0, \ldots, M$, $M = 10$, were used; further, by the optimization algorithm these values were not used. Let us note, that the algorithm (2) based on the Wiener process model is fully data driven; since the unique parameter of the model can be eliminated in (2) there is no need to estimate that parameter.

We assumed the global minimum found if for a budget of $n = 35$ function evaluations some measurement location $x_i, i \in \{1, \ldots n\}$, was generated that satisfied the condition:

$$|x_i - x^*| \leq 0.002.$$

In the tables below the numbers (out of 1000) of runs are given where global minimum was not found with the prescribed accuracy.

Table 1. Results of the P-algorithm.

Alg\ Func	Wien	Exp	Gauss
Wien	5.6	26	25.8
Exp	13.5	22.4	18.6
Gauss	56.5	67.2	9.1

Table 2. Results of the one-step Bayesian algorithm.

Alg\ Func	Wien	Exp	Gauss
Wien	6.9	28.4	24.3
Exp	8.6	27.9	33.8
Gauss.	55.4	68.2	6.4

The results presented in the tables clearly show that the algorithms based on the statistical smooth function model perform more poorly in minimization of stiffly oscillating functions than vice versa.

In the talk will be presented detailed results of the experiments with variety of statistical models and corresponding to the results conclusions on selection of statistical models. For example, the results of experiments show that the algorithms based on the statistical models, the sample functions of which are more oscillating (larger number of sharp local minima), distribute points for computing objective function values more globally than those based on smooth function models.

5. Summary

An experimentation methodology is proposed aimed at the substantiation of selection of an appropriate statistical model for the construction of a global optimization algorithm.

References

[1] Törn, Aimo and Antanas Žilinskas. *Global Optimization*, Springer, 1988.

[2] Žilinskas, Antanas and Anatoly Zhigljavsky. Stochastic Global Optimization: a Review on the Occasion of 25 Years of Informatica, *Informatica*, vol.27, 2016, in print.

Proceedings of GOW'16, pp. 57 – 60.

A Branch-and-Bound Algorithm for Bi-Objective Problems

Julia Niebling[1] and Gabriele Eichfelder[1]

[1]*Technische Universität Ilmenau, Institute for Mathematics, Weimarer Straße 25, 98693 Ilmenau, Germany,* {julia.niebling, gabriele.eichfelder}@tu-ilmenau.de

Abstract An improved discarding test for a branch-and-bound algorithm for box-constrained bi-objective optimization problems is presented. The aim of the algorithm is to compute a covering of all global optimal solutions. We introduce the algorithm which uses selection, discarding and termination tests. The discarding tests are the most important aspect, because they examine in different ways whether a box can contain optimal solutions. For this, we are using the αBB-method from global scalar optimization and present and discuss an improved test compared to those from the literature.

Keywords: Multi-objective optimization, Branch-and-bound algorithm, αBB-method

1. Introduction

Multi-objective optimization problems arise for instance in engineering or economics. In such problems various objective functions have to be minimized simultaneously. In general there is no point which minimizes all objective functions at the same time. In this article we are dealing with box-constrained bi-objective optimization problems. The problem is formulated as follows:

$$\min_{x \in X_0} f(x) = (f_1(x), f_2(x))^T \tag{P}$$

where $f_i \colon \mathbb{R}^n \to \mathbb{R},\ i = 1, 2$, are twice continuously differentiable functions. The set $X_0 \subseteq \mathbb{R}^n$ is called *box*, that is $X_0 = \{x \in \mathbb{R}^n \mid \underline{x} \leq x \leq \overline{x}\}$ with two points $\underline{x}, \overline{x} \in \mathbb{R}^n$. Thereby we write $x \leq y$ if $x_i \leq y_i$ for all $i = 1, \ldots, n$. A common optimality concept in bi-objective optimization is the following:

Definition 1. *A point $x^* \in X_0$ is said to be **efficient** of (P) if there does not exist another $x \in X_0$ such that $f_j(x) \leq f_j(x^*)$ for all $j = 1, 2$ and $f_k(x) < f_k(x^*)$ for at least one $k \in \{1, 2\}$. The set of all efficient points is called **efficient set**. We say x^1 **dominates** x^2 if $x^1, x^2 \in X_0$, $f_j(x^1) \leq f_j(x^2)$ for all $j = 1, 2$ and $f_k(x^1) < f_k(x^2)$ for at least one $k \in \{1, 2\}$.*

We can define similar terms in the image space.

Definition 2. *Let $x^* \in X_0$. A point $y^* = f(x^*)$ is said to be **nondominated** if x^* is efficient. The set of all nondominated points is called **nondominated set**. We say y^1 **dominates** y^2 if $y^1, y^2 \in \mathbb{R}^2$, $y^1_j \leq y^2_j$ for all $j = 1, 2$ and $y^1_k < y^2_k$ for at least one $k \in \{1, 2\}$.*

Based on the algorithm presented in [3] we determine a covering of the efficient or nondominated set. The procedure is a branch-and-bound algorithm which uses bounding criteria based on interval analysis. The most important step of the algorithm is the discarding test. For our procedure three tests are applied. The first one uses the αBB-method introduced in [6] for determining convex underestimators of the considered functions. These can be used to determine lower bounds of the objective functions similar to those gained directly by interval arithmetic in [3]. These lower bounds can be used for a discarding test. We improve this test by using the basic idea of Benson's outer approximation algorithm for multi-objective convex

optimization problems, see [1], [2] and [5]. Furthermore we also use the monotonicity test presented in [3]. In section 2 the main algorithm and the new discarding test are introduced. Some numerical results are shown and discussed in section 3. We will end with an outlook for future research.

2. The algorithm

For our procedure we also use interval analysis, which is a well-known tool in global optimization, see for instance [7]. We denote the set of all n-dimensional real boxes with $\mathbb{R}^n$ and the width of a box $X = [\underline{x}, \overline{x}] \in \mathbb{R}^n$ with $\omega(X)$, i.e. $\omega(X) = \|\overline{x} - \underline{x}\|$. The midpoint $\text{mid}(X)$ of X is calculated by $\text{mid}(X) := \frac{\overline{x}+\underline{x}}{2}$. The natural interval extension of a function $f\colon \mathbb{R}^n \to \mathbb{R}^2$ is denoted by $F\colon \mathbb{R}^n \to \mathbb{R}^2$.
The principal steps of the algorithm are the following:

main algorithm {
  $\mathcal{L}_\mathcal{W} := \{X_0\}, \mathcal{L}_\mathcal{S} := \{\}, \mathcal{L}_{\mathcal{PNS}} = \{f(\text{mid}(X_0))\}$
  while $\mathcal{L}_\mathcal{W} \neq \emptyset$ do
  {
    select a box X^* from $\mathcal{L}_\mathcal{W}$ and delete X^* from $\mathcal{L}_\mathcal{W}$; *Selection rule*
    bisect X^* perpendicularly to a direction of maximum width $\to X^1$ and X^2;
    for $l = 1, 2$ do
    {
      Store $f(\text{mid}(X^l))$ in $\mathcal{L}_{\mathcal{PNS}}$ and update $\mathcal{L}_{\mathcal{PNS}}$;
      if (X^l cannot be discarded) *Discarding test*
        if (X^l satisfies the termination criteria) *Termination rule*
          Store X^l in $\mathcal{L}_\mathcal{S}$;
        else
          Store X^l in $\mathcal{L}_\mathcal{W}$;
    }
  }
}

The lists $\mathcal{L}_\mathcal{W}$, $\mathcal{L}_\mathcal{S}$ and $\mathcal{L}_{\mathcal{PNS}}$ are respectively the working list, the solution list and a list with known objective values, which are the nondominated points of the objective values of the midpoints of the so far considered boxes. As selection and termination rule we use similar ones to those proposed in [3].

Selection rule: Select the box $X^* \in \mathcal{L}_\mathcal{W}$ with a minimum lower bound of f_1.

This lower bound is calculated by underestimating f_1 within the considered box. We will explain this method later. Certainly it is possible to replace f_1 by f_2.

Termination rule: Store $X \in \mathcal{L}_\mathcal{S}$ if the following condition for given $\varepsilon,\ \delta > 0$ holds: $\omega(X) \leq \delta$ and $\omega(F(X)) \leq \varepsilon$

Our main focus is on the **discarding tests**. We also make use of the monotonicity test from [3]. For the other tests the main idea is to underestimate both objective functions with a convex function. Such a function can be calculated as explained in [6]. For a given box $X = [\underline{x}, \overline{x}]$, a function $g\colon X \to \mathbb{R}$ and a sufficiently large $\alpha > 0$ a convex underestimating function is

$$g_\alpha(x) = g(x) + \frac{\alpha}{2}(\underline{x} - x)^T(\overline{x} - x).$$

For computing α we use interval arithmetic applied to the Hessian matrix of g to derive a lower bound of the smallest eigenvalue on X_0.

A first possibility for a discarding test for a given box $X^l \in \mathbb{R}^n$ is to calculate the convex underestimators for f_1 and f_2 and search for a minimal value in X^l. We denote these minimal values by $a_j,\ j = 1, 2$. The *ideal point* of f_α in X^l is $a := (a_1, a_2)^T$, which is a lower bound for all objective values of f in X^l. The test compares a with some known objective values which are stored in the list $\mathcal{L}_{\mathcal{PNS}}$. If a is dominated by such a point, this point dominates all points of $f(X^l)$. Hence we can discard X^l. The case that X^l cannot be discarded is shown in Figure 1. Furthermore, Figure 1 illustrates a case, where all objective values are dominated by at least one point of $\mathcal{L}_{\mathcal{PNS}}$. But a is not dominated by any of them. Therefore we improve the discarding test. We will use the idea of Benson's outer approximation algorithm ([1], [2]) for convex multi-objective optimization problems and construct supporting hyperplanes. We use only one step of Benson's algorithm and construct only one hyperplane, see Figure 2.

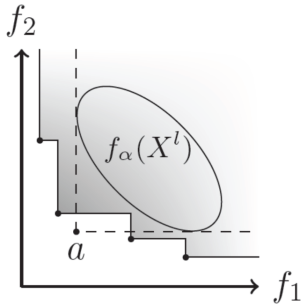


Figure 1. Image set $f_\alpha(X^l)$, points of $\mathcal{L}_{\mathcal{PNS}}$ and ideal point a

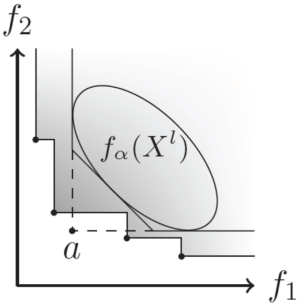


Figure 2. Improved outer approximation of $f_\alpha(X^l)$

How this works is very similar to one step of the algorithm explained in [2]. More precisely we have to solve two optimization problems. The first one is a scalar-valued nonlinear convex problem which computes a point at the boundary of $f_\alpha(X^l)$. Afterwards we solve a simple linear problem to get a normal vector of the supporting hyperplane at the calculated boundary point of $f_\alpha(X^l)$. With arguments of duality theory we are able to prove the correctness of the procedure.
Next we test whether the approximation of $f_\alpha(X^l)$ is dominated by some of the points of $\mathcal{L}_{\mathcal{PNS}}$. If the whole approximation set is contained in the shaded region of Figure 2, we can discard X^l.

3. Numerical results

The algorithm was tested with different options. We use the different discarding tests alone and in combination with each other. Here, we only present those results which compare the new improved discarding test with the method using the ideal point from the literature. We decided to use the monotonicity test at all times, because of its fast eliminating of big boxes. We used different test functions. By way of example consider the two dimensional nonconvex Fonseca-Flemming-function [4]

$$f(x_1, x_2) = \begin{pmatrix} 1 - \exp(-(x_1 - \frac{1}{\sqrt{2}})^2 - (x_2 - \frac{1}{\sqrt{2}})^2) \\ 1 - \exp(-(x_1 + \frac{1}{\sqrt{2}})^2 - (x_2 + \frac{1}{\sqrt{2}})^2) \end{pmatrix}$$

defined on $X_0 = [-4, 4] \times [-4, 4]$. In Figures 3a and 3b the discarded subboxes of X_0 are shaded in grey in different ways. The white set is the union of all boxes from $\mathcal{L}_{\mathcal{S}}$.

The algorithm which used the monotonicity test and the discarding test based on the ideal point needs 385 box divisions. The improved discarding test using the improved outer approximation reduces this to only 315 divisions. As one can see in the figures, the improved

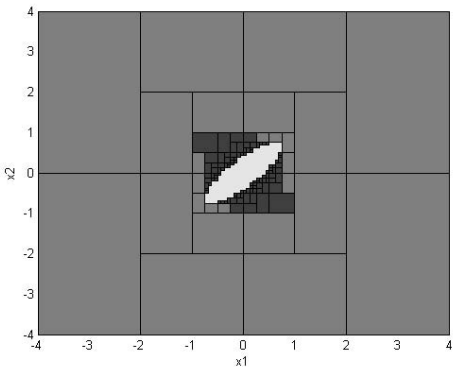


Figure 3a. Feasible region with discarded boxes based on the monotonicity test (grey) and on the ideal point (dark grey)

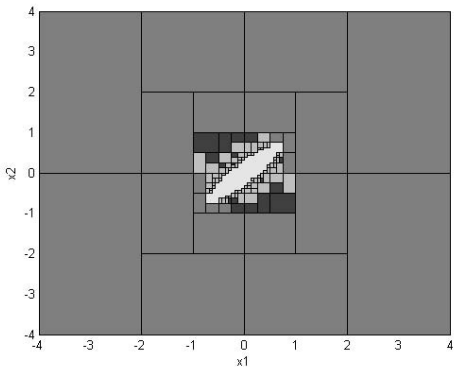


Figure 3b. Feasible region with discarded boxes based on the monotonicity test (grey), on the ideal point (dark grey) and on the improved test (light grey)

test is able to discard much bigger boxes near the solution boxes. Additionally the covering of the efficient set is tighter. These results were very similar for other test functions and demonstrate that the new improved test works well.

4. Outlook

In addition to extending the algorithm to problems with convex constraints we plan to generalize it to more than two objective functions. We also plan to find out in numerical tests whether improved outer approximations of $f_\alpha(X^l)$ gained by more steps of Benson's algorithm improve the speed of the algorithm. Moreover, we want to extend our approaches in view of [5].

Acknowledgments

The first author thanks Thuringian State Graduate Support Regulation for financial support.

References

[1] H. P. Benson. An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple objective linear programming problem. *Journal of Global Optimization*, 13(1):1–24, 1998.

[2] M. Ehrgott, L. Shao, and A. Schöbel. An approximation algorithm for convex multi-objective programming problems. *Journal of Global Optimization*, 50(3):397–416, 2011.

[3] J. Fernández and B. Tóth. Obtaining the efficient set of nonlinear biobjective optimization problems via interval branch-and-bound methods. *Computational Optimization and Applications*, 42(3):393–419, 2009.

[4] C. Fonseca and P. Fleming. Multiobjective genetic algorithms made easy: selection sharing and mating restriction. In *Proceedings 1st International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, pages 45–52, 1995.

[5] A. Löhne, B. Rudloff, and F. Ulus. Primal and dual approximation algorithms for convex vector optimization problems. *Journal of Global Optimization*, 60(4):713–736, 2014.

[6] C. D. Maranas and C. A. Floudas. Global minimum potential energy conformations of small molecules. *Journal of Global Optimization*, 4(2):135–170, 1994.

[7] A. Neumaier. *Interval methods for systems of equations.* Cambridge University Press, 1990.

Proceedings of GOW'16, pp. 61 – 64.

On Achieving a Desired Flux Distribution on the Receiver of a Solar Power Tower Plant *

N.C. Cruz[1], J.L. Redondo[1], J.D. Álvarez[1], M. Berenguel[1] and P.M. Ortigosa[1]

[1] *University of Almería, ceiA3, Dpt. of Informatics* {ncalvocruz,jlredondo,jhervas,beren,ortigosa}@ual.es

Abstract The heliostat field of solar tower power plants must be carefully configured to get the maximum profit of solar energy while keeping the system in a regular operation state. Field control tasks include both deciding what heliostats need to be activated and assigning each one a certain aiming point over the receiver. In fact, current plants have hundreds, even thousands, of available heliostats. In this context, there are desirable flux distributions of the concentrated energy over the receiver that should be achieved to grant an efficient operation while also avoiding thermal stress, premature aging and undesirable temperature gradients over the receiver surface. In this work, a meta-heuristic algorithm is presented to be able to reproduce any desired flux distribution over the receiver, what implies solving a large-scale optimization problem. It selects both the subset of active heliostats and their corresponding aim points for a given operational instant on minimizing an error function.

Keywords: Global Optimization, Heliostat field flux distribution, Solar power tower plant, Aiming strategy

1. Introduction

Solar tower power plants (STPP) are one of the most interesting facilities to generate large-scale clean electricity due to their overall efficiency, their mature technological basis and their relative stability of production. This kind of systems mainly consist of a set of steerable highly-reflectance mirrors with solar tracking capabilities, known as 'heliostats', which are responsible for concentrating the incident solar radiation over a receiver along the day. Then, the concentrated energy over the receiver is transferred to a working fluid (the heat transfer fluid (HTF)) in circulation, whose temperature gets increased, and can be used for electrical energy generation on a classic thermodynamical cycle. In Figure 1 a basic schema of an STTP is shown. The interested reader is referred to [2, 7] for further information of this technology.

The operative field of modern STPP is generally formed by a vast set of heliostats as it is commonly over-dimensioned to face unfavorable operating conditions such as cloudy days. However, not all of them need to be operated for the nominal case to achieve the expected power requirements. In fact, the receiver should not be exposed to an excessive or uncontrolled income of power over its surface. The flux distribution of the reflected solar radiation over the receiver must be controlled to avoid dangerous temperature gradients, thermal stress and premature aging of its components [1, 3, 5, 8]. This is a key factor for increasing the operative life of the receiver, which has a direct influence on the production costs of STTP as highlighted in [5]. The flux distribution over the receiver is a direct consequence of which heliostats are active and to which aiming point they are targeting to. In this context, it is necessary to face a two-layered optimization problem in which it must be decided both the subset of available heliostats to activate and their corresponding aiming point at the receiver. These

*This work has been funded by grants from the Spanish Ministry of Economy and Competitiveness (TIN2015-66680-C2-1-R and ENERPRO DPI 2014-56364-C2-1-R), Junta de Andalucía (P11-TIC7176 and P12-TIC301). Nicolás Calvo Cruz is supported by a FPU Fellowship from the Spanish Ministry of Education. Juana López Redondo and José Domingo Álvarez Hervás are fellows of the Spanish 'Ramón y Cajal' contract program, co-financed by the European Social Fund

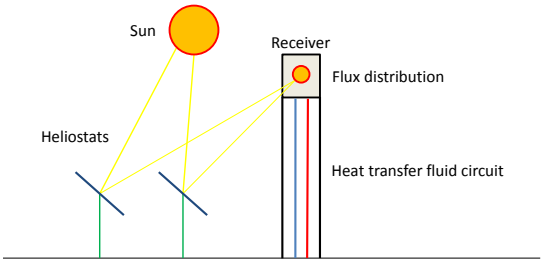


Figure 1. Scheme of a solar tower power plant.

tasks are usually supported by manual human decisions, what is an implicit limitation of the number of aiming points that can be handled and the adaptability of the field.

In the recent works of [3, 5], two similar optimization problems related to predefined aiming points assignation are addressed. They are focused on minimizing the standard deviation of the flux density distribution and flux spread minimization respectively. In [3] a Genetic Algorithm is successfully used while in [5] a TABU search is applied with good results. However, this work aims two define and solve a two-layered optimization problem in which both the subset of heliostats to activate and their corresponding aiming points are optimized to achieve any user-given flux distribution (instead of being linked to static general ideas such as flux spread minimization) by minimizing an error function. Additionally, the aiming point assignment is expanded to a continuous search-space. By proceeding this way, the field would be significantly more adaptable and configurable. We have been working with a meta-heuristic algorithm for heliostat selection and a local gradient-based search procedure for final aiming points assignation with promising results. In the next section, we introduce the mathematical formulation of the problem at hand. Then, the optimization procedure is summarized. Finally, some experimental results are shown and conclusions are drawn.

2. Mathematical formulation

As commented in the previous section, the key idea of the present problem can be summarized in this sentence: the intention is to replicate a desired flux distribution over a flat plane receiver by selecting both a subset of heliostats to be activated and their corresponding aiming points over the receiver. This idea leads to face a complex large-scale optimization problem.

In order to model this problem, we can start by defining the whole heliostat field as an ordered set $H = \{h_1, h_2, \cdots, h_N\}$ with cardinality N. The reference flux distribution to be achieved in a certain common instant t is defined by a two-dimensional function F which expresses, for any point (x, y) on the receiver plane, the radiation density (kW/m^2) at that point. We consider the X and Y directions to be positive towards the East and North respectively over the plane of the receiver, which is due North. Every heliostat h projects a certain flux distribution f_h over the receiver when it is operative, which is also a known two-dimensional function of the radiation density. A certain candidate solution C can be seen as an ordered sequence of length N with the structure $C = \{c_1, c_2 \cdots, c_N\}$. In C, the position of every element is directly mapped to the corresponding heliostat in H so c_h defines the particular configuration of the heliostat h, which can be $\emptyset$ when it is not active or a certain position (x, y) on the receiver plane. Therefore, there are $2N^*$ variables under optimization at the second layer of the problem, where N^* is the number of finally active heliostats. In this context, a certain valid field configuration C defines the corresponding achieved flux distribution F^* over the receiver, which is formed by the convolution of every sub-flux distribution f_h (discarding non-active heliostats). Then, the objective function of the problem at hand can be defined as the difference between the reference and the achieved flux distributions $O = |F - F^*(C)|$. Consequently, the optimization problem is defined, from a minimization perspective, as:

$$min\, O = min\, |F - F^*(C)| \tag{1}$$

Assuming that the flux distribution expressions are assumed to be continuous, Eq. 1 implies a de facto discretization for both the reference and the achieved flux distributions, that can be seen as monochromatic images, in order to study their differences. Therefore, after defining a discretization grid over the receiver plane, the problem can be formulated as

$$min\, O = min \sum_{x=X_1}^{X_T} \sum_{y=Y_1}^{Y_T} |F(x,y) - F^*(C)(x,y)| \tag{2}$$

where $\{X_1, \cdots, X_T\}$ and $\{Y_1, \cdots, Y_T\}$ are the discrete sets of X and Y coordinates on the receiver respectively. However, both sets can be defined with different cardinalities.

In relation to the definition of every heliostat-linked flux f_h over the receiver plane, we work with the analytical definition of a bi-dimensional Gaussian density function to model it:

$$f_h(x,y) = \frac{P}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} e^{\left(-\frac{1}{2(1-\rho^2)}\left(\frac{(x-\mu_x)^2}{\sigma_x^2} + \frac{(y-\mu_y)^2}{\sigma_y^2} - \frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y}\right)\right)} \tag{3}$$

where x and y are the coordinates on the plane defined by the receiver rectangular aperture, P is the power contribution of the heliostat h over the receiver, ρ is the correlation between x and y, σ_x and σ_y are the standard deviation along x and y respectively and μ_x and μ_y, which are the mean in the Gaussian probability function, define the central point of the flux distribution, i.e. the aiming point of the heliostat h. This approach is similar to the one selected by [3, 5], where a specific circular Gaussian density function is applied according to the HFLCAL model [6].

As previously commented, the flux information of every heliostat needs to be known, what is usually achieved by using CPU-time demanding ray-tracing or convolution-based simulations as done in [3, 5]. However, for this work, a synthetic fluxes database has been generated to be used as a plain input for the optimization procedure. Finally, the reference flux distribution to achieve, F, can be also defined by using Eq. 3 or any other user-given bi-dimensional expression. The most recommendable testing approach is to form the reference flux by convolving a known sub-set of existing heliostat. By proceeding this way it is known that it is possible to achieve the reference flux with the deployed heliostats.

3. Optimization procedure and preliminary results

The algorithm starts by solving the first layer of the problem, what determines an active subset of heliostats and their initial starting points, according to the reference flux, for the second stage. Then, a local gradient-based optimizer is used to sharpen the selected aiming point of every heliostat to minimize the error between the reference and the obtained flux maps.

At its first stage, the algorithm generates different candidate configurations C and looks for the most promising one until the termination criteria are satisfied (i.e., an user-defined number of cycles or a certain threshold). There are two initial solutions that are always considered in this procedure: the sets formed by the most and less number of available heliostats to achieve the total power in the reference flux (with independence of its shape). These solutions fix two thresholds for any other future candidate solution: its number of active heliostats must be between the two initial. Then, new candidate solutions are formed by mutating and mixing the existing ones along a search procedure to minimize the objective function O. It must be noted that the aiming points of candidate solutions at the first stage, which are used for relative evaluations, are assigned according to the power of the heliostats and the shape of the reference flux. Then, the second stage of the algorithm takes the best solution obtained from the first part of the search and tries to improve its quality by applying a gradient-based local search while minimizing O until the termination criteria are fulfilled (i.e., an user-defined

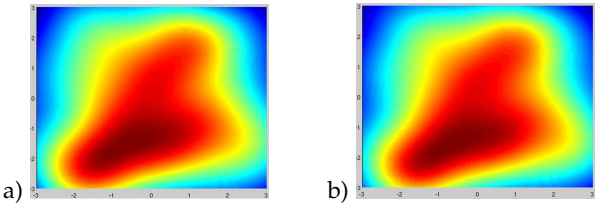


Figure 2. Result of flux distribution replication of the reference in a) is shown in b).

number of cycles or a certain threshold). At this point, the active heliostats subset will not been changed any more but only their corresponding aiming points.

In Figure 2 a result for the local optimizer is shown. In that test, the reference flux has been formed by randomly convolving 50 heliostats over a 6x6 receiver and defining the reference flux in Figure 2 a). The local optimizer has been able to replicate the desired map by optimizing the aim point of the original 50 active heliostats from a totally random start as shown in Figure 2 b). Additional experiments have been carried out up to 200 heliostats with positive results.

4. Conclusions

A generic two-layered optimization procedure is being developed for STPP which is intended to be able to configure the heliostat field to achieve a given distribution flux by selecting both the active heliostats and their aiming points. The preliminary results are promising, i.e. the flux distribution replication for a defined active set is operative with a good overall performance. For future work, the applicability of the procedure in control tasks could be studied.

Acknowledgments

The authors wish to thank Alberto Flores Martínez and Ramón Ferri García for their support.

References

[1] M. Becker and M. Carrasco. Solar Thermal Central Receiver Systems: Performance evaluation standards for solar central receivers. *Springer Verlag*, vol. 3, 1991.

[2] O. Behar, A. Khellaf and K. Mohammedi. A review of studies on central receiver solar thermal power plants. *Renewable and Sustainable Energy Reviews*, 23:12–39, 2013.

[3] S.M. Besarati, D.Y. Goswami and E.K. Stefanakos. Optimal heliostat aiming strategy for uniform distribution of heat flux on the receiver of a solar power tower plant. *Energy Conversion and Management*, 84:234–243, 2014.

[4] F.M.F. Siala and M.E. Elayeb. Mathematical formulation of a graphical method for a no-blocking heliostat field layout. *Renewable energy*, 23(1):77-92, 2001.

[5] A. Salomé, F. Chhel, G. Flamant, A. Ferrière and F. Thiery. Control of the flux distribution on a solar tower receiver using an optimized aiming point strategy: Application to THEMIS. *Solar Energy*, 94:352–366, 2013.

[6] P. Schwarzbözl, M. Schmitz and R. Pitz-Paal. Visual HFLCAL - A Software Tool for Layout and Optimisation of Heliostat Fields. In SolarPACES'09, 2009.

[7] W. Stine and M. Geyer. Power from the sun, 2001. Public website, available from http://powerfromthesun.net/book.html (Last access: March 12, 2016)

[8] C.J. Winter, R.L. Sizmann and L.L. Vant-Hull. Solar power plants: fundamentals, technology, systems, economics. *Springer-Verlag New York*, 1991.

Proceedings of GOW'16, pp. 65 – 68.

The Alternating Direction Column Generation Method (Global Optimization without Branch-and-Bound)

Ivo Nowak[1] and Norman Breitfeld[2]

[1]*Hamburg University of Applied Sciences, Berliner Tor 21, 20099 Hamburg, Germany,* ivo.nowak@haw-hamburg.de

[2]*Technische Universität Berlin, Marchstraße 18, 10587 Berlin, Germany,* norman.breitfeld@win.tu-berlin.de

Abstract In this paper, we present a new decomposition method, called ADCG (Alternating Direction Column Generation) method, for globally solving quasi-separable nonconvex MINLPs, which is not based on the branch-and-bound approach. The basic idea of ADCG is to restrict the feasible set by an upper bound of the objective value and to check via a (column generation based) globally convergent alternating direction method if the resulting MINLP is feasible or not. Convergence of ADCG to a global solution is shown by using the fact that the duality gap of a general nonconvex projection problem is zero (in contrast to the Lagrangian dual of a general nonconvex program). We discuss algorithmic details, like checking the infeasibility of the projection problem and updating the target value and describe how ADCG can be accelerated by an inexact sub-problem solver. Furthermore, we will report first numerical results with a preliminary implementation of ADCG using Pyomo.

Keywords: Decomposition, Nonconvex optimization, Column generation, Alternating direction method

1. Introduction

We consider a general *quasi-separable* (or *block-separable*) (convex or nonconvex) MINLP of the form:

$$\min\{c(x) : x \in G \cap P\} \tag{1}$$

with

$$\begin{aligned} P &:= \{x \in [\underline{x}, \overline{x}] : Ax \leq b\} \\ G &:= \bigtimes_{k \in K} G_k \\ G_k &:= \{y \in [\underline{x}_{I_k}, \overline{x}_{I_k}] : y_i \in \{0,1\},\ i \in I_k^{\text{int}},\ g_j(y) \leq 0,\ j \in J_k\} \end{aligned}$$

where $c(x) := \langle c, x \rangle$ is a linear objective function and the matrix $A \in \mathbb{R}^{m \times n}$, the vector $b \in \mathbb{R}^m$ specify the linear *coupling constraints*, $I_k^{\text{int}} \subseteq I_k \subset [n]$, $J_k \subset J$ and $x_{I_k} := (x_i)_{i \in I_k}$ denotes the subvector of x. The (possibly nonconvex) functions $g_j : \mathbb{R}^{n_k} \to \mathbb{R}$ with $j \in J_k$ specify the nonlinear constraints of the problem. The restriction to inequality constraints is only for notational simplicity. Note that a general sparse MINLP can be reformulated as a quasi-separable optimization problem by adding new variables and copy-constraints, see, e.g., [10].

Most exact algorithms for solving the MINLP (1) are based on the branch-and-bound approach and variants like branch-cut-and-price [5] or branch-decompose-and-cut [13], see [4, 3] for an overview of MINLP-solvers. A main difficulty of this approach is a possibly rapidly growing branch-and-bound tree, which makes it difficult to solve large-scale models in reasonable time.

Parallel decomposition methods, such as Column Generation (CG), can solve huge MINLPs with up to several hundred millions of variables if the duality gap is not too large, e.g. in crew and transport planning [1, 11]. However, for many MINLPs the duality gap is not small, and in this case traditional decomposition methods may be not efficient [10].

In this paper, we present a recently proposed decompositon method, called ADCG (Alternating Direction Column Generation) method, for globally solving MINLPs (1) with a possibly large duality gap [12]. Since ADCG is a proximal optimization method, which is not based on the branch-and-bound approach, the generation of a (possibly huge) branch-and-bound tree is avoided. Furthermore, it is possible to use fast inexact MINLP-solvers for solving the sub-problems, and the sub-problems can be solved in parallel. Since processors with many cores are available, many sub-problems can be solved simultaneously.

2. Alternating direction methods

The Alternating Direction Method (ADM) was originally developed for finite element problems [6] and is based on Uzawas algorithm [14]. A review of ADMs including a convergence proof for convex problems is given in [2]. An ADM for solving MIPs is presented in [9]. A simplified ADM is shown in Figure 1.

1. $y^i \leftarrow$ project x^{i-1} onto G regarding $c + A^T\lambda^{i-1}$
2. $x^i \leftarrow$ project y^i onto P regarding $c - A^T\lambda^{i-1}$
3. $\lambda^i \leftarrow$ update λ^{i-1}

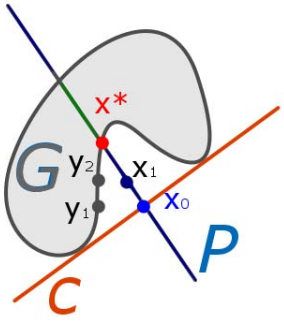


Figure 1. Basic steps of an ADM for solving $\min\{c(x) : x \in G \cap P\}$ starting from a point $x^0 \in P$ (solution of a convex relaxation). The trial points x^i and y^i converge towards the solution point x^*.

The projection of a point $x^{i-1} \in P$ onto G is performed by solving the separable program:

$$y^i = \underset{y \in G}{\operatorname{argmin}}\, c(y) + (\lambda^{i-1})^T Ay + \rho \sum_{k \in K} \|y - x^{i-1}_{I_k}\|^2_{\Sigma_k} \tag{2}$$

where Σ_k is a positive definite scaling matrix. Problem (2) decomposes into low-dimensional sub-problems, which can be solved in parallel. The projection of a point $y^i \in G$ onto P is performed by solving the QP-master-problem

$$x^i = \underset{x \in P}{\operatorname{argmin}}\, c(x) - (\lambda^{i-1})^T Ax + \rho \sum_{k \in K} \|A_k(y^i_{I_k} - x_{I_k})\|^2_2 \tag{3}$$

where $Ax = \sum_{k \in K} A_k x_{I_k}$.

Traditional ADMs update λ^i by a subgradient step and are in general not globally convergent, see [8] for an example.

A globally convergent ADM for (locally) solving quasi-separable nonconvex NLPs, called ALADIN (Augmented Lagrangian based Alternating Direction Inexact Newton method), is proposed in [8]. This method updates λ^i by performing a line search for increasing a dual function of the following nonconvex projection problem at an iteration point $x^{i-1} \in P$:

$$\min\{c(y) + \rho \sum_{k \in K} \|y_{I_k} - x^{i-1}_{I_k}\|^2_{\Sigma_k} : y \in G \cap P\} \tag{4}$$

Global convergence of ALADIN is proved in [8] by using the fact that the duality gap of the Lagrangian dual of (4) regarding the coupling constraints is zero (in contrast to the Lagrangian dual of (1)).

3. The alternating direction column generation method

Motivated by the approach of [8] and by the excellent performance of column generation methods for solving huge network optimization problems [1, 11], the new method, called Alternating Direction Column Generation method (ADCG), combines column generation and the ADM of [8] for globally solving nonconvex MINLPs, see [12] for details.

ADCG is based on adding the target constraint $c(x) \leq \sigma$ to the polyhedron P, denoted by P_σ, and updating the target value σ by increasing σ if $G \cap P_\sigma$ is infeasible and decreasing σ otherwise. In order to make an efficient warm-start after updating σ possible and to check the feasibility of $G \cap P_\sigma$, the dual line search of [8] for approximately computing dual solutions of (4) is replaced by a column generation method.

The column generation master-problem is given by

$$\begin{aligned} \min \quad & c(y) + \rho \cdot \sum_{k \in K} r_k + \gamma \cdot s \\ & Ay \leq b \\ & c(y) \leq \sigma + s \\ & (y_{I_k}, r_k) \in \operatorname{conv}(\bar{S}_k), \quad k \in K \\ & s \geq 0 \end{aligned} \tag{5}$$

where s is a slack-variable. The (extended) column pool $\bar{S}_k := \{(\hat{y}, \|\hat{y} - x^{i-1}_{I_k}\|^2_{\Sigma_k}) : \hat{y} \in S_k\}$ is defined by the sample set $S_k \subseteq G_k$ consisting of previously generated trial points.

The feasible set $G \cap P_\sigma$ is empty, if and only if after convergence of the column generation method some slack variables s of the solution of (5) are not zero. Assuming that the sub-problems (2) are solved to global optimality, it is proved in [12] that algorithm ADCG terminates in finitely many steps with a global ϵ-minimizer of (1) under mild conditions.

ADCG is a proximal descent method, which computes a solution in the neighborhood of the starting point (because in each iteration a descent point in the neighborhood of the current trial point is computed by approximately solving a projection problem). This makes the method more robust regarding (near) symmetric optimization problems with many ϵ-optimal solutions than branch-and-bound methods, which often need to perform many branching steps, if the optimization problem has many ϵ-optimal solutions.

4. pyADCG

We are currently working on a Python-implementation of ADCG using Pyomo [7], called pyADCG. An initial point $x^0 \in P$ and inner and outer approximations of G for starting ADCG are computed by

1. calculating a point $\hat{x} \in \operatorname{conv}(G \cap P)$ using traditional column generation
2. driving $\hat{x}$ towards $G \cap P$ using a simplified version of ADCG

Numerical results using pyADCG will be presented. Since solving sub-problems in pyADCG is not much more difficult than solving traditional pricing problems, we expect that the computational cost of pyADCG will be similar to a traditional column generation method.

References

[1] R. Borndörfer, A. Löbel, M. Reuther, T. Schlechte, and S. Weider. Rapid branching. *Public Transport*, 5:3–23, 2013.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3:1–122, 2011.

[3] S. Burer and A. Letchford. Non-convex mixed-integer nonlinear programming: A survey. *Surveys in Operations Research and Management Science*, 17 (2):97–106, 2012.

[4] M. R. Bussieck and S. Vigerske. *MINLP Solver Software*, 2014.
`www.math.hu-berlin.de/~stefan/minlpsoft.pdf`

[5] J. Desrosiers and M. Lübbecke. Branch-price-and-cut algorithms. In J. Cochran, L. Cox, P. Keskinocak, J. Kharoufeh, and J. Smith, editors, *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Inc., 2010.

[6] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximations. *Computers and Mathematics with Applications*, 2:17–40, 1976.

[7] W. E. Hart, C. Laird, J.-P. Watson, and D. L. Woodruff. *Pyomo–optimization modeling in python*, volume 67. Springer Science & Business Media, 2012.

[8] B. Houska, J. Frasch, and M. Diehl. An Augmented Lagrangian based Algorithm for Distributed Non-Convex Optimization. *SIAM Journal of Optimization*, 26(2):1101–1127, 2016.

[9] B. Geißler, A. Morsi, L. Schewe, and M. Schmidt. Solving Power-Constrained Gas Transportation Problems using an MIP-based Alternating Direction Method. *Computers & Chemical Engineering*, 82: 303–317, 2015.

[10] I. Nowak. *Relaxation and Decomposition Methods for Mixed Integer Nonlinear Programming*. Birkhäuser, 2005.

[11] I. Nowak. *Parallel Decomposition Methods for Nonconvex Optimization - Recent Advances and New Directions*, In: *Proceedings of Mathematical and Applied Global Optimization (MAGO 2014), XII Global Optimization Workshop 2014*, Málaga, Spain, pp. 73–76, 2014.

[12] I. Nowak. *Column Generation based Alternating Direction Methods for solving MINLPs*, 2015.
`www.optimization-online.org/DB_HTML/2015/12/5233.html`

[13] T. Ralphs and M. Galati. Decomposition and dynamic cut generation in integer linear programming. *Mathematical Programming*, 106(2):261–285, 2006.

[14] H. Uzawa. Iterative methods for concave programming. Studies in Linear and Nonlinear Programming, pages 154–165. 1958.

Proceedings of GOW'16, pp. 69 – 72.

Optimization of Enzymes Inactivation in High Pressure Processes *

Miriam R. Ferrández[1], Juana L. Redondo[1], Benjamin Ivorra[2], Ángel M. Ramos[2] and Pilar M. Ortigosa[1]

[1] *University of Almería, Agrifood Campus of International Excellence, ceiA3, Almería, Spain*
{mrferrandez,jlredondo,ortigosa}@ual.es

[2] *Complutense University of Madrid, Madrid, Spain* {ivorra,angel}@mat.ucm.es

Abstract The demand of highest quality foods in terms of taste and their properties preservation without the use of additives is constantly increasing. Consequently, new approaches to food processing have been developed, as for example high-pressure technology which has proven to be very valuable because it allows to maintain good properties of food like some vitamins and, at the same time, to reduce some undesirable bacteria. This technology avoids the use of high temperatures during the process (not like Pasteurization), which may have adverse effect on some nutritional properties of the food, its flavour, etc. The models for some enzymatic inactivations, which depend on the pressure and temperature profiles are presented. This work deals with the optimization of the inactivation of certain enzymes when high pressure treatment on food processing is applied. The optimization algorithms will minimize the inactivation not only of a certain isolated enzyme but also to several enzymes that can be involved simultaneously in the high-pressure process.

Keywords: High Pressure Processing, Global Optimization, Metaheuristic algorithms, Food Technology

1. Introduction

Nowadays, the demand of safe and minimally processed food prepared for immediate consumption (ready–to–use and ready–to–eat) has increased significantly. High Pressure (HP) processing is one of the technologies that can be used for the preparation of these products [4]. This technique is very effective in prolonging the shelf life of some foods and it is already being applied to industry, though without the use of the optimal configurations of pressure and temperature.

In this work we model the effect of the combination of high pressure and thermal treatments on food processing, focussing on the inactivation of certain enzymes [2, 3]. It can be considered that during the food processing two coupled physical phenomena take place: on the one hand, the evolution of enzymatic activity, which has been described by a first–order equation, and, on the other hand, the thermal problem given by a partial differential equation. Then, the optimization problem involves a different Eyring-Arrhenius equation for modelling each enzymatic inactivation and a heat transfer equation. Considering that for each particular kind of food and high pressure equipment the problem can be different as the most important enzymes to be inactivated, in this work we only focus on solving the specific cases of Vitamin C [9] and bacteria BSAA [1]. In particular, our aim is to optimize the high-pressure process such that the activity of the bacteria BSAA is minimized and the Vitamin C is maximized. For

*This work has been funded by grants from the Spanish Ministry of Economy and Competitiveness (TIN2015-66680-C2-1-R, MTM2011-22658 and MTM2015-64865); Junta de Andalucía (P11-TIC7176 and P12-TIC301), in part financed by the European Regional Development Fund (ERDF); and the research group MOMAT (Ref. 910480) supported by "Banco de Santander" and "Universidad Complutense de Madrid". Juana López Redondo is a fellow of the Spanish "Ramón y Cajal" contract program, co-financed by the European Social Fund.

solving these problems, a metaheuristic hybrid method based on a population of solutions that evolve toward the optimum is applied. The hybridized method uses a gradient-based local search that can accelerates the convergence to the optima.

2. Mathematical model

When HP is applied in Food Technology, it is necessary to take into account the thermal effects that are produced by variations of temperature due to the compression/expansion that takes place in both the food sample and the pressurizing fluid.

Often, in HP processing, the packed food is surrounded by the pressurizing fluid inside a cylindrical steel pressure vessel. It can be assumed that thermally induced flow instabilities are almost negligible. Due to axial symmetry of the problem, we can consider reducing its domain to only half a cross section. Therefore, we use cylindrical coordinates such that our computational domain is the rectangle $\Omega = [0, L] \times [0, H]$, where four two–dimensional sub–domains are distinguished (see Figure 1): the one that contains the food sample, denoted by Ω_F; the cap of the sample holder (usually it is made of rubber), Ω_C; the domain occupied by the pressurizing medium, Ω_P, and the one of the steel surrounding the previous domains, Ω_S.

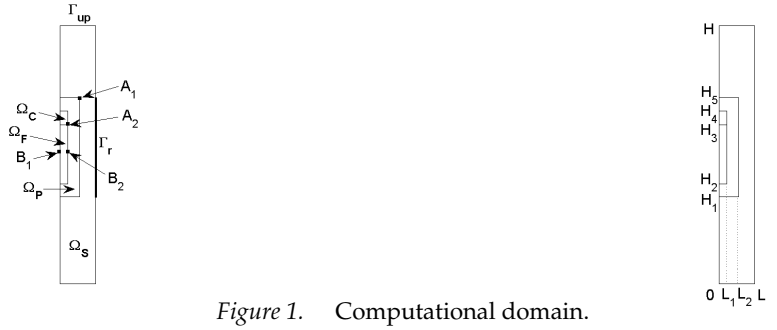


Figure 1. Computational domain.

2.1 Enzymatic inactivation

The evolution of the activity A of an enzyme is described by the following first–order kinetic equation:

$$\frac{dA}{dt}(t) = -\kappa(P(t), T(t))\, A(t), \tag{1}$$

where t is the time (min), $P(t)$ is the pressure (MPa) at time t, $T(t)$ is the temperature (K) at time t, $\kappa(P, T)$ is the inactivation rate (min^{-1}) corresponding to the pressure–temperature conditions given by (P, T) and $A(t)$ is the activity of the enzyme under study.

The effect of temperature and pressure on the inactivation rate is provided by a suitable combination of Arrhenius equation and Eyring equation [1]:

$$\kappa(P, T) = \kappa_{\text{ref}} \exp\left(-B\left(\frac{1}{T} - \frac{1}{T_{\text{ref}}}\right)\right) \exp\left(-C(P - P_{\text{ref}})\right), \tag{2}$$

where T_{ref} is a reference temperature (K), P_{ref} is a reference pressure (MPa) and κ_{ref} is the inactivation rate at reference conditions (min^{-1}). The parameters B and C express the temperature dependence of κ (K) and the pressure dependence of κ (MPa^{-1}), respectively.

In particular, we use equations (1) and (2) to model the behaviour of each of the following enzyme and vitamin with appropriate parameters:

- **Bacillus Subtilis α–Amylase (BSAA):** It is produced by a bacteria called Bacillus Subtilis, present in the ground. This enzyme can contaminate food and modify its taste [1].

- **Vitamin C:** It is present in some fruits and it has good antioxidant properties. Rapid degradation of this vitamin is produced at high temperatures. Then, HP processing is being used as preservation technique [9].

2.2 Heat transfer modelling

We focus on solid type foods processed under HP treatment in a cylindrical pressure vessel. We consider that the filling ratio of the food sample inside the vessel is much higher than the one of the pressurizing medium. Under these assumptions, the heat transfer by conduction model is suitable [6]:

$$\rho C_p \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) = \alpha \frac{dP}{dt} T \text{ in } \Omega \times (0, t_{\mathrm{f}}), \tag{3}$$

where $\rho = \rho(T, P)$ is the density ($\mathrm{Kg\,m^{-3}}$), $C_p = C_p(T, P)$ is the heat capacity ($\mathrm{J\,Kg^{-1}K}$), $k = k(T, P)$ is the thermal conductivity ($\mathrm{W\,m^{-1}K^{-1}}$) and t_{f} is the final time (s). $P = P(t)$ is the pressure (Pa) applied by the equipment (this is chosen by the user within the machine limitations) and $\alpha = \alpha(T, P)$ is the thermal expansion coefficient ($\mathrm{K^{-1}}$) of the food or the pressurizing fluid, depending on the domain Ω_{F} or Ω_{P}, respectively.

The boundary conditions are: on the refrigeration boundary, $\Gamma_{\mathrm{r}} \subset \{L\} \times \{0, H\}$ (see Figure 1), the temperature is set to a given refrigeration temperature, T_{r}; on the upper boundary, $\Gamma_{\mathrm{up}} = \{0, L\} \times \{H\}$, heat is transferred by convection with the room; the rest, $\Gamma \setminus \{\Gamma_{\mathrm{r}} \cup \Gamma_{\mathrm{up}}\}$, are the symmetry axis and the equipment walls which are isolated, so the heat flux in both is zero.

Therefore, the following 2D problem results:

$$\begin{cases} \rho C_p \dfrac{\partial T}{\partial t} - \dfrac{1}{r}\dfrac{\partial}{\partial r}\left(rk\dfrac{\partial T}{\partial r}\right) - \dfrac{\partial}{\partial z}\left(k\dfrac{\partial T}{\partial z}\right) = \alpha \dfrac{dP}{dt} T & \text{in} \quad \Omega \times (0, t_{\mathrm{f}}), \\ k\dfrac{\partial T}{\partial \mathbf{n}} = 0 & \text{on} \quad \Gamma \backslash (\Gamma_{\mathrm{r}} \cup \Gamma_{\mathrm{up}}) \times (0, t_{\mathrm{f}}), \\ k\dfrac{\partial T}{\partial \mathbf{n}} = h(T_{\mathrm{amb}} - T) & \text{on} \quad \Gamma_{\mathrm{up}} \times (0, t_{\mathrm{f}}), \\ T = T_{\mathrm{r}} & \text{on} \quad \Gamma_{\mathrm{r}} \times (0, t_{\mathrm{f}}), \\ T(0) = T_0 & \text{in} \quad \Omega, \end{cases} \tag{4}$$

where $\mathbf{n}$ is the outward unit normal vector in the boundary of the domain, T_0 is the initial temperature, T_{amb} is the ambient temperature and h ($\mathrm{W\,m^{-2}K^{-1}}$) is the heat transfer coefficient.

3. Optimization algorithms and preliminary results

Our goal is to optimize the high-pressure process such that the activity of the bacteria BSAA has been minimized and the Vitamin C has been maximized. The optimization parameter vector comprises the initial temperature, the refrigeration temperature and the pressure provided to the equipment. The objective function uses the solution of the previous mathematical model. It returns the final concentration values of bacteria and vitamin obtained under the HP processing described by these parameters.

As a first approximation, we study several mono–objective problems derived from the original: to minimize the final bacteria concentration or to maximize the vitamin, separately. Also, we add some constraints and minimize the bacteria maintaining the vitamin concentration above 97% or maximize the vitamin keeping the bacteria below 40%. Even some temperature restrictions are considered.

An hybrid global optimization algorithm based on Controlled Random Search (CRS) method (see [5] and [8]) has been designed for these mono–objective problems. The final hybrid

method is a combination of three methods: a multi–layer line search [7], CRS and a Steepest Descent (SD) algorithm as local optimizer. The first provides a good initial population and avoids CRS gets stuck in a particular region of the admissible space without exploring others. The local optimizer (SD) improves the accuracy of the solutions and accelerates the convergence of the population.

For each individual, its objective function value is compared with the best element found by CRS algorithm at the previous iteration of the multi-layer method. If this CRS solution gives a better objective function value, a new individual is generated close to it using the secant method. Otherwise, the new individual for the population is created also by the secant method but in another region of the admissible space far enough to explore another solutions.

After analyzing the preliminary results it can be shown that the hybrid algorithm improves the results and the computational cost of the CRS algorithm when it is used isolated and it is able to solve the proposed problems with almost 100% success.

4. Summary

In this work, it is optimized the high-pressure process where the activity of the bacteria BSAA is minimized and the Vitamin C is maximized. A metaheuristic hybrid method has been devised to this aim. The preliminary results are promising, i.e. for the instances considered, the proposed algorithm achieves almost 100% success.

References

[1] Siegfried Denys, Linda R Ludikhuyze, Ann M Van Loey, and Marc E Hendrickx. Modeling conductive heat transfer and process uniformity during batch high-pressure processing of foods. *Biotechnology Progress*, 16(1):92–101, 2000.

[2] Siegfried Denys, Ann M Van Loey, and Marc E Hendrickx. A modeling approach for evaluating process uniformity during batch high hydrostatic pressure processing: combination of a numerical heat transfer model and enzyme inactivation kinetics. *Innovative Food Science & Emerging Technologies*, 1(1):5–19, 2000.

[3] Christoph Hartmann and Antonio Delgado. Numerical simulation of thermal and fluiddynamical transport effects on a high pressure induced inactivation. *Simulation Modelling Practice and Theory*, 13(2):109–118, 2005.

[4] Rikimaru Hayashi, Yukio Kawamura, Teruko Nakasa, and Osamu Okinaka. Application of high pressure to food processing: pressurization of egg white and yolk, and properties of gels formed. *Agricultural and Biological Chemistry*, 53(11):2935–2939, 1989.

[5] Eligius MT Hendrix, Pilar M Ortigosa, and Inmaculada García. On success rates for controlled random search. *Journal of Global Optimization*, 21(3):239–263, 2001.

[6] Juan Antonio Infante, Benjamin Ivorra, Ángel Manuel Ramos, and Jose Maria Rey. On the modelling and simulation of high pressure processes and inactivation of enzymes in food engineering. *Mathematical Models and Methods in Applied Sciences*, 19(12):2203–2229, 2009.

[7] Benjamin Ivorra, Bijan Mohammadi, and Ángel Manuel Ramos. A multi-layer line search method to improve the initialization of optimization algorithms. *European Journal of Operational Research*, 247(3):711–720, 2015.

[8] Wyn L. Price. A controlled random search procedure for global optimisation. *The Computer Journal*, 20(4):367–370, 1977.

[9] Lise Verbeyst, Ragne Bogaerts, Iesel Van der Plancken, Marc Hendrickx, and Ann Van Loey. Modelling of vitamin C degradation during thermal and high-pressure treatments of red fruit. *Food and Bioprocess Technology*, 6(4):1015–1023, 2013.

Proceedings of GOW'16, pp. 73 – 76.

An Integrality Gap Minimization Heuristic for Binary Mixed Integer Nonlinear Programming

Wendel Melo[1], Marcia Fampa[1] and Fernanda Raupp[2]

[1]*Federal University of Rio de Janeiro, Brazil,* wendelalexandre@gmail.com, fampa@cos.ufrj.br

[2]*National Laboratory for Scientific Computing, Brazil,* fernanda@lncc.br

Abstract In this work, we present a new heuristic procedure to find feasible points for binary mixed integer nonlinear programming problems. Using a branch-and-bound scheme, our approach solves, at each explored node of the tree, at least one nonlinear programming subproblem that minimizes the integrality gap.

Keywords: Mixed integer nonlinear programming, Heuristic, Branch and bound, Integer gap minimization.

1. Introduction

In this work, we address the Binary Mixed Integer NonLinear Programing (BMINLP) problem:

$$(P) \quad \begin{array}{ll} \min\limits_{x,\,y} & f(x,y) \\ \text{subject to:} & g(x,y) \leq 0, \\ & x \in X,\ y \in Y \cap \{0,\,1\}^{n_y}, \end{array} \tag{1}$$

where X and Y are polyhedral subsets of $\mathbb{R}^{n_x}$ and $\mathbb{R}^{n_y}$, respectively. The functions $f : X \times Y \to \mathbb{R}$ and $g : X \times Y \to \mathbb{R}^m$ are convex and twice continuously differentiable.

Finding a feasible solution for an optimization problem with combinatorial nature can constitute a difficult task. In many cases, exact algorithms are not able to provide a feasible solution in reasonable time. Thus, researches on new feasibility heuristics are of fundamental importance, since they can be very useful for: (a) speeding up the convergence of exact algorithms; (b) allowing the application of local search algorithms; (c) providing feasible solutions quickly to real problems being represented in an optimization model. In this work, we propose a new feasibility heuristic to (P). Our heuristic is based on a Branch-And-Bound (B&B) procedure, where, at each explored node of the B&B tree, subproblems are solved aiming to minimize the integrality gap of the solutions obtained. In this way, our approach is able to find a sequence of feasible solutions until a stopping criterion is satisfied. We call our algorithm Integrality Gap Minimization Heuristic, or, in short, IGMH.

2. Subproblems of interest

IGMH is based on the fact that the integrality gap of an ordinary solution (x, y) can be measured by the expression:

$$\text{gap}(y) = \sum_{i=1}^{n_y} y_i(1 - y_i), \tag{2}$$

where the right-hand side of the above expression measures "how far" y is from satisfying the integrality constraint. We also observe that the function gap$(\cdot)$ is quadratic, concave and separable, with global minimum at value 0, which is reached when all components of y have binary values. The function gap$(\cdot)$ was also considered in [3, 4] for the development of (exact) algorithms to solve particular cases of binary problems. In these previous works, however, this function is penalized in the objective function, and the authors propose solving the new problem iteratively, with no partitioning of the space.

IGMH adopts a B&B procedure to divide the space over the binary variables y. Let Y^k be the partition related to the k-th node in the B&B tree, z^u be an upper bound to (P), possibly initialized as $+\infty$ and $\epsilon_c > 0$ (e.g. 1e-3) be a convergence tolerance parameter. We define the integrality gap minimization problem in the partition Y^k as:

$$
\begin{aligned}
(\hat{P}^G(Y^k, z^u, \epsilon_c)) \qquad \min_{x,\,y} \quad & \sum_{i=1}^{n_y} y_i(1-y_i) && (3)\\
\text{subject to:} \quad & g(x,y) \;\leq\; 0 && (4)\\
& f(x,y) \;\leq\; z^u - \epsilon_c && (5)\\
& 0 \leq y \leq 1 && (6)\\
& x \in X,\; y \in Y^k\,. && (7)
\end{aligned}
$$

Note that problem $(\hat{P}^G(Y^k, z^u, \epsilon_c))$ does not have the integrality constraint for y, being, therefore, a (continuous) NonLinear Programming (NLP) problem. It minimizes the gap function gap$(\cdot)$ in an attempt to find a solution that satisfies the integrality constraint. The level cut constraint (5) for the original objective function aims to guarantee that the obtained solution has objective value strictly inferior than the current upper bound z^u. Observe that, while z^u is defined as $+\infty$, this constraint is always satisfied, and thus it may be omitted from the problem formulation. It is important to mention that, although problem $(\hat{P}^G(Y^k, z^u, \epsilon_c))$ has convex constraints and a nonconvex (concave) objective function, IGMH can use a (local) NLP algorithm, which can lead to obtaining a local optimum of this problem.

Assuming that $(\hat{P}^G(Y^k, z^u, \epsilon_c))$ is feasible, let $(\bar{x}^k, \bar{y}^k)$ be an optimal solution (possibly a local one) obtained by solving this problem. If $\bar{y}^k$ is integer, a local search can be performed around $(\bar{x}^k, \bar{y}^k)$ by solving the problem:

$$
\begin{aligned}
(P(\bar{y}^k)) \quad \min_{x} \quad & f(x, \bar{y}^k) \\
\text{subject to:} \quad & g(x, \bar{y}^k) \;\leq\; 0, \\
& x \in X.
\end{aligned}
\qquad (8)
$$

Note that $(P(\bar{y}^k))$ originates from (P) by fixing the integer variable at value $\bar{y}^k$. In this context, let $\tilde{x}^k$ be an optimal solution obtained for $(P(\bar{y}^k))$. We point out that $(\tilde{x}^k, \bar{y}^k)$ is the best solution for (P) having $\bar{y}^k$ as the value for the integer variable y, and it can be used to update the current upper bound by setting $z^u := f(\tilde{x}^k, \bar{y}^k)$.

The IGMH procedure is presented as Algorithm 1. Observe that the procedure uses a B&B scheme to divide the solution space over the integer variable y. At each node k of the B&B tree, problem $(\hat{P}^G(Y^k, z^u, \epsilon_c))$ seeks an integer solution that improves the current upper bound z^u. It is worth to say that, since the algorithm uses a local NLP solver, a local optimal solution can be obtained in this stage. If the obtained solution $(\bar{x}^k, \bar{y}^k)$ is integer, a local search will be conducted around $\bar{y}^k$ by solving $(P(\bar{y}^k))$. The solution $(\tilde{x}^k, \bar{y}^k)$ is used to update the current upper bound z^u, and, then, problem $(\hat{P}^G(Y^k, z^u, \epsilon_c))$ is solved again with this value updated. These steps are repeated until $(\hat{P}^G(Y^k, z^u, \epsilon_c))$ is infeasible or gives a non integer optimal solution. In the first case, the algorithm prunes the current node k, while, in the second case, a new branch occurs.

Input: (P): addressed BMINLP problem, ϵ_c: convergence tolerance
Output: (x^*, y^*): feasible solution of (P) (or fail)

1 $z^u = \infty$;
2 $Y^0 = Y$;
3 Let $N := \{0\}$ be the initial set of open nodes on the B&B enumeration tree ;
4 $i = 1$;
5 **while** $N \neq \emptyset$ **do**
6 Select a node k from N ;
7 **if** $(\hat{P}^G(Y^k, z^u, \epsilon_c))$ *is feasible* **then**
8 Let $(\bar{x}^k, \bar{y}^k)$ be an optimal solution (possibly local) of $(\hat{P}^G(Y^k, z^u, \epsilon_c))$;
9 **while** $\bar{y}^k$ *is integer* **do**
10 Let $\tilde{x}^k$ be an optimal solution of $(P(\bar{y}^k))$;
11 $(x^*, y^*) = (\tilde{x}^k, \bar{y}^k)$;
12 $z^u = f(\tilde{x}^k, \bar{y}^k)$;
13 **if** $(\hat{P}^G(Y^k, z^u, \epsilon_c))$ *is infeasible* **then**
14 **go to line 21** ;
15 Let $(\bar{x}^k, \bar{y}^k)$ be an optimal solution (possibly local) of $(\hat{P}^G(Y^k, z^u, \epsilon_c))$;
  `// Branching`
16 Select a variable y_j with non integer value $\bar{y}^k_j$;
17 $Y^{i+1} = Y^k \cap \{y \in \mathbb{R}^{n_y} : y_j = 0\}$;
18 $Y^{i+2} = Y^k \cap \{y \in \mathbb{R}^{n_y} : y_j = 1\}$;
19 $N = N \cup \{i+1, i+2\}$;
20 $i = i + 2$;
21 $N = N \backslash \{k\}$;

Algorithm 1: IGMH Algorithm.

We point out that for the B&B scheme in IGMH, there is no routine to obtain lower bounds for each partition. For this reason, pruning by bounds cannot be performed. Pruning by optimality in partitions is also impossible, since obtaining an integer solution in a partition does not guarantee that there are not better integer solutions in the same partition. In this way, IGMH can only discard partitions through pruning by infeasibility. However, observe that the objective level cut constraint (5) generates infeasibility in some partitions where the upper bound cannot be improved, that is, partitions that would be pruned by bound in a traditional B&B for integer programming are pruned by infeasibility in our proposed approach. Similarly, when the best integer solution of a partition is obtained, this constraint will cause infeasibility in the current partition (or in its subpartitions). So, pruning by optimality that would occur in a traditional B&B also occurs by infeasibility here. Pruning by infeasibility that occurs in a traditional B&B occurs in the same way in IGMH.

In the way it is presented in Algorithm 1, it is important to say that IGMH converges to the optimal solution of the original problem (P), since it is based on a B&B enumeration and no partition of space is improperly discarded. However, finding the optimal solution of the original problem with guarantee of optimality can be a very slow process. Since solving problems $(\hat{P}^G(Y^k, z^u, \epsilon_c))$ tends to give a feasible solution to (P) very fast, and the focus of this work is the development of a feasibility heuristic, we can adopt alternative stopping criteria as a maximum running time, a maximum number of iterations, or a maximum number of feasible solutions found. Considering the heuristic version of the algorithm, partitions in which IGMH gets (local) optimal solutions to problem $(\hat{P}^G(Y^k, z^u, \epsilon_c))$, at which the objective level cut (5) is active can be discarded to accelerate the execution of the algorithm. IGMH assumes, in this way, that if this constraint is active, this is a signal that there is no integer solution able to improve upper bound in the respective partition. Note that since problem $(\hat{P}^G(Y^k, z^u, \epsilon_c))$ is nonconvex, that assumption is not necessarily true. So, if this strategy is adopted, the algorithm loses the guarantee of optimality in case it is executed without any alternative stopping criterion.

3. Computational results

A total of 152 test instances of the Open source MINLP Project[1] were used to compare the performance of IGMH with 3 other feasibility heuristics found in the literature: (a) Feasibility Pump [2]; (b) *Feasibility Pump* based on Outer Approximation (OA-FP) [1]; (c) Diving [2]. All heuristics were implemented in C++, using: `Mosek 7.0` (to solve NLP problems); Cplex 12.6 (to solve the MILP problems in OA-FP). `Mosek 7.0` was also used to solve $(P(\bar{y}^k))$ in IGMH. Since this software cannot be applied to nonconvex problems, we used `Ipopt 3.12.4` with `MA27` to solve $(\hat{P}^G(Y^k, z^u, \epsilon_c))$. The numerical tests were conducted in a computer with core i7 4790 (3.6 GHz) processor, 16 GB of RAM memory over the operational system Open Suse Linux 13.1. The maximum running time was set to 10 minutes for each test instance.

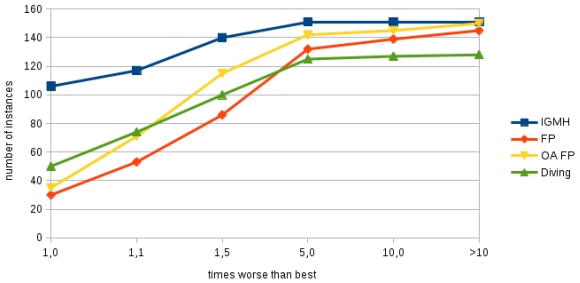


Figure 1a. Best solutions found by the heuristics for all problems.

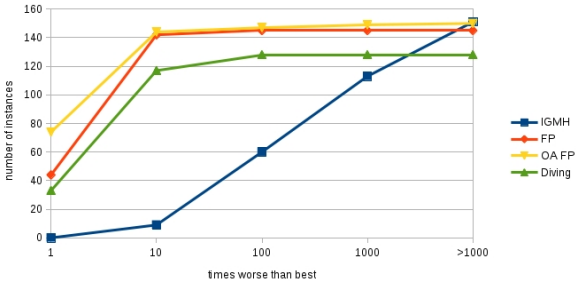


Figure 1b. Computational time spent by the heuristics for all problems.

Figures 1a and 1b show the relative comparison of the performance of the algorithms in terms of solution quality (Figure 1a) and running time (Figure 1b). Analyzing the first graphic, we observe that IGMH outperforms the other heuristics in terms of solution quality and in number of instances in which at least one feasible solution was found (151 instances). FP, OA-FP and Diving were capable of finding at least one feasible solution for 145, 150 and 128 test instances, respectively. For 101 test instances, IGMH found the best optimal solution among all the algorithms evaluated. Concerning FP, OA-FP and Diving, this number was 30, 35 and 51, respectively (note that two or more heuristics could find the best solution for the same instance in some cases). In contrast, when analyzing Figure 1b, we see that IGMH spends considerably more computational time than the other heuristics. This is due in part to the fact that other heuristics stop at the first feasible solution found for not having ways to improve this solution. IGMH, in turn, has a mechanism that allows it to continue after obtaining a feasible solution, hoping to achieve feasible solutions even better.

References

[1] P. Bonami, G. Cornuéjols, A. Lodi, and F. Margot. A feasibility pump for mixed integer nonlinear programs. *Mathematical Programming Series A*, 119:331–352, 2009.

[2] P. Bonami, and J. Gonçalves. Heuristics for convex mixed integer nonlinear programs. *Computational Optimization and Applications*, 51:729–747, 2012.

[3] W. Murray, and K. Ng. An algorithm for nonlinear optimization problems with binary variables. *Computational Optimization and Applications*, 47(2):257–288, 2010.

[4] M. Raghavachari. On Connections Between Zero-One Integer Programming and Concave Programming Under Linear Constraints. *Operations Research*, 17(4):680–684, 1969.

[1] http://egon.cheme.cmu.edu/ibm/page.htm

Proceedings of GOW'16, pp. 77 – 80.

On B&B Algorithms in Greenhouse Climate Control *

Marleen Hermelink[1], Eligius M. T. Hendrix[2] and Rene Haijema[1]

[1] *Operations Research and Logistics, Wageningen University* {marleen.hermelink, rene.haijema}@wur.nl

[2] *Computer Architecture, Universidad de Málaga* eligius@uma.es

Abstract Climate control is essential for managing the production of horticulture in greenhouses. One of the traditional ways to control the climate in the greenhouse is Model Predictive Control (MPC). This paper studies the potential of using branch and bound algorithms in order to effectively reach the best control in MPC.

Keywords: Control, Branch and bound, Greenhouse, Model predictive control

1. Introduction

In the past two decades, horticultural production has undergone a technological revolution. Producers around the world have become increasingly competitive with help of the new technologies available, and so also have producers in the Mediterranean area. Where a few years ago 100 tons of tomato per hectare was considered impressive, now a harvest of 300 tons per hectare is quite standard [2]. This exorbitant productivity has been achieved mainly through the implementation of greenhouses. These are considered ideal for growing crops, as they provide an enclosed environment which allows for controlled climate and fertigation [8].

The climate affects not only the yield of the crop, but also the quality of the products [8]. To maximize the economic benefit of the horticultural farm, a balance must be found between improving production and the costs of obtaining the right climatic conditions [8]. Most of the crops grown in greenhouses are adapted to temperatures between 17-27°C with a lower and upper limit of 10°C and 35°C [2]. They require a humidity within a range of 60-80% [3]. Temperatures outside this range lead to sub-optimal crop production and even to permanent crop damage [1]. Too high humidity levels can lead to the development of fungi on the crop, while a humidity that is too low can cause water stress [2, 8], both of which lead to a decrease in production.

Inside the greenhouse, the climatic variables which can be controlled are temperature, humidity, Photosynthetically Active Radiation (PAR) and CO_2 concentration. Temperature is the variable that influences crop growth most directly, and is thus traditionally the main focus of climate control inside the greenhouse [2, 8]. On the other hand, humidity has an indirect effect on crop growth through its influence on crop transpiration, and should thus also be taken into consideration. However, humidity and temperature are negatively correlated [8]. To address this, the general solution is to keep temperature as the main control variable, but to adjust the desired temperature depending on the relative humidity. Hence, from now on, it is assumed that climate control refers to temperature control, and that through temperature also humidity is controlled as sketched in Figure 1.

*This work has been funded by grant TIN2015-66680-c2-2-R from the Spanish state, in part financed by the European Regional Development Fund (ERDF).

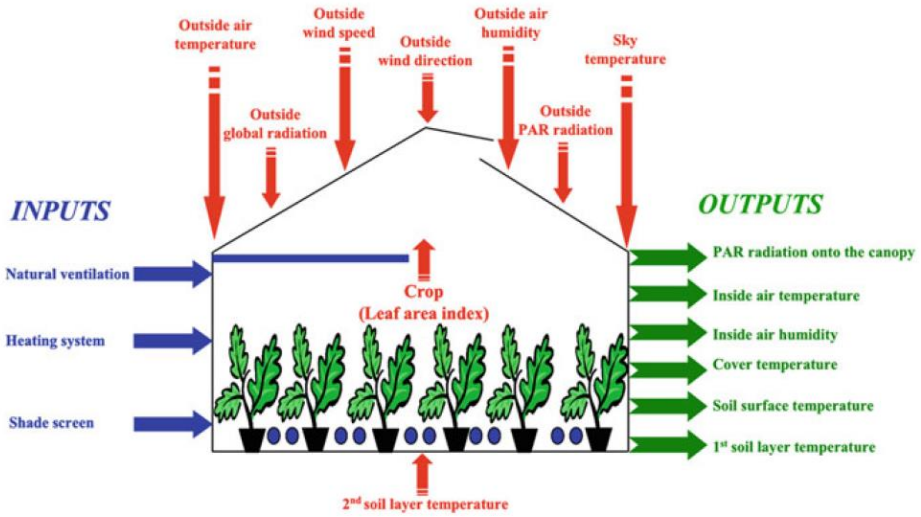


Figure 1. Inputs and outputs of climatic conditions in the greenhouse. Taken from [8].

2. Model predictive control

One of the alternatives to control the climate in a greenhouse is Model Predictive Control (MPC). It takes the future development of the states into account. To include the future in its

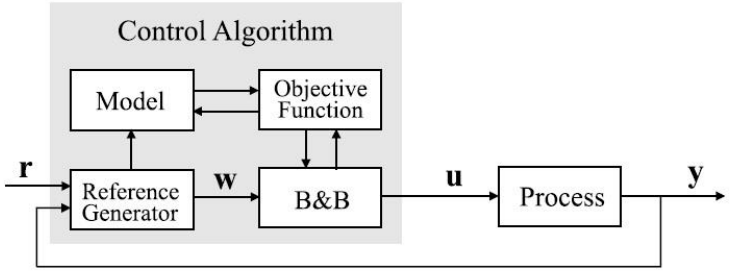


Figure 2. Model Predictive Control scheme. Taken from [6].

control strategy, MPC has a certain prediction horizon. This is the number of discrete time intervals ahead that MPC takes into account to choose a control action [9]. The dynamics of the MPC algorithm are shown in Figure 2. At a certain time instant, the MPC first samples the state of the system. It then uses a (nonlinear) model of the process to predict what would happen to the system for different combinations of control actions over the prediction horizon. An optimization technique is then used to compute the best control action sequence over the prediction horizon, based on the minimization of an objective function [6]. From the chosen control action sequence, only the first control action is actually implemented. After the implementation, MPC moves to the next time instant, the prediction horizon also shifts one time unit, and the process starts again. For this reason, MPC is also called the receding (or rolling) horizon principle [9].

Mathematically, MPC can be described as follows. The future system outputs for a certain prediction horizon PH are predicted at time instant k by using a model of the process [6]. The predicted output values $\hat{y}(k+i), i = 1, \ldots, PH$ depend on the states of the process at the current time k and on the future control signals $u(k+i), i = 1, \ldots, PH-1$. The control signals change only up to the control horizon CH, and remain constant from CH to PH. Therefore, the last control signal within the control horizon ($u(k + CH - 1)$) is applied to all intervals

after CH. So $u(k+i) = u(k+CH-1)$, for $i = CH, \ldots, PH-1$ [6]. The optimal sequence of control actions $U(k)$ must be determined over PH,

$$U(k) = [u(k), u(k+1), ..., u(k+PH-1)], \tag{1}$$

$$U(k) \in U_{PH}$$

where U_{PH} is the set of all sequences of size PH formed as combinations of control alternatives [4]. The best sequence of future control signals is found by optimizing an objective function.

In most cases the objective function is a cost function $J(k)$ describing the control goals [6]. The performance of MPC is highly dependent on the process model used [6]. If a linear time-invariant model is used, a solution can be obtained analytically. If the optimization problem is quadratic and the non-linear optimization problem is convex, the problem can be solved using fast gradient-descent methods, guaranteeing a global solution [6]. However, in most cases both non-linear models and constraints are used, resulting in a non-convex problem [6]. In that case, the most relevant solving techniques are *Sequential Quadratic Programming* [5] and the *simplex method* [7]. However, as these methods rely on iterative optimization, in the presence of non-linear constraints they hamper the application of MPC to fast systems due to their high computational costs. This makes them unsuitable for systems with short sampling times, as is the case in a greenhouse. Additionally, the convergence can lead to local minima, causing poor performance of the MPC algorithm [6].

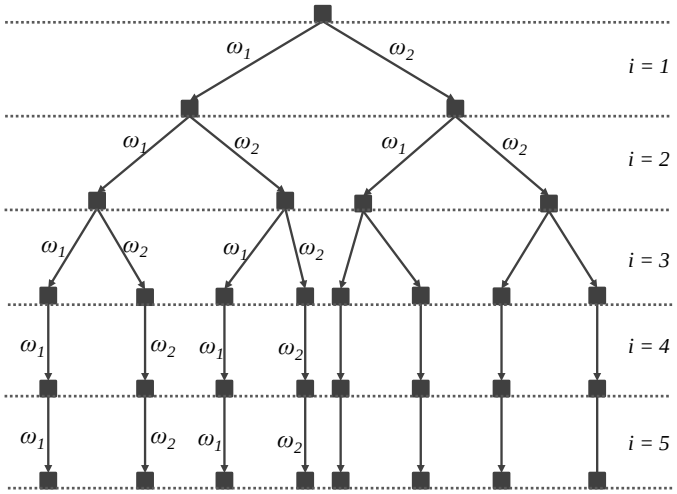


Figure 3. Example of branch and bound applied to MPC, $PH = 5$ and $CH = 3$.

3. Branch and bound in MPC

The B&B optimization of a MPC problem implicitly builds a search tree. At interval k, each time interval i, with $i = 1, \ldots, PH$, represents a level in the decision tree ($i = 0$ at the initial node) [6]. At each level i within the control horizon CH, $i = 1, \ldots, CH$, the algorithm must

decide the actuator setting $u(k+i-1)$. The system has S number of control alternatives for $u(k+i-1)$. Therefore, each node will have S number of branches [6]. For the illustration, consider a greenhouse with a heater as single actuator u, which can be turned off or on, so $u \in \{0,1\}$. Since the problem has only two control alternatives, at each level of i of the tree each node will branch in two. So $u(k+i-1)$ can be either 0 or 1. The j^{th} branch, with $j = 1, \ldots, S$, is represented by ω_j [6]. Figure 3 gives a graphical representation of a tree with two alternative control actions. Figure 3 also shows that no branching takes place beyond the control horizon ($i > CH$). Hence, control action $u(k+CH-1)$ is applied successively until the PH horizon. In other words, when moment CH is reached, the control action at the last interval within the CH is simply applied [6].

Our contribution will show how Global Optimization branch and bound can be applied to a MPC framework.

4. Conclusions

MPC is a concept applied to greenhouse climate control. Optimization of the underlying process may expose a multimodal behavior. This contribution investigates the potential of using the branch and bound framework in this context.

References

[1] Jesper Mazanti Aaslyng, Jens B Lund, Niels Ehler, and Eva Rosenqvist. Intelligrow: a greenhouse component-based climate control system. *Environmental Modelling & Software*, 18(7):657–666, 2003.

[2] Wilfried Baudoin, Remi Nono-Womdim, NeBambi Lutaladio, Alison Hodder, Nicolás Castilla, Cherubino Leonardi, Stefania De Pascale, and Muien Qaryouti. *Good agricultural practices for greenhouse vegetable crops: principles for mediterranean climate areas*. Number 217. Food and Agriculture Organization of the United Nations, 2013.

[3] N Castilla and F Nuez. Manejo del cultivo intensivo con suelo. *El cultivo de tomate. Madrid: Mundi-prensa*, pages 212–217, 1995.

[4] PM Ferreira and AE Ruano. Discrete model based greenhouse environmental control using the branch & bound algorithm. In *Proceedings of the 17th IFAC World Congress, Seoul, Korea*, volume 611, 2008.

[5] Philip E. Gill, Walter Murray, and Margaret H. Wright. *Practical optimization*. Academic press, 1981.

[6] Luís F Mendonça, JM Sousa, and JMG Sá da Costa. Optimization problems in multivariable fuzzy predictive control. *International Journal of Approximate Reasoning*, 36(3):199–221, 2004.

[7] John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.

[8] Francísco Rodríguez, Manuel Berenguel, José Luís Guzmán, and Armando Ramírez-Arias. *Modeling and Control of Greenhouse Crop Growth*. Springer, 2015.

[9] JM Sousa, R Babuška, and HB Verbruggen. Fuzzy predictive control applied to an air-conditioning system. *Control Engineering Practice*, 5(10):1395–1406, 1997.

Proceedings of GOW'16, pp. 81 – 84.

On sBB Branching for Trilinear Monomials*

Emily Speakman[1] and Jon Lee[1]

[1]*Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI, USA.*
{eespeakm, jonxlee}@umich.edu

Abstract The case of having three or more quantities multiplied together occurs frequently in global-optimization models. In the context of spatial branch-and-bound, using volume as a comparison measure, we investigate the choice of branching variable and branching point and provide some analytic insights.

Keywords: Global optimization, Mixed-integer non-linear optimization, Factorable formulation, Trilinear, Trilinear hull, McCormick relaxation, Spatial branch-and-bound, Branching point.

1. Introduction

The spatial branch-and-bound (sBB) algorithm (see [1, 7, 10], for example) is designed to find a globally-optimal solution of factorable mathematical-optimization formulations (see [4]). This divide-and-conquer technique works by introducing auxiliary variables to express every function of the original formulation as a labeled directed graph (DAG). From these DAGs, relaxations are formed and refined (see [2], for example). For a given function, the DAG can be constructed in more than one way, and therefore the algorithm has a choice to make. This choice can have a strong impact on the quality of the convex relaxation obtained from the formulation, and since sBB obtains bounds from these convex relaxations, this choice can have a significant impact on the algorithm.

There has been substantial research on how to obtain quality convex relaxations (see [3] for references), and some consideration has been given to constructing DAGs in a favorable way. In particular, [11] obtained analytic results regarding the convexifications obtained from different DAGs for trilinear monomials. [11] compute both the extreme point and inequality representations of alternative relaxations and calculate their n-dimension volumes as a comparison measure. Using volume as a measure gives a way to analytically compare formulations and corresponds to a uniform distribution of the optimal solution across a relaxation.

Along with finding good convex relaxations, another important choice in the implementation of sBB is the branching variable and branching point. There has been extensive computational research into branching-point selection (e.g., see [2]). The commonly used approaches (see [9]) are to take the midpoint of the upper and lower bounds of a variable, to branch on the value of the variable at the current solution, or to take a convex combination of the two. This last method ensures that the branching point is not too close to a bound. These alternatives are intuitive and have been supported by empirical evidence. Our work aims to provide analytic results for branching-point selection.

In our work, we focus on trilinear monomials; that is, functions of the form $f = x_1x_2x_3$, where each x_i is a simple variable. This is an important class of functions for sBB, because these results apply to monomials that involve auxiliary variables. This means that whenever a formulation contains the product of three (or more) expressions (possibly complex themselves), our results apply. In addition, the case of non-zero lower bounds is particularly im-

*The authors gratefully acknowledge partial support from NSF grant CMMI-1160915 and ONR grant N00014-14-1-0315.

portant; even if an original variable has a lower bound of zero there is no guarantee that this will also be the case for an auxiliary variable. Furthermore, after branching, the lower bound of a variable will no longer be zero for at least one child.

Using the same notation as [11], for variables $x_i \in [a_i, b_i]$, $i = 1, 2, 3$, let $\mathcal{O}_i := a_i(b_j b_k) + b_i(a_j a_k)$. Then construct a labeling such that $\mathcal{O}_1 \leq \mathcal{O}_2 \leq \mathcal{O}_3$. We can assume (w.l.o.g) that

$$a_1b_2b_3 + b_1a_2a_3 \leq b_1a_2b_3 + a_1b_2a_3 \leq b_1b_2a_3 + a_1a_2b_3. \tag{Ω}$$

This condition also arises in the complete characterization of the inequality description for the convex hull of the trilinear monomial $f = x_1x_2x_3$ (see [6, 5]). The (complicated) inequality description of the convex hull is directly used by some global-optimization software (e.g., `BARON` and `ANTIGONE`). However, other software (e.g., `COUENNE` and `SCIP`) instead uses the iterative McCormick technique to obtain a (simpler) convex relaxation for the trilinear case. These alternative approaches reflect the tradeoff between using a more complicated but stronger convexification and a simpler, weaker one. Furthermore it is not obvious which method will lead to a faster algorithm for a given problem. [11] use volume to compare the alternative (equally) simple relaxations arising from double McCormick with the trilinear convex hull.

The classic result of McCormick [4] is used to convexify *bilinear* monomials. Here, we are interested in the convex hull of the points $(f, x_1, x_2) := (a_1a_2, a_1, a_2)$, (a_1b_2, a_1, b_2), (b_1a_2, b_1, a_2), (b_1b_2, b_1, b_2); a tetrahedron in $\mathbb{R}^3$. To derive the facets of the tetrahedron, we multiply out the following inequalities and substitute the variable f for all instances of x_1x_2.

$$\begin{aligned} (x_1 - a_1)(x_2 - a_2) \geq 0, \quad (x_1 - a_1)(b_2 - x_2) \geq 0, \\ (b_1 - x_1)(x_2 - a_2) \geq 0, \quad (b_1 - x_1)(b_2 - x_2) \geq 0. \end{aligned}$$

When we use McCormick iteratively to convexify the *trilinear* monomial $f := x_1x_2x_3$, we have three choices of double-McCormick convexifications corresponding to which pair of variables we deal with first. For example, we could first group the variables x_1 and x_2, introduce an auxiliary variable $w = x_1x_2$, and convexify, and then convexify $f = wx_3$ also using the McCormick inequalities. However, we could instead group as $x_2(x_1x_3)$ or $x_1(x_2x_3)$.

Concretely (using the same notation as [11]) consider the monomial $f = x_ix_jx_k$, and assume that we first group the variables x_i and x_j. We let $w_{ij} = x_ix_j$, and so $f = w_{ij}x_k$.

Convexify $w_{ij} = x_ix_j$:

$$\begin{aligned} w_{ij} - a_jx_i - a_ix_j + a_ia_j \geq 0, \\ -w_{ij} + b_jx_i + a_ix_j - a_ib_j \geq 0, \\ -w_{ij} + a_jx_i + b_ix_j - b_ia_j \geq 0, \\ w_{ij} - b_jx_i - b_ix_j + b_ib_j \geq 0. \end{aligned}$$

Convexify $f = w_{ij}x_k$:

$$\begin{aligned} f - a_kw_{ij} - a_ia_jx_k + a_ia_ja_k \geq 0, \\ -f + b_kw_{ij} + a_ia_jx_k - a_ia_jb_k \geq 0, \\ -f + a_kw_{ij} + b_ib_jx_k - b_ib_ja_k \geq 0, \\ f - b_kw_{ij} - b_ib_jx_k + b_ib_jb_k \geq 0. \end{aligned}$$

For each of the three double-McCormick relaxations, [11] use Fourier-Motzkin elimination to project out the auxiliary variable and obtain a system in the original variables f, x_i, x_j and x_k (i.e. in $\mathbb{R}^4$). In doing so they are able to compute and compare the volume of the system resulting from each choice along with the volume of the convex hull (also in $\mathbb{R}^4$). A key result of this paper is that the 'optimal' double-McCormick relaxation is obtained when first grouping variables x_1 and x_2. We refer to the polytope arising from this relaxation as $\mathcal{P}_3$.

From [11], we have formulae for the volume of the convex hull and the best double-McCormick relaxation, parameterized in terms of the upper and lower variable bounds. Let $c_i \in [a_i, b_i]$ be the branching point of variable x_i. By substituting $a_i = c_i$ and $b_i = c_i$ for a given variable x_i into the appropriate formula and summing the results, we obtain the total resulting volume given that we branch on variable x_i at point c_i. Using this approach we show that when using the convex hull and branching on *any* variable, the midpoint gives the smallest total volume. In this sense, the commonly-used midpoint is indeed the optimal branching

point. We compare the results from branching at the midpoint of each variable and show that branching on the first variable (labeled according to Ω) gives the smallest total volume. We then show how many steps of branching we can complete before the labeling of the variables must change and branching on a different variable becomes optimal.

Next, we consider the double-McCormick relaxation, $\mathcal{P}_3$. For this, we show that when branching on variable x_3 the optimal branching point is the midpoint. However when we consider branching on either variable x_1 or variable x_2 the optimal branching point is *not* the midpoint. From [8], any double-McCormick relaxation reduces to the convex hull when the lower bounds are all zero. However, once we branch we no longer have all zero lower bounds; hence the difference in the optimal branching point when using a double-McCormick relaxation compared with the convex hull. We show that even in these cases, the sum of the two resulting (double-McCormick relaxation) volumes from branching is a convex function in the branching point over the appropriate domain. We also show that the minimum of this function always occurs at a point greater than the midpoint. Convexity is nice because we can then find the optimal branching point via a simple bisection search.

2. Results

2.1 Trilinear hull

From [11], we have that the 4-d volume of the convex hull is given by:

$$\begin{aligned}\text{Vol}_{\mathcal{P}_H} = &(b_1 - a_1)(b_2 - a_2)(b_3 - a_3) \times \\ &\left(b_1(5b_2b_3 - a_2b_3 - b_2a_3 - 3a_2a_3) + a_1(5a_2a_3 - b_2a_3 - a_2b_3 - 3b_2b_3)\right)/24,\end{aligned}$$

and the volume of the smallest double-McCormick relaxation (referred to as $\mathcal{P}_3$) comes from first grouping variables x_1 and x_2, and is given by:

$$\text{Vol}_{\mathcal{P}_3} = \text{Vol}_{\mathcal{P}_H} + \frac{(b_1 - a_1)(b_2 - a_2)^2(b_3 - a_3)^2\left(5(a_1b_1b_2 - a_1b_1a_2) + 3(b_1^2a_2 - a_1^2b_2)\right)}{24(b_1b_2 - a_1a_2)}.$$

Theorem 1. *Let $c_i \in [a_i, b_i]$ be the branching point for x_i. With the full convex hull, the smallest total volume after branching is obtained when $c_i = \frac{a_i+b_i}{2}$, i.e., branching at the midpoint is optimal.*

Theorem 2. *With the full convex hull (and branching at the midpoint of a variable), branching on x_1 obtains the smallest total volume and branching on x_3 obtains the largest total volume.*

Proposition 3. *With the full convex hull and branching on x_1 at the midpoint, for the left interval, if sBB bounds tightening does not occur, the optimal branching variable will not change until*

$$\left\lceil \log_2\left(\frac{a_2(b_1 - a_1)}{a_1(b_2 - a_2)}\right)\right\rceil \textit{ steps.}$$

Proposition 4. *With the full convex hull and branching on x_1 at the midpoint, for the right interval, if sBB bounds tightening does not occur, the optimal branching variable will not change until*

$$\left\lceil \log_2\left(\frac{b_2(b_1 - a_1)}{b_1(b_2 - a_2)}\right)\right\rceil \textit{ steps.}$$

2.2 Best double-McCormick relaxation

Some software does not use the explicit convex hull for trilinear monomials but instead employs repeated McCormick to obtain a relaxation. Here we describe some branching-point analysis for the double-McCormick relaxation $\mathcal{P}_3$ (the relaxation with the smallest volume).

Theorem 5. *Let $c_3 \in [a_3, b_3]$ be the branching point for x_3. Using the relaxation $\mathcal{P}_3$, the smallest total volume after branching is obtained when $c_3 = \frac{a_3+b_3}{2}$, i.e., branching at the midpoint is optimal.*

Next we consider branching on x_1 and x_2. Even for the special case of $a_i = 0$ and $b_i = 1$ for $i = 1, 2$, the midpoint for say x_1 is *not* the optimal branching point when using the relaxation $\mathcal{P}_3$. Substituting these values into the volume formulae, we find that the minimum of the appropriate (convex) function is obtained when branching at $x_1 = \frac{\sqrt{3}}{3} \approx 0.577$.

Theorem 6. *For $i = 1, 2$ and using the relaxation $\mathcal{P}_3$, the total volume of the relaxations after branching on x_i:*

$$Vol_{\mathcal{P}_3}\Big|_{a_i=c_i} + Vol_{\mathcal{P}_3}\Big|_{b_i=c_i}$$

is a convex function in the branching point c_i, over the domain $c_i \in [a_i, b_i]$.

Proposition 7. *For $i = 1, 2$ and using the relaxation $\mathcal{P}_3$, the minimum of the convex function*

$$Vol_{\mathcal{P}_3}\Big|_{a_i=c_i} + Vol_{\mathcal{P}_3}\Big|_{b_i=c_i}$$

over the domain $c_i \in [a_i, b_i]$ occurs at some value of $c_i > \frac{a_i+b_i}{2}$.

3. Conclusions

We have presented some analytic results on branching variable and branching-point selection in the context of sBB applied to models having functions involving the multiplication of three or more terms. Of course variables often appear in multiple functions. Therefore when deciding on a branching variable or a branching point we may obtain conflicting guidance. But this is an issue with any branching rule, including those tested empirically, and it is always a challenge to find good ways to combine local information to make algorithmic decisions.

References

[1] C.S. Adjiman, S. Dallwig, C.A. Floudas, and A. Neumaier. A global optimization method, αBB, for general twice-differentiable constrained NLPs: I. Theoretical advances. *Computers & Chemical Engineering*, 22(9):1137–1158, 1998.

[2] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods & Software*, 24(4-5):597–634, 2009.

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

[4] G.P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I. Convex underestimating problems. *Mathematical Programming*, 10:147–175, 1976.

[5] C.A. Meyer and C.A. Floudas. Trilinear monomials with mixed sign domains: Facets of the convex and concave envelopes. *Journal of Global Optimization*, 29:125–155, 2004.

[6] C.A. Meyer and C.A. Floudas. Trilinear monomials with positive or negative domains: Facets of the convex and concave envelopes. *Frontiers in Global Optimization*, pages 327–352, 2004.

[7] H.S. Ryoo and N.V. Sahinidis. A branch-and-reduce approach to global optimization. *Journal of Global Optimization*, 8(2):107–138, 1996.

[8] H.S. Ryoo and N.V. Sahinidis. Analysis of bounds for multilinear functions. *Journal of Global Optimization*, 19(4):403–424, 2001.

[9] N. Sahinidis. *BARON 15.6.5: Global Optimization of Mixed-Integer Nonlinear Programs,* User's Manual, 2015.

[10] 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.

[11] E. Speakman and J. Lee. Quantifying double McCormick. arXiv:1508.02966v2, `http://arxiv.org/abs/1508.02966`, 2015.

Proceedings of GOW'16, pp. 85 – 88.

Operational Zones for Global Optimization Algorithms*

Yaroslav D. Sergeyev[1,2], Dmitri E. Kvasov[1,2] and Marat S. Mukhametzhanov[1,2]

[1] *University of Calabria, Via P. Bucci, 42C – 87036, Rende (CS), Italy*

[2] *Lobachevsky State University, Nizhni Novgorod, Russia*

{yaro, kvadim, marat}@dimes.unical.it

Abstract The problem of comparing deterministic and stochastic global optimization methods is considered in the contribution. For this purpose, a new comparison methodology called "operational zones" is presented and described. Lipschitz deterministic and nature-inspired metaheuristic methods are then compared numerically by using the introduced operational zones.

Keywords: Operational characteristics, Operational zones, Lipschitz global optimization, Metaheuristics

1. Introduction

Let us consider the following box-constrained global optimization problem:

$$f^* = f(x^*) = \min f(x), \quad x \in D = [a, b] \subset \mathbb{R}^N, \tag{1}$$

where the objective function $f(x)$ is supposed to be "black-box", hard to evaluate, multiextremal, and non-differentiable. Let us also suppose that it satisfies the Lipschitz condition with some norm $\|\cdot\|$ over a hyperinterval D:

$$|f(x_1) - f(x_2)| \leq L\|x_1 - x_2\|, \quad x_1, x_2 \in D, \tag{2}$$

with L being the Lipschitz constant (possibly unknown), $0 < L < \infty$.

There exists a huge number of methods for solving the stated problem (see, e. g., [2, 14, 17, 20, 21, 23]). Among them we find deterministic Lipschitz-based algorithms and stochastic metaheuristics. These two types of methods have different advantages and disadvantages. Most of the metaheuristic population-based algorithms are relatively simple and easy to implement, have attractive nature-inspired interpretations, and are often used by practitioners to solve practical decision-making problems (see such problems, e. g., in [4, 11]). However, they do not always manifest strong theoretical convergence and often do not guarantee the global optimality of the found solutions. On the other hand, the Lipschitz deterministic methods are more complicated, but have strong theoretical convergence properties and can provide a guaranteed solution to problem (1)–(2).

Therefore, a comprehensive numerical comparison of the global optimization methods belonging to these two classes is important from the practical point of view. For this purpose, a new methodology called "operational zones" (see, e. g., [18]) is presented in this contribution to compare several deterministic and stochastic methods.

*This work was supported by the project No. 15-11-30022 "Global optimization, supercomputing computations, and applications" of the Russian Science Foundation, 2015–2017.

2. Operational zones

The approach of operational zones is based on the concept of "operational characteristics" proposed in 1978 by Grishagin in [5] (see also [21] for their description) for analyzing performance of deterministic methods in terms of costly function evaluations (trials). Operational characteristics can be considered as predecessors of the subsequently re-discovered "performance profiles" from [1] and "data profiles" from [13]. The operational characteristic of a method on a class of test problems is a non-decreasing function that indicates the number of problems solved (in some predefined sense) by this method after each function evaluation within a prescribed trials budget. It is convenient to represent the operational characteristic of a method in a graph where each pair (for increasing values of trials) corresponds to a point on the plane. Among different methods (or particular instances of a method) the better method is that with the highest operational characteristic.

For example, in Figure 1 the operational characteristic of a deterministic method on a class of 100 test functions is shown by red dashed line. It can be seen from this figure that, e. g., after 270 trials the method under consideration has solved 79 test problems.

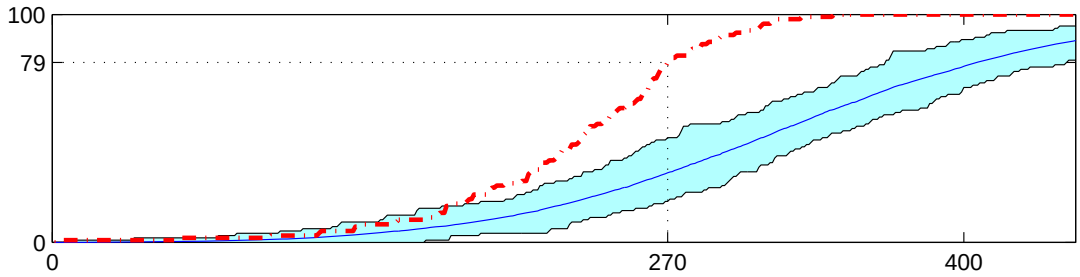


Figure 1. The operational characteristic of a deterministic method is indicated by red dashed line; the operational zone of a stochastic method is represented by cyan color, with the average operational characteristic given by continuous blue line

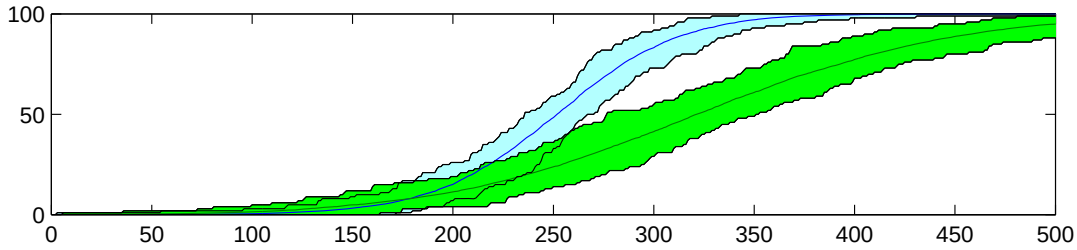
Although the operational characteristics are very representative for comparing deterministic algorithms, they cannot be used for a comparison of stochastic methods executing multiple launches with different randomly generated parameters. The operational zones are designed to fill up this gap in the following way.

Let us consider every instance of a stochastic method as a particular method. It corresponds to a particular launch of the method among the total number M of launches, characterized by a concrete set of randomly generated parameters of the method. A particular operational characteristic can be associated to this method's instance. So, for each of M launches of the algorithm the respective M operational characteristics can be constructed and the upper and lower bounds of all the operational characteristics can be therefore determined. These bounds define the whole operational zone of the method. For example, in Figure 1 the operational zone of a metaheuristic population-based algorithm is reported by cyan color; the blue line inside this zone corresponds to the average operational characteristic among the whole bundle of the operational characteristics of the method forming the method's operational zone.

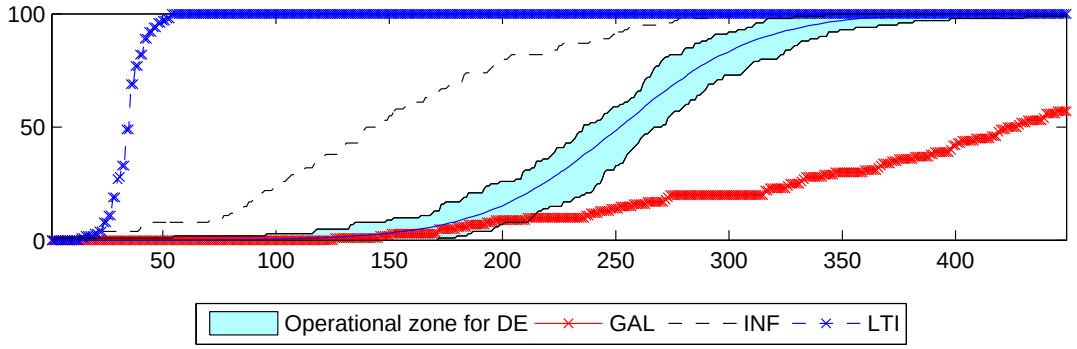
In this way, operational characteristics of deterministic methods can be compared with operational zones of stochastic algorithms, thus providing us with an intuitive visual tool for studying global optimization methods of different nature. The concept of operational zones will be used in this contribution to compare Lipschitz deterministic methods with stochastic population-based metaheuristics.

3. Comparison of Lipschitz-based and metaheuristic methods

The following Lipschitz deterministic methods and nature-inspired metaheuristics have been considered in our numerical experiments that will be presented:



(a) Operational zones for the methods DE (cyan colored) and PSO (green colored)



(b) Operational characteristics for Lipschitz methods GAL, INF, and LTI; and the operational zone for the method DE

Figure 2. Operational zones for comparing performance of deterministic and stochastic methods

GAL: **G**eometric method with an **A** priori given **L**ipschitz constant L (see, e. g., [10]).

INF: **INF**ormation-statistical method with a global estimate of L (see, e. g., [21]).

LTI: Geometric method with a local tuning and local improvement (see, e. g., [12, 19]).

The Lipschitz constants for the method GAL were estimated by the values obtained over 10^{-7}-grid. The methods INF and LTI use the reliability parameter r that was set equal to 2 for the INF method and equal to 1.1 for the LTI method.

DE: **D**ifferential **E**volution algorithm described, e. g., in [16], and implemented in `http://www1.icsi.berkeley.edu/~storn/DenewC.zip`.

PSO: **P**article **S**warm **O**ptimization algorithm proposed in [7] and implemented in its standard version.

Moreover, the following widely used metaheuristic population-based algorithms can be also considered for a further numerical comparison:

GA: **G**enetic **A**lgorithm in its standard version (see, e. g., [8, 18]) as implemented in `http://www.egr.msu.edu/~kdeb/codes/rga/`.

ABC: **A**rtificial **B**ee **C**olony algorithm proposed in [6] and implemented in `http://sci2s.ugr.es/EAMHCO#Software`.

FA: **F**irefly **A**lgorithm discussed in [22].

A detailed information on the parameters of these metaheuristics can be found, e. g., in [9, 8].

Several one-dimensional (e. g., from [19, 15]) and multidimensional (e. g., from [4, 3]) test classes were used in our numerical experiments with the considered global optimization methods. For each test function the problem (1) was considered to be solved if an algorithm has generated a trial point into an ε-neighborhood of the global minimizer (known for all the test functions).

Performance of methods of deterministic and stochastic nature can be easy compared by using operational zones. For example, operational zones of the considered algorithms over one-dimensional Pintér's test class from [15] are given in Figure 2(a)-(b). In this case, the value $\varepsilon = 10^{-5}(b-a)$ was used with the trials maximum number equal to 10000; the population size for each metaheuristic algorithm was set equal to 10. The advantages of the DE method with respect to the PSO (see Figure 2(a)) and the Lipschitz methods with adaptive es-

timate of the Lipschitz constant (the INF and the LTI methods) with respect to the considered metaheuristics on the taken test class can be observed in an intuitive way.

References

[1] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91:201–213, 2002.

[2] C. A. Floudas and P. M. Pardalos, editors. *Encyclopedia of Optimization (6 Volumes)*. Springer, 2nd edition, 2009.

[3] M. Gaviano, D. E. Kvasov, D. Lera, and Ya. D. Sergeyev. Algorithm 829: Software for generation of classes of test functions with known local and global minima for global optimization. *ACM Transactions on Mathematical Software*, 29:469–480, 2003.

[4] J. W. Gillard and D. E. Kvasov. Lipschitz optimization methods for fitting a sum of damped sinusoids to a series of observations. *Statistics and its Interface*, 2016. To appear.

[5] V. A. Grishagin. Operating characteristics of some global search algorithms. *Problems of Stochastic Search*, 7:198–206, 1978. In Russian.

[6] D. Karaboga and B. Basturk. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39(3):459–471, 2007.

[7] J. Kennedy and R. Eberhart. Particle swarm optimization. In *1995 IEEE International Conference on Neural Networks: Proceedings*, pages 1942–1948, 1995.

[8] D. E. Kvasov and M. S. Mukhametzhanov. One-dimensional global search: Nature-inspired vs. Lipschitz methods. In T. E. Simos, editor, *ICNAAM 2015: 13th International Conference of Numerical Analysis and Applied Mathematics*. AIP Conference Proceedings, 2016. To appear.

[9] D. E. Kvasov, M. S. Mukhametzhanov, and Ya. D. Sergeyev. A numerical comparison of some deterministic and nature-inspired algorithms for black-box global optimization. In B. H. V. Topping and P. Iványi, editors, *Proceedings of the Twelfth International Conference on Computational Structures Technology*, page 169, Stirlingshire, United Kingdom, 2014. Civil-Comp Press. doi: 10.4203/ccp.106.169.

[10] D. E. Kvasov and Ya. D. Sergeyev. Univariate geometric Lipschitz global optimization algorithms. *Numerical Algebra, Control and Optimization*, 2(1):69–90, 2012.

[11] D. E. Kvasov and Ya. D. Sergeyev. Deterministic approaches for solving practical black-box global optimization problems. *Advances in Engineering Software*, 80:58–66, 2015.

[12] D. Lera and Ya. D. Sergeyev. Acceleration of univariate global optimization algorithms working with Lipschitz functions and Lipschitz first derivatives. *SIAM Journal on Optimization*, 23(1):508–529, 2013.

[13] J. J. Moré and S. Wild. Benchmarking derivative-free optimization algorithms. *SIAM Journal on Optimization*, 20:172–191, 2009.

[14] R. Paulavičius and J. Žilinskas. *Simplicial Global Optimization*. Springer, New York, 2014.

[15] J. D. Pintér. Global optimization: software, test problems, and applications. In P. M. Pardalos and H. E. Romeijn, editors, *Handbook of Global Optimization*, volume 2, pages 515–569. Kluwer, 2002.

[16] K. Price, R. Storn, and J. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization*. Springer, 2005.

[17] Ya. D. Sergeyev and D. E. Kvasov. *Diagonal Global Optimization Methods*. Fizmatlit, Moscow, 2008. In Russian.

[18] Ya. D. Sergeyev, D. E. Kvasov, and M. S. Mukhametzhanov. Operational zones for comparing metaheuristic and deterministic one-dimensional global optimization algorithms. *Mathematics and Computers in Simulation*, 2016. Submitted.

[19] Ya. D. Sergeyev, M. S. Mukhametzhanov, D. E. Kvasov, and D. Lera. Derivative-free local tuning and local improvement techniques embedded in the univariate global optimization. *Journal of Optimization Theory and Applications*, 2016. In Press.

[20] Ya. D. Sergeyev, R. G. Strongin, and D. Lera. *Introduction to Global Optimization Exploiting Space-Filling Curves*. Springer, New York, 2013.

[21] R. G. Strongin and Ya. D. Sergeyev. *Global Optimization with Non-Convex Constraints: Sequential and Parallel Algorithms*. Kluwer Academic Publishers, Dordrecht, 2000. (3rd ed., 2014, Springer, New York).

[22] X.-S. Yang and X. He. Firefly algorithm: Recent advances and applications. *International Journal of Swarm Intelligence*, 1(1):36–50, 2013.

[23] A. A. Zhigljavsky and A. Žilinskas. *Stochastic Global Optimization*. Springer, New York, 2008.

Proceedings of GOW'16, pp. 89 – 92.

On Difference of Convex Optimization to Visualize Statistical Data and Dissimilarities

Emilio Carrizosa[1], Vanesa Guerrero[1] and Dolores Romero Morales[2]

[1]*Dpto. Estadística e Investigación Operativa, Universidad de Sevilla, Spain*
E. Carrizosa: ecarrizosa@us.es
V. Guerrero: vguerrero@us.es

[2]*Copenhagen Business School, Denmark* drm.eco@cbs.dk

Abstract In this talk we address the problem of visualizing in a bounded region a set of individuals, which has attached a dissimilarity measure and a statistical value. This problem, which extends the standard Multidimensional Scaling Analysis, is written as a global optimization problem whose objective is the difference of two convex functions (DC). Suitable DC decompositions allow us to use the DCA algorithm in a very efficient way. Our algorithmic approach is used to visualize two real-world datasets.

Keywords: DC programming, Visualization

Summary

In the Big Data era, Data Visualization is an area of interest to specialists from a wide variety of disciplines, [11, 12, 18, 19]. The information managed must be processed and, what is even more important, understood. Data Visualization techniques arise to respond to this requirement by developing specific frameworks to depict complex data structures as easy-to-interpret graphics, [27, 34].

Mathematical Optimization has contributed significantly to the development of this area during recent years, see [10, 22, 28] and the references therein. Nowadays, complex datasets pose new challenges in order to visualize the data in such a way that patterns are captured and useful information is extracted. Special attention is paid to represent the underlying dissimilarity relationships that data may have. Classical dimensionality reduction techniques, such as Principal Component Analysis, [29], or Multidimensional Scaling (MDS), [21, 24, 36], have been customized to deal with more complex data structures, [1, 3, 13], and to make the interpretability of the results easier via, for instance, sparse models, [5, 6, 14].

Apart from adapting existing methods, specific problems may call also for new approaches. For instance, in addition to the dissimilarity measure, the data may have attached a statistical variable, to be related with the size of each object in the graphical representation of the dataset, [16]. This is the case for geographical data, to be visualized on a map in which countries are resized according to, for instance, population rates, but maintaining the neighboring relationships of countries. This type of representations, known as cartograms, [35], leads to plots in which countries are replaced by geometrical objects, frequently circles or rectangles, while the neighborhood relationships and the size of the objects are sought to be well represented. A key issue is how such problems are expressed as optimization programs, and which optimization tools are available to cope with them. For uses of optimization applied to cartograms construction and related visualization frameworks we refer the reader to [4, 7, 8, 16, 17, 20, 23, 31, 33] and references therein.

In this talk we present a new mathematical programming framework to build a visualization map, in which a set of N individuals are depicted as convex objects in a bounded region

$\Omega \subset \mathbb{R}^n$, usually $n \leq 3$. These objects must have a volume proportional to a given statistical value associated with the individuals, $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_N)$, and they should be placed accordingly to a dissimilarity measure attached to the individuals, $\boldsymbol{\delta} = (\delta_{ij})_{i,j=1,\ldots,N}$. In order to locate the objects in Ω, a reference object $\mathcal{B}$ is used, to be translated and expanded. However, since our final goal is to obtain a visualization map which allows the analysts to understand the data they are working with, a criterion which somehow controls the appearance of the plot needs to be also considered. We will deal with this paradigm by focusing on how the objects are spread out over Ω.

Leaving aside the statistical values $\boldsymbol{\omega}$, the purpose of representing dissimilarities between individuals reminds to MDS, [3, 13, 14, 21, 24, 25, 26, 36], which aims to represent the dissimilarity between individuals as empirical distances between points in an unbounded space of lower dimension. Although our visualization model may seem very close to MDS, it has the special feature of representing in the bounded region Ω not only dissimilarities as distances between objects, but also the statistical measure $\boldsymbol{\omega}$ through the volumes of the objects in Ω. Our visualization tool is able to rescale the dissimilarities between the individuals and the statistical values associated to them to fit in Ω. Observe that fitting the objects into Ω may yield representations in which the objects intersect if their sizes are not small enough, but, on the other hand, too small objects obstruct the visualization of the statistical measure. Ideally the objects should be spread out across the visualization map. This aim will be also taken into account when modeling the problem.

The methodology proposed in this talk has applications in fields others than Data Visualization, such as for instance, Location Analysis or Distance Geometry. In location problems, the facilities to be located are usually considered as points. However, a natural extension is to consider facilities as dimensional structures, see [15], and DC techniques have been specifically applied to this generalization, [2, 9]. Ours can also be seen as a problem in Distance Geometry optimization, as carefully reviewed in [26]. In Distance Geometry, a graph realization problem consists of finding a configuration of points such that their (Euclidean) distances fit a given dissimilarity matrix. Among them is the Sensor Network Location problem, [30, 32, 37, 38], in which one assumes that some individuals are anchors (their location is known) and the remaining ones are sensors, whose location is to be obtained so that their Eculidean distances fit the dissimilarities. Thus, our method can also be applied to the Sensor Network Location problem, in which sensors and anchors have a nonnegligible area.

In this talk, the construction of a visualization map with the three characteristics mentioned above is written as a global biobjective optimization problem with convex constraints. We show that the objective function of the aggregate problem can be expressed as a difference of convex (DC) function, and thus DC optimization tools can be used to solve the optimization program.

Acknowledgments

This research is funded in part by Projects MTM2015-65915-R (Spain), P11-FQM-7603 and FQM-329 (Andalucía), all with EU ERD Funds.

References

[1] H. Abdi, L. J. Williams, D. Valentin, and M. Bennani-Dosse. STATIS and DISTATIS: optimum multitable principal component analysis and three way metric multidimensional scaling. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(2):124–167, 2012.

[2] R. Blanquero, E. Carrizosa, and P. Hansen. Locating objects in the plane using global optimization techniques. *Mathematics of Operations Research*, 34(4):837–858, 2009.

[3] I. Borg and P.J.F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005.

[4] K. Buchin, B. Speckmann, and S. Verdonschot. Evolution strategies for optimizing rectangular cartograms. In N. Xiao, M.-P. Kwan, M.F. Goodchild, and S. Shekhar, editors, *Geographic Information Science*, volume 7478 of *Lecture Notes in Computer Science*, pages 29–42. Springer, 2012.

[5] E. Carrizosa and V. Guerrero. Biobjective sparse principal component analysis. *Journal of Multivariate Analysis*, 132:151–159, 2014.

[6] E. Carrizosa and V. Guerrero. rs-Sparse principal component analysis: A mixed integer nonlinear programming approach with VNS. *Computers & Operations Research*, 52:349–354, 2014.

[7] E. Carrizosa, V. Guerrero, and D. Romero Morales. A multi-objective approach to visualize adjacencies in weighted graphs by rectangular maps. Technical report, IMUS, Sevilla, Spain, 2015.

[8] E. Carrizosa, V. Guerrero, and D. Romero Morales. Piecewise rectangular visualization maps: A large neighborhood seach approach. Technical report, IMUS, Sevilla, Spain, 2015.

[9] E. Carrizosa, M. Muñoz-Márquez, and J. Puerto. Location and shape of a rectangular facility in $\mathbb{R}^n$. Convexity properties. *Mathematical Programming*, 83(1-3):277–290, 1998.

[10] E. Carrizosa and D. Romero Morales. Supervised classification and mathematical optimization. *Computers & Operations Research*, 40(1):150–165, 2013.

[11] C. P. Chen and C.-Y. Zhang. Data-intensive applications, challenges, techniques and technologies: A survey on big data. *Information Sciences*, 275:314–347, 2014.

[12] J. Choo and H. Park. Customizing computational methods for visual analytics with big data. *IEEE Computer Graphics and Applications*, 33(4):22–28, 2013.

[13] T. F. Cox and M. A. A. Cox. *Multidimensional scaling*. CRC Press, 2000.

[14] V. De Silva and J. B. Tenenbaum. Sparse multidimensional scaling using landmark points. Technical report, Stanford University, 2004.

[15] J.M. Díaz-Báñez, J. A. Mesa, and A. Schöbel. Continuous location of dimensional structures. *European Journal of Operational Research*, 152(1):22–44, 2004.

[16] M. Dörk, S. Carpendale, and C. Williamson. Visualizing explicit and implicit relations of complex information spaces. *Information Visualization*, 11(1):5–21, 2012.

[17] D. Dorling. Area cartograms: their use and creation. In *Concepts and Techniques in Modern Geography series no. 59*. University of East Anglia: Environmental Publications, 1996.

[18] K. Fountoulakis and J. Gondzio. Performance of first- and second-order methods for big data optimization. Technical Report ERGO-15-005, 2015.

[19] K. Fountoulakis and J. Gondzio. A second-order method for strongly convex ℓ_1-regularization problems. Forthcoming in *Mathematical Programming*, 2015.

[20] E. Gomez-Nieto, F. San Roman, P. Pagliosa, W. Casaca, E. S. Helou, M. C. F. de Oliveira, and L. G. Nonato. Similarity preserving snippet-based visualization of web search results. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):457–470, 2014.

[21] J. C. Gower. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53(3-4):325–338, 1966.

[22] P. Hansen and B. Jaumard. Cluster analysis and mathematical programming. *Mathematical Programming*, 79(1-3):191–215, 1997.

[23] R. Heilmann, D. A. Keim, C. Panse, and M. Sips. Recmap: Rectangular map approximations. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 33–40. IEEE Computer Society, 2004.

[24] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[25] H. A. Le Thi and T. Pham Dinh. D.C. Programming Approach to the Multidimensional Scaling Problem. In A. Migdalas, P.M. Pardalos, and P. Värbrand, editors, *From Local to Global Optimization*, volume 53 of *Nonconvex Optimizations and Its Applications*, pages 231–276. Springer, 2001.

[26] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino. Euclidean distance geometry and applications. *SIAM Review*, 56(1):3–69, 2014.

[27] S. Liu, W. Cui, Y. Wu, and M. Liu. A survey on information visualization: recent advances and challenges. *The Visual Computer*, 30(12):1373–1393, 2014.

[28] S. Olafsson, X. Li, and S. Wu. Operations research and data mining. *European Journal of Operational Research*, 187(3):1429–1448, 2008.

[29] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.

[30] T. K. Pong and P. Tseng. (Robust) edge-based semidefinite programming relaxation of sensor network localization. *Mathematical programming*, 130(2):321–358, 2011.

[31] R. L. Rabello, G. R. Mauri, G. M. Ribeiro, and L. A. N. Lorena. A clustering search metaheuristic for the point-feature cartographic label placement problem. *European Journal of Operational Research*, 234(3):802–808, 2014.

[32] A. M.-C. So and Y. Ye. Theory of semidefinite programming for sensor network localization. *Mathematical Programming*, 109(2-3):367–384, 2007.

[33] B. Speckmann, M. van Kreveld, and S. Florisson. A linear programming approach to rectangular cartograms. In *Proceedings of the 12th International Symposium on Spatial Data Handling*, pages 527–546. Springer, 2006.

[34] J. Thomas and P.C. Wong. Visual analytics. *IEEE Computer Graphics and Applications*, 24(5):20–21, 2004.

[35] W. Tobler. Thirty five years of computer cartograms. *Annals of the Association of American Geographers*, 94(1):58–73, 2004.

[36] W.S. Torgerson. *Theory and Methods of Scaling*. Wiley, 1958.

[37] P. Tseng. Second-order cone programming relaxation of sensor network localization. *SIAM Journal on Optimization*, 18(1):156–185, 2007.

[38] Z. Wang, S. Zheng, Y. Ye, and S. Boyd. Further relaxations of the semidefinite programming approach to sensor network localization. *SIAM Journal on Optimization*, 19(2):655–673, 2008.

Proceedings of GOW'16, pp. 93 – 96.

Extensions on Ellipsoid Bounds for Quadratic Programs

Marcia Fampa[1] and Francisco Pinillos Nieto[1]

[1]*Federal University of Rio de Janeiro, Rio de Janeiro, Brazil,* {fampa,paquito}@cos.ufrj.br

Abstract Ellipsoid bounds for strictly convex quadratic integer programs have been proposed in [1, 2]. The idea is to underestimate the objective function of the problem by another convex quadratic function for which an integer minimimizer is easily computed. We initially propose in this paper a different way of constructing the quadratic underestimator for the same problem and then extend the idea to more general problems where the objective function is convex (not necessarily strictly convex), and box constraints are introduced. The quality of the proposed bounds is evaluated experimentally and compared to related existing methodologies.

Keywords: Quadratic Integer Programming, Ellipsoid bound

1. Introduction

The quadratic integer programming problem has been focus of intense research in the last years. Obtaining tight dual bounds to its optimal objective value is essential for the successful application of branch-and-bound (B&B) algorithms. When the problem is convex, a common approach to obtain dual bounds is simply to relax the integrality constraints and solve the continuous relaxation. An alternative approach has been proposed in [1] for the case where the objective function is strictly convex. A convex quadratic underestimator for the objective function is used on the relaxation proposed, which has the same continuous minimizer as the objective function and has an integer minimimizer that can be easily obtained. More recently, in [2], the authors have generalized the idea introduced in [1], proposing classes of underestimators with the *strong rounding property* introduced in [3], property given to functions for which an integer minimizer is obtained by rounding the continuous minimizer. The authors propose heuristics to search for underestimators in these classes leading to the tightest possible *ellipsoid bounds*, name introduced in [2] and motivated by the geometric interpretation of the ellipsoidal sublevel sets of the convex quadratic underestimators. The results in [2] show that ellipsoid bounds are never worse than the continuous bounds and can be quickly computed, however they apply only for the minimization of strictly convex quadratic functions.

We initially propose in this paper a new method to generate underestimators for strictly convex quadratic objective functions. Next, we extend the ellipsoid bounds to some convex quadratic objective functions (not necessarily strictly convex). Finally, we show how the idea can be applied to the case where box constraints are introduced. Numerical results compare the approaches proposed in this paper to related existing approaches from the literature.

2. Ellipsoid bounds for strictly convex quadratics

In this section, we address strictly convex quadratic integer programs of the form

$$\min\{q(x) := x^T Q x + c^T x \ : \ x \in \mathbb{Z}^n\} \text{ (SCIQP)}$$

where Q is a positive definite $n \times n$ matrix, and $c \in \mathbb{R}^n$. We denote the eigenvalues of Q by $\lambda_{\max} := \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n := \lambda_{\min}$, and the corresponding eigenvectors by $u_1, u_2, \ldots, u_n$. The eigenvalue decomposition of Q is given by $Q = U \Lambda U^T$.

Ellipsoid bounds for (SCIQP) [1, 2] are given by the solution of an integer relaxation of the problem, where we minimize an underestimator function q' of q, over $\mathbb{Z}^n$. Denoting by $\bar{x}$ the unique continuous minimizer of q, given by $-(1/2)Q^{-1}c$, it is possible to verify that

$$q(x) = q(\bar{x}) + (x - \bar{x})^T Q(x - \bar{x}),$$

for all $\mathbb{R}^n$. Defining, then

$$q'(x) := q(\bar{x}) + (x - \bar{x})^T Q'(x - \bar{x}), \tag{1}$$

where $Q - Q' \succeq 0$ and $Q' \succeq 0$, it is straightforward to see that $q(x) \geq q'(x)$, for all $x \in \mathbb{R}^n$.

Nevertheless, to apply the ellipsoid bounds efficiently, it is essential that the integer minimimum of q' can be computed easily.

In [1], it is shown that if Q' is diagonal, an integer minimizer of q' is simply obtained by rounding $\bar{x}$. More specifically, an integer minimizer is given by $\lfloor \bar{x} \rceil = (\lfloor \bar{x}_1 \rceil, \ldots, \lfloor \bar{x}_n \rceil)^T$, where for any $a \in \mathbb{R}$, $\lfloor a \rceil$ denotes the closest integer to a. Furthermore, the authors present the following semidefinite programming problem, for which the optimal solution is a diagonal matrix $Q' := \text{Diag}(t)$ that leads to the best possible ellipsoid bound to (SCIQP), among all positive semidefinite diagonal matrices.

$$\max\{\sum_{i=1}^{n} t_i(\lfloor \bar{x}_i \rceil - \bar{x}_i)^2 \ : \ Q - \text{Diag}(t) \succeq 0 \, , t \geq 0 \ \} \text{ (SDP)}$$

Generalizing the results in [1], some classes of matrices Q' are presented in [2] for which the quadratic function q' has the *strong rounding property*. Guided by the geometric characterization of the ellipsoidal sublevel sets of q', the authors initially define *quasiround* sets, as follows.

Definition 1. *Let $B(x, \rho)$ be the Euclidean ball of radius $\rho \in \mathbb{R}_+$ and center $x \in \mathbb{R}^n$. Given $\alpha \in \mathbb{R}_+$ and $x_0 \in M$, we call a set $M \subseteq \mathbb{R}^n$ α-quasiround w.r.t. x_0 if there exist radii $\rho_1, \rho_2 \in \mathbb{R}_+$ such that $B(x_0, \rho_1) \subseteq M \subseteq B(x_0, \rho_2)$ and $\rho_2 - \rho_1 \leq \alpha$.*

Next, defining $d(x, D) := \inf\{||x - y||\colon y \in D\}$, for any $x \in \mathbb{R}^n$ and $D \subset \mathbb{R}^n$, and $\alpha(x) := d(x, \mathbb{Z}^n \setminus \{\lfloor x \rceil\}) - d(x, \mathbb{Z}^n)$, the authors in [2] show the two following results.

Theorem 2. *Let $q' : \mathbb{R}^n \to \mathbb{R}$. Assume that the sublevel sets of q', given by $\mathcal{L}_{q'}(z) := \{x \in \mathbb{R}^n : q'(x) \leq z\}$, are $\alpha(\bar{x})$-quasiround w.r.t. $\bar{x}$, for any continuous minimizer $\bar{x}$ of q', and for all $z \leq q'(\lfloor \bar{x} \rceil)$. Then q' has the strong rounding property.*

Corollary 3. *Let $\lambda'_{\max} := \lambda'_1 \geq \lambda'_2 \geq \ldots \geq \lambda'_n =: \lambda'_{\min}$ be the eigenvalues of Q'. If relation* (2) *presented below is satisfied, then the quadratic function q' has the strong rounding property.*

$$\lambda'_{\max} \leq \lambda'_{\min}/(1 - \alpha(\bar{x})\sqrt{\lambda'_{\min}})^2 \, . \tag{2}$$

Computing the matrix Q' satisfying (2) that leads to the best lower bound for (SCQIP) cannot be done as efficiently as it can be done for the case where Q' is diagonal, where one solves problem (SDP). Therefore, in this case the authors in [2] propose a heuristic to construct Q'. The eigenvectors of Q' are set equal to the eigenvectors of Q and, to satisfy (2), each eigenvalue λ'_i of Q' is set as the minimum between λ_i and $\lambda_{\min}/(1 - \alpha(\bar{x})\sqrt{\lambda_{\min}})^2$. The heuristic choice for the eigenvectors and eigenvalues of Q' guarantee that q' underestimates q and has the strong rounding property. Results in [2] show that this heuristic can lead to bounds computed much faster than the bounds given by the solution of (SDP).

An intuition about the ellipsoid bounds is given by the level sets of q and q', which correspond to ellipsoids centered at $\bar{x}$. The ellipsoid corresponding to q' for the level fixed as $q'(\lfloor \bar{x} \rceil)$ should contain the the ellipsoid corresponding to q for the same level, should contain $\lfloor \bar{x} \rceil$ on its border, and no integer point in its interior. We note that to have strong bounds, it is good to have both ellipsoids as similar as possible. Therefore, setting the eigenvectors of Q' equal to

the eigenvectors of Q looks like an interesting idea. However, the upper bounds imposed by (2) on the eigenvalues of Q' may change too much the shape of the corresponding ellipsoid, when compared to Q, and lead to weak bounds. Alternatively, following ideas introduced in [1], we propose the use of an easily computed rank one matrix $Q' = \delta vv^T$, where $\delta \in \mathbb{R}_+$ and $v \in \mathbb{Z}^n \setminus \{0\}$. For our matrix Q', the goal is to have the angle between the vector v and the eigenvector u_n of Q, corresponding to $\lambda_{\min}$, as close as possible to $90°$. Properly selecting integer components for v, we show that the integer minimum of q' can still be easily computed and discuss the benefits of this choice. For a given vector v, the parameter δ is selected to maximize the bound, while ensuring that q' underestimates q. Algorithm 1 shows the major steps of our procedure.

Algorithm 1 : Computation of $Q'(Q = U\Lambda U^T)$)

$(\alpha, \beta) := \arg\min_{\{(i,j):i,j\in\{1,\ldots,n\},i\neq j\}} \left\{ \left| \left\lfloor \frac{u_n(i)}{u_n(j)} \right\rceil - \frac{u_n(i)}{u_n(j)} \right| \right\}$;

$v(\alpha) := -1$;

$v(\beta) := \left\lfloor \frac{u_n(\alpha)}{u_n(\beta)} \right\rceil$;

$v(k) := 0$, for all $k \in \{1, \ldots, n\}$, $k \neq \alpha$, $k \neq \beta$;

Select δ^* as the largest $\delta \in \mathbb{R}_+$ such that $Q - \delta vv^T \succeq 0$;

$Q' := \delta^* vv^T$;

return (Q',δ^*, v).

Note that if $\frac{u_n(\alpha)}{u_n(\beta)} \in \mathbb{Z}$, then

$$v^T u_n = -u_n(\alpha) + \left\lfloor \frac{u_n(\alpha)}{u_n(\beta)} \right\rceil u_n(\beta) = 0,$$

i.e., in this case the vector v is orthogonal to u_n. Otherwise, we select the indexes α and β aiming at having the angle between v and u_n as close as possible to $90°$.

Accordingly to Observation 4 in [1], the parameter δ^* in Algorithm 1 can be computed as

$$\delta^* := \frac{1}{||v^T(\sqrt{\Lambda}U^T)^{-1}||_2^2}.$$

We note that as $v \in \mathbb{Z}^n$, a lower bound to (SCIQP), is given by

$$lb := q(\bar{x}) + \delta^*(\lfloor v^T\bar{x} \rceil - v^T\bar{x})^2. \tag{3}$$

Finally, for our selected vector v, it is also straightforward to verify that the above lower bound is indeed the integer minimum of q'.

3. Generalizations for the ellipsoid bounds

In this section we discuss a generalization of the ellipsoid bounds for the case where the matrix Q is positive semidefinite, but not positive definite. Let us consider problem (CQIP) similar to (SCIQP), where the only difference is the fact that $\text{rank}(Q) = k < n$. The eigenvalues of Q are given by $\lambda_{\max} := \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k > \lambda_{k+1} = \lambda_{k+2} = \ldots = \lambda_n = 0$, and the corresponding eigenvectors by $u_1, u_2, \ldots, u_n$. It is important to note that although the continuous minimizer of (CQIP) is not unique, for any continuous minimizer $\bar{x}$, we still can verify that $q(x) := q(\bar{x}) + (x - \bar{x})^T Q(x - \bar{x})$.

Therefore, we again search for a rank one-matrix $Q' = \delta^* vv^T$ such that $q'(x) := q(\bar{x}) + (x - \bar{x})^T Q'(x - \bar{x})$ underestimates $q(x)$ for all $x \in \mathbb{R}^n$, and the integer minimum of q', or at least a good lower bound for it, is easily computed. As in the strictly convex case if $v \in \mathbb{Z}^n$, then a lower bound for (CQIP) is given by (3). In order to compute for a given vector v, the parameter

δ^* as the largest $\delta \in \mathbb{R}_+$ such that $Q - \delta vv^T \succeq 0$, we generalize the results presented in [1] as it follows.

Remark 4. *Let $Q = U\Lambda U^T$ be the eigenvalue decomposition of the $n \times n$ matrix Q, where rank$(Q) = k \leq n$. Let $v \in \mathbb{R}^n$ and $v = v_k + v_p$, where $v_k \in Im(Q)$ and $v_p \in ker\ (Q)$. Let δ^* be the largest $\delta \in \mathbb{R}_+$ such that $Q - \delta vv^T \succeq 0$.*

Then, if $v_p \neq 0$,

$$\delta^* = 0,$$

otherwise,

$$\delta^* := \frac{1}{||v^T\Sigma||_2^2},$$

where Σ is the pseudoinverse of $\sqrt{\Lambda}U^T$.

We discuss how to apply the result above to compute bounds for special classes of positive semidefinite matrices Q and extend Algorithm 1 to (CQIP). We finally extend Algorithm 1 also to the case where box constraints are added to the problem.

4. Numerical results and conclusions

We implemented the heuristic proposed in [2], Algorithm 1, and solved problem (SDP) for the strictly convex case. Considering randomly generated instances, we show how Algorithm 1 compares to the other methods. Our algorithm generates better bounds than the heuristic for all instances with shorter running time than (SDP) requires to be solved. Ellipsoid bounds are also always better than the continuous bounds $q(\bar{x})$. We also implemented the generalized Algorithm 1 for the convex case with and without box constraints. In both cases we are able to generate lower bounds better than the continuous minimum of the quadratic function in short running time. The experiments with Algorithm 1 show that the rotation of the eigenvectors of Q is an important feature in our method to get good ellipsoid bounds, and the extensions proposed in this work make it possible to apply the methodology to more general quadratic problems.

References

[1] C. Buchheim, A. Caprara, and A. Lodi. An effective branch-and-bound algorithm for convex quadratic integer programming. *Mathematical Programming (Series A)*, 135(1–2):369–395, 2012.

[2] C. Buchheim, R. Hübner, and A. Schöbel. Ellipsoid Bounds for Convex Quadratic Integer Programming. *SIAM Journal on Optimization*, 25(2):741–769, 2015.

[3] R. Hübner and A. Schöbel. When is rounding allowed in integer nonlinear optimization? *European Journal of Operations Research*, 237(2):404–410, 2014.

Proceedings of GOW'16, pp. 97 – 100.

Solving MINLP Problems by a Penalty Framework*

Ana Maria A.C. Rocha[1,2], M. Fernanda P. Costa[3] and Edite M.G.P. Fernandes[2]

[1]*Production and Systems Department, University of Minho, Braga, Portugal,* arocha@dps.uminho.pt

[2]*Algoritmi Research Centre, University of Minho, Braga, Portugal*

[3]*Centre of Mathematics, University of Minho, Braga, Portugal,* mfc@math.uminho.pt

Abstract A penalty framework for globally solving mixed-integer nonlinear programming problems is presented. Both integrality constraints and nonlinear constraints are handled separately by hyperbolic tangent penalty functions. The preliminary numerical experiments show that the proposed penalty approach is effective and the hyperbolic tangent penalties compete with other popular penalties.

Keywords: MINLP, Exact penalty, DIRECT

1. Introduction

A penalty approach for globally solving mixed-integer nonlinear programming (MINLP) problems is presented. A continuous relaxation of the MINLP problem is carried out by converting it to a finite sequence of bound constrained nonlinear programming (BCNLP) problems with only continuous variables. The MINLP problem is addressed in the form:

$$\begin{array}{ll} \underset{x\in X\subset R^n}{min} & f(x) \\ \text{subject to} & g_j(x)\leq 0, j=1,\ldots,p \\ & h_l(x)=0, l=1,\ldots,m \\ & x_i\in\mathbb{R} \text{ for } i\in I_c\subseteq I\equiv\{1,\ldots,n\} \\ & x_j\in\mathbb{Z} \text{ for } j\in I_d\subseteq I \end{array} \tag{1}$$

where $f, g_j, h_l : \mathbb{R}^n \rightarrow \mathbb{R}$ are continuous possibly nonlinear functions in a compact subset of $\mathbb{R}^n$, herein defined as $X = \{x : -\infty < lb_i \leq x_i \leq ub_i < \infty, i = 1,\ldots,n\}$ and $I_c \cap I_d = \emptyset$ and $I_c \cup I_d = I$. Let C be the following subset of $\mathbb{R}^n$, $C = \{x \in X : g_j(x) \leq 0, j = 1,\ldots,p,\ h_l(x) = 0, l = 1,\ldots,m\}$ (that we assume to be compact) and let $W \subseteq C$ be the nonempty feasible region of the problem (1) $W = \{x \in C \subset \mathbb{R}^n : x_j \in \mathbb{Z} \text{ for } j \in I_d \subseteq I\}$. A penalty continuous formulation of the MINLP problem is used. First, a continuous relaxation of the MINLP problem (1) is obtained by relaxing the integrality conditions from $x_j \in \mathbb{Z}$, $j \in I_d$ to $x_j \in \mathbb{R}$, $j \in I_d$, and by adding a penalty term to the objective function that aims to penalize integrality constraint violation (see [2, 5]). Second, the resulting nonlinear programming (NLP) penalty problem is formulated as a BCNLP problem with an objective penalty function that is related to the objective function of the continuous relaxation of the MINLP and the nonlinear constraints violation.

Thus, our contribution in this article is directed to the combination of two penalty terms aiming to penalize integrality violation and nonlinear inequality and equality constraints violation separately. The penalty term for the integrality constraints is based on the hyperbolic tangent function [2] and the inequality and equality constraints violation is dealt with penalties that also rely on the hyperbolic tangent function. The solution of the BCNLP penalty prob-

*This work has been supported by COMPETE: POCI-01-0145-FEDER-007043 and FCT - Fundação para a Ciência e Tecnologia, within the projects UID/CEC/00319/2013 and UID/MAT/00013/2013.

lem are then obtained using the DIRECT algorithm [4], a deterministic algorithm for finding global solutions inside hyperrectangles. We illustrate the performance of the proposed exact penalty approach on three well-known test problems.

2. Penalty functions for MINLP

A penalty approach that can be extended to solve MINLP problems is investigated. In this context, a penalty function selected from a class of penalty functions for solving general integer problems [2, 5, 6] is used. Problem (1) is equivalent to the following continuous reformulation (in the sense that they have the same global minimizers), which comes out by relaxing the integer constraints on the variables and adding a particular penalty term to the objective function, as follows:

$$\begin{array}{rl} \underset{x\in C}{min} & \phi(x;\varepsilon) \equiv f(x) + P(x;\varepsilon) \\ \text{subject to} & x_i \in \mathbb{R},\ i = 1,\ldots,n, \end{array} \tag{2}$$

where $\varepsilon \in \mathbb{R}^+$ is a penalty parameter, and

$$P(x;\varepsilon) = \frac{1}{\varepsilon} \sum_{j\in I_d} \min_{lb_j \le d_i \le ub_j \wedge d_i \in \mathbb{Z}} \tanh\left(|x_j - d_i| + \varepsilon\right) \tag{3}$$

is the penalty term based on the hyperbolic tangent function, which is differentiable and strictly increasing on the set X [2]. The resulting penalty function in the NLP problem (2) is termed exact since $\exists \bar{\varepsilon} \in \mathbb{R}^+$ such that for all $\varepsilon \in (0, \bar{\varepsilon}]$, problems (1) and (2) have the same global minimizers (see Theorem 2.1 in [5]). Assuming that the set C is compact, the proof of Theorem 2.1 in [5] is based on specific assumptions on the objective function f and on the penalty term $P(x;\varepsilon)$. The particular case in (3) satisfies those assumptions (see Property 2.5 in [2]).

Furthermore, combining this idea with a penalty-based strategy for the nonlinear inequality and equality constraints, the BCNLP problem arises in the form

$$\begin{array}{rl} \underset{x\in X}{min} & \Psi(x;\varepsilon,\mu) \equiv \phi(x;\varepsilon) + \mu \left(\displaystyle\sum_{j=1}^{p} \tanh(\max\{g_j(x), 0\}) + \sum_{l=1}^{m} \tanh(|h_l(x)|) \right) \\ \text{subject to} & x_i \in \mathbb{R},\ i = 1,\ldots,n, \end{array} \tag{4}$$

where we have extended the use of the '$\tanh(\cdot)$' to the general constraints violation and $\mu > 0$ is the penalty parameter. Ψ is a non-differentiable penalty function, although continuously differentiable at infeasible points, if f and the constraint functions are differentiable. An issue that remains to be established is the exactness property of the penalty function $\Psi(x;\varepsilon,\mu)$ in the context of using problem (4) to find an optimal solution to (2).

Algorithm 1 describes the proposed penalty framework aiming to find a global minimizer of the MINLP problem (1) by computing a global minimizer of the BCNLP problem formulated in (4), where $z^k \in X$, $z_j^k \in \mathbb{Z}, j \in I_d$ results from rounding x_j^k to the nearest integer and $z_i^k = x_i^k, i \in I_c$.

Besides forcing the integer variables to take integer values, another important issue is to reduce the overall nonlinear constraint violation, which is measured in terms of the maximum violation by $\eta(x^k) = \max_{j=1,\ldots,p;l=1,\ldots,m} \left\{\max\{g_j(x^k), 0\}, |h_l(x^k)|\right\}$. Although more complex rules may be selected to control the reduction of parameters like $\varepsilon, \eta, \delta$ and the growth of parameter μ, we use simple schemes for these preliminary experiments.

To solve the BCNLP problems formulated in (4), a deterministic algorithm that uses only function evaluations, DIRECT [4] is used. DIRECT is efficient, in the sense that a few function evaluations are required, to find just an approximation to the solution, although the number of evaluations grows faster when a high quality solution is required. The problem to

Input: (x^*, f^*) (global solution), x^0, ε^1, μ^1, η^1, δ^1
Set $k = 1$;
while $\|x^{k-1} - x^*\| > 1E-3$ *or* $\eta(x^{k-1}) > 1E-4$ *or* $f^{k-1} > f^* + 1E-3$ **do**
  Compute x^k such that $\Psi(x^k; \varepsilon^k, \mu^k) \leq \Psi(x; \varepsilon^k, \mu^k) + \delta^k$, for all $x \in X$;
  if $\|x^k - z^k\| > 1E-3$ *and* $\phi(x^k; \varepsilon^k) - \phi(z^k; \varepsilon^k) \leq \varepsilon^k \|x^k - z^k\|$ **then**
    $\varepsilon^{k+1} = 0.1\varepsilon^k$; $\eta^{k+1} = \eta^k$; $\delta^{k+1} = \delta^k$;
  else
    if $\eta(x^k) \leq \eta^k$ **then**
      $\mu^{k+1} = \mu^k$; $\eta^{k+1} = \max\{0.1\eta^k, 1E-4\}$; $\delta^{k+1} = 0.1\delta^k$;
    else
      $\mu^{k+1} = 2\mu^k$; $\eta^{k+1} = \eta^k$; $\delta^{k+1} = \delta^k$;
    if $\|x^k - x^*\| > 1E-1$ **then**
      $\varepsilon^{k+1} = 0.9\varepsilon^k$;
  Set $k = k + 1$;

Algorithm 1: Penalty-based algorithm

be addressed by DIRECT has the following form: for fixed $\varepsilon^k, \mu^k, \delta^k$, find $x^k \in X$ such that $\Psi(x^k; \varepsilon^k, \mu^k) \leq \Psi(x; \varepsilon^k, \mu^k) + \delta^k$ for all $x \in X$, assuming that the objective function $\Psi(x; \cdot)$ is Lipschitz continuous on X.

DIRECT is designed to completely explore the search space and is mainly characterized by sequentially dividing the space X into hyperrectangles and evaluating Ψ at their centers. To perform a balance between global and local search, the algorithm makes use of two important concepts: potentially optimal hyperrectangle and grouping according to size. The center c_i, the objective function value, $\Psi(c_i; \cdot)$, and the size d_i - originally given by the distance from the center to a corner - of the hyperrectangle i are used to define the groups of hyperrectangles, to select the potentially optimal hyperrectangles and divide them into smaller ones, until typically a maximum number of function evaluations is reached.

3. Numerical results

To make a preliminary evaluation of the practical behavior of the proposed penalty framework, based on the penalty presented in (4), we use three well-known MINLP problems (see [7]) which have two solutions, one global and one local:

$$(P1)\ \begin{array}{ll}\min & f(x) \equiv -x_1 - x_2 \\ \text{s.t.} & x_1 x_2 - 4 \leq 0, \\ & 0 \leq x_1 \leq 4, \\ & x_2 \in \{0, \ldots, 6\} \\ & f^* = -6.6666667\end{array}$$

$$(P2)\ \begin{array}{ll}\min & f(x) \equiv 35 x_1^{0.6} + 35 x_2^{0.6} \\ \text{s.t.} & 600x_1 - 50x_3 - x_1 x_3 + 5000 = 0 \\ & 600x_2 + 50x_3 - 15000 = 0 \\ & 0 \leq x_1 \leq 34, 0 \leq x_2 \leq 17, \\ & x_3 \in \{100, \ldots, 300\} \\ & f^* = 189.311627\end{array}$$

$$(P3)\ \begin{array}{ll}\min & f(x) \equiv 2x_1 + x_2 \\ \text{s.t.} & 1.25 - x_1^2 - x_2 \leq 0 \\ & x_1 + x_2 - 1.6 \leq 0 \\ & 0 \leq x_1 \leq 1.6, \\ & x_2 \in \{0, 1\} \\ & f^* = 2\end{array}$$

In the context of the proposed penalty algorithm, we have also tested the three most popular general constraint penalties yielding the final penalty function:

$$\Psi(x; \varepsilon, \mu) = \phi(x; \varepsilon) + \mu \left(\sum_{j=1}^{p} (\max\{g_j(x), 0\})^q + \sum_{l=1}^{m} (|h_l(x)|)^q \right) \quad \text{for } q = 1/2, 1, 2. \tag{5}$$

The penalty algorithm is coded in MATLAB programming language (Matlab Version 8.1.0.604 (R2013a)), the MATLAB code 'DIRECT.m' [3] is invoked, and the numerical experiments were carried out on a PC Intel Core 2 Duo Processor E7500 with 2.9GHz and 4Gb of memory.

Table 1 contains the results obtained by the present study with the penalty presented in (4) and with the penalty functions in (5) for comparison, where f is the computed solution, 'C.viol.' and 'I.viol.' are the general constraint and the integrality violations, respectively, Nf_{eval} is the number of function evaluations, It is the number of iterations and T is the CPU time (in seconds). For comparison, the results of a hybrid stochastic algorithm [1] and of an exact branch-and-reduce algorithm [7] are also shown. The herein listed results inside parentheses mean that the condition of the stopping rule of the algorithm related to that quantity is not satisfied. Our penalty algorithm always converges to the global solution and is able to reach good approximate solutions in a reasonable time. The practical performance of the penalty presented in (4) is comparable to the penalty in (5) with $q = 1$ and these two are superior to the other two penalties in comparison. It can be concluded that the proposed penalty approach for MINLP is effective and deserves further developments.

Table 1. Numerical results based on $\varepsilon^1 = 1$, $\mu^1 = 100$, $\eta^1 = 0.1$, $\delta^1 = 1$ and a maximum of 18 iterations.

Problem	Method	f	C.viol.	I.viol.	Nf_{eval}	It	T
$(P1)$	this study with (4)	-6.666661E+00	0.00E+00	5.65E-06	17643	1	3.9E+00
	penalty (5) and $q = 1/2$	-6.666661E+00	0.00E+00	5.65E-06	17717	1	4.0E+00
	penalty (5) and $q = 1$	-6.666661E+00	0.00E+00	5.65E-06	17643	1	3.7E+00
	penalty (5) and $q = 2$	-6.666661E+00	0.00E+00	5.65E-06	147756	8	3.1E+01
	in [1][a]	-6.666657E+00	0.00E+00	–	11513		3.3E+01
	in [7][b]	-6.666667E+00	–	–	–	–	7.0E-01[c]
$(P2)$	this study with (4)	(1.893756E+02)	3.53E-05	9.31E-04	170026	18	7.3E+01
	penalty (5) and $q = 1/2$	(2.016560E+02)	3.82E-07	(2.04E+00)	116382	18	5.5E+01
	penalty (5) and $q = 1$	(1.893756E+02)	3.66E-05	9.31E-04	175662	18	7.5E+01
	penalty (5) and $q = 2$	(1.893240E+02)	(1.41E-04)	(3.52E-03)	250082	18	1.0E+02
	in [1][a]	1.892946E+02	0.00E+00	–	13109		1.1E+02
	in [7][b]	1.893116E+02	–	–	–	–	7.0E-01[c]
$(P3)$	this study with (4)	2.000417E+00	0.00E+00	4.16E-04	13901	1	3.0E+00
	penalty (5) and $q = 1/2$	(2.027163E+00)	0.00E+00	(1.36E-02)	351368	18	7.3E+01
	penalty (5) and $q = 1$	2.000417E+00	0.00E+00	4.16E-04	13901	1	3.0E+00
	penalty (5) and $q = 2$	2.000395E+00	2.13E-05	4.37E-04	177651	11	3.7E+01
	in [1][a]	2.000000E+00	0.00E+00	–	4199		3.6E+01
	in [7][b]	2.000000E+00	–	–	–	–	7.0E-01[c]

[a] A multistart based Hooke-and-Jeeves filter method (best solution). [b] A branch-and-bound algorithm that relies on a domain reduction methodology. [c] CPU time in seconds on a Sun SPARC station 2.

References

[1] M.F.P. Costa, F.P. Fernandes, E.M.G.P. Fernandes, and A.M.A.C. Rocha. Multiple solutions of mixed variable optimization by multistart Hooke and Jeeves filter method. *Applied Mathematical Sciences*, 8(44):2163–2179, 2014.

[2] M.F.P. Costa, A.M.A.C. Rocha, R.B. Francisco, and E.M.G.P. Fernandes. Firefly penalty-based algorithm for bound constrained mixed-integer nonlinear programming. *Optimization*, 65(5):1085–1104, 2016.

[3] D.E. Finkel. DIRECT Optimization Algorithm User Guide. Report, Center for Research in Scientific Computation, North Carolina State University, 2003.

[4] D.R. Jones, C.D. Perttunen, and B.E. Stuckmann. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.

[5] S. Lucidi and F. Rinaldi. Exact penalty functions for nonlinear integer programming problems. *Journal of Optimization Theory and Applications*, 145(3):479–488, 2010.

[6] S. Lucidi and F. Rinaldi. An exact penalty global optimization approach for mixed-integer programming problems. *Optimization Letters*, 7(2):297–307, 2013.

[7] H.S. Ryoo and N.V. Sahinidis. Global optimization of nonconvex NLPs and MINLPs with applications in process design. *Computers and Chemical Engineering*, 19(5):551–566, 1995.

Proceedings of GOW'16, pp. 101 – 104.

Improvements to the Supporting Hyperplane Optimization Toolkit Solver for Convex MINLP

Andreas Lundell[1], Jan Kronqvist[2] and Tapio Westerlund[2]

[1]*Mathematics and Statistics, Faculty of Science and Engineering,*
Åbo Akademi University, 20500 Åbo, Finland andreas.lundell@abo.fi

[2]*Process Design and Systems Engineering, Faculty of Science and Engineering,*
Åbo Akademi University, 20500 Åbo, Finland {jan.kronqvist, tapio.westerlund}@abo.fi

Abstract In this paper, some enhancements to the supporting hyperplane optimization toolkit (SHOT) solver for convex MINLP are described. These improvements, *e.g.*, utilizing certain features of the subsolvers as well as relaxation strategies, increase the computational efficiency of the solver.

Keywords: Convex MINLP, Supporting Hyperplane Optimization Toolkit (SHOT), Supporting hyperplanes, Cutting planes

1. Introduction

The extended supporting hyperplane (ESH) algorithm [2, 3] is an iterative method for solving convex mixed-integer nonlinear programming (MINLP) problems to global optimality. This is accomplished by solving mixed-integer linear programming (MILP) subproblems describing a linear relaxation of the nonlinear feasible set. The linear relaxations, in the form of supporting hyperplanes, are generated on points of the boundary of the nonlinear feasible region found by performing a linesearch between the solution points of the MILP problem and an interior point to the integer-relaxed MINLP problem. The supporting hyperplane generated excludes the previous MILP solution point and improves the linear approximation of the feasible region. When this procedure is repeated the solution of the MILP subproblems will converge to the global solution of the MINLP problem. Performing a linesearch to obtain a supporting hyperplane, was proposed for continuous nonlinear programming (NLP) in [6], and for MINLP in [5]. The ESH algorithm is also similar to the extended cutting plane (ECP) algorithm [7], with the main difference being that in the latter, cutting planes are utilized instead of supporting hyperplanes and thus no linesearch is performed.

The ESH algorithm is implemented along with primal bound strategies in the solver SHOT (supporting hyperplane optimization toolkit). SHOT utilizes several of the open source projects made available through the COIN-OR initiative, and the goal is to release the solver as open source as well. In a comparison with other state-of-the-art MINLP solvers on all 333 convex MINLP problems in MINLPLib2 [4], SHOT proved to be very effective [2]. Some of the recent additions and hints at the future development of SHOT are discussed in this paper. These enhancements include: solving mixed-integer quadratic programming (MIQP) subproblems, extending the usage of lazy constraints in the MILP subsolvers and extensions to the integer-relaxation strategies employed.

2. Utilizing MIQP capabilities of the subsolvers

In algorithms describing the feasible region by an outer polyhedral approximation through hyperplanes or cutting planes, such as ECP, ESH or outer approximation (OA) [1], MILP sub-

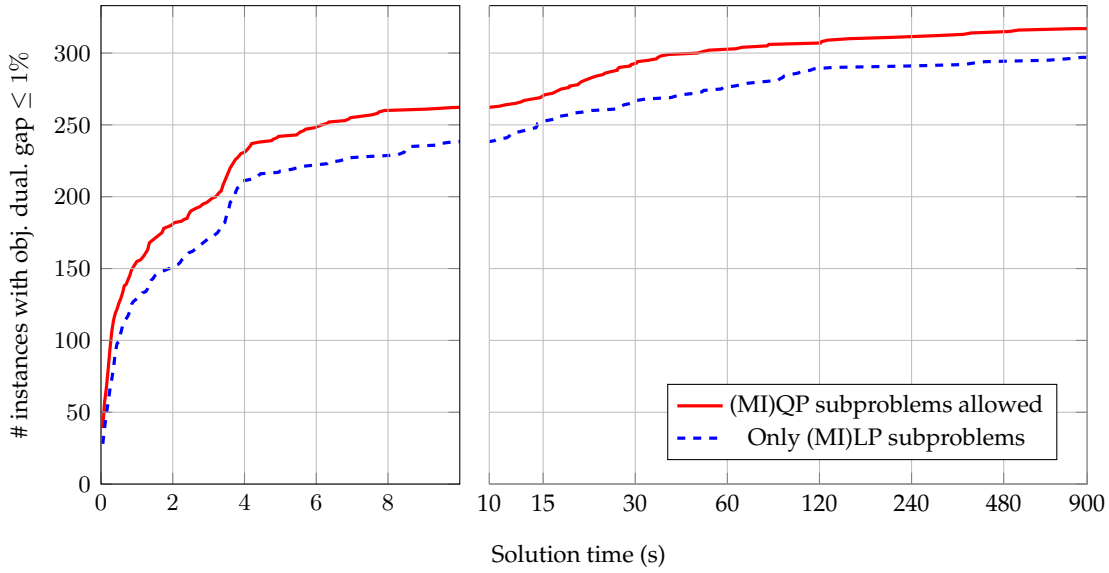


Figure 1. A solution profile illustrating the number of instances solved to an objective duality gap of $\leq 1\%$ when solving (MI)QP subproblems instead of (MI)LP ones.

problems are usually solved by calling on subsolvers. However, some MILP solvers (*e.g.*, CPLEX and Gurobi), can actually solve MIQP problems as well. If the MINLP problem has a quadratic objective function this will often greatly enhance the performance of the MINLP solver, as the nonlinear quadratic objective can be directly expressed in the MIQP subproblem without linear approximations. In SHOT this means that a MIQP problem is solved in each iteration instead of an MILP problem. Many of the problems in MINLPLib2 have a quadratic objective function and linear constraints only, meaning that these problems can be solved to optimality in one iteration only. Benchmark results obtained when running SHOT with or without utilizing MIQP subsolvers is illustrated in Figure 1. All benchmarks in this paper were performed on a Linux-based 64 bit computer using an Intel Xeon 3.6 GHz processor with four physical and eight logical cores as well as 32 GB system memory.

Note that CPLEX and Gurobi can also solve convex mixed-integer quadratically constrained quadratic programming (MIQCQP) problems, and therefore, it is possible to let the subsolved directly handle quadratic constraints while the rest of the nonlinear constraints are handled by the ESH method. However, this was not considered in the benchmark mentioned above.

3. Utilizing an adaptive LP step

By default in SHOT, the variable integer constraints are ignored in the first iterations, and LP (or QP) problems are thus solved instead of mixed-integer problems. For a problem not dominated by the integer requirements this will quickly give a good initial description of the nonlinear feasible region as integer-relaxed problems are often magnitudes faster to solve. After some criterion has been met, *i.e.*, the maximum nonlinear constraint violation is below a certain threshold or an iteration limit has been reached, the integer requirements are included. This strategy has proved to work well, as indicated in Figure 2. In this benchmark, the maximum number of integer-relaxed iterations was set at 300 and the maximum nonlinear constraint termination tolerance at 0.001.

In addition to this initial relaxation strategy, integer-relaxed problems can also be solved in later iterations. If the same solution for the discrete variables is obtained in several subsequent iterations, the MILP solver can be viewed as solving LP instead of MILP problems, and to speed up the solution process the integer variables can therefore be fixed and an NLP problem

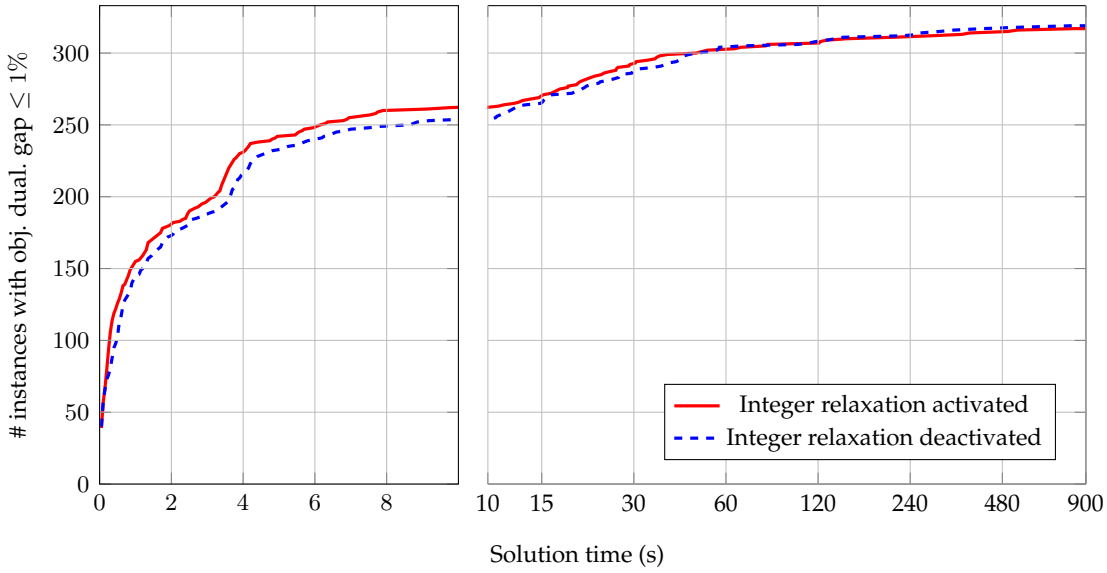


Figure 2. A solution profile illustrating the number of instances solved to an objective duality gap of $\leq 1\%$ when initially solving integer-relaxed (*i.e.*, QP/LP) subproblems.

with the original constraints of the MINLP problem solved instead. The solution to this NLP problem gives a primal solution candidate that can be used as a cutoff value for the MILP solver as long as it is within the interior of the nonlinear feasible set, *i.e.*, fulfills all nonlinear constraints. If it is on the boundary or, due to numerical tolerances, in the exterior of this set, a supporting hyperplane can be added instead in this point to avoid numerical difficulties.

An alternative technique for handling that the same integer solution is returned in several subsequent iterations, is to fix the integer variables to their respective values and instead solve a series of LP or QP problems and generating supporting hyperplanes until the maximum constraint violation of the new solution point is less than a specified epsilon tolerance. This is often much faster than solving an NLP problem, however it comes at the cost that we most often do not get a new primal solution candidate unless it is situated on the boundary of the nonlinear feasible region.

4. Adding supporting hyperplanes as lazy constraints

MILP solvers such as CPLEX or Gurobi have the possibility to add constraints as so-called lazy constraints. These are only considered when verifying a solution, and lazy constraints are thus not normally included in the MILP model until they become active. Since SHOT often creates several new constraints in each iteration (if several integer-feasible solutions are found), the number of constraints increases rapidly, directly affecting the solution time of each iteration. However, since the information contained in the early linearizations may become practically redundant in later ones, the performance can be increased by adding the hyperplanes as lazy constraints and let the MILP solver decide whether to use them or not. The results, when using the initial relaxed step described in Section 3 and shown in Figure 2, indicate that for problems with a long solution time (about > 60 s in the benchmark) the large amount of initial extra linearizations created have a negative impact on the performance in the long run. This further motivates the usage of lazy constraints instead of traditional ones.

There are two main strategies on how to use the lazy constraint technique in SHOT: The first is to add the constraints as lazy instead of as regular ones in each iteration and solve the subproblems normally. The second option is to integrate the creation of supporting hyperplanes as lazy constraints through the means of so-called callbacks in the subsolver. In the latter case,

Table 1. The differences in solution times for SHOT when applied to some convex instances in MINLPLib2 and utilizing the constraint strategies mentioned in Section 4.

	Solution time (s) when adding the constraints as		
Problem name	*normal constraints*	*lazy constraints*	*lazy constraints through callback*
`batchs201210m`	14.1	23.8	**7.6**
`clay0305h`	13.7	17.7	**4.2**
`flay05h`	**10.8**	33.6	29.5
`fo7`	13.0	24.5	**11.4**
`sssd25-08`	99.7	25.6	**3.3**
`syn30m04h`	14.1	22.6	**4.7**
`tls4`	21.8	19.3	**2.4**

whenever the MILP solver finds a new integer-feasible solution, a callback function is called that performs a normal ESH linesearch step between this point and the internal point, and generates a new supporting hyperplane on the boundary of the nonlinear feasible set. This hyperplane is then added as a lazy constraint, removing the integer-feasible point, after which the subsolver will continue the search for a new integer-feasible solution without needing to rebuild the branching tree. The second strategy is actually a very easy way of extending, *e.g.*, CPLEX or Gurobi to also being able to handle convex MINLP problems.

A comparison of performance for the lazy constraint functionality is provided in Table 1. It illustrates that adding lazy constraints through callbacks is preferred in most cases.

Acknowledgments

Jan Kronqvist acknowledges the financial support from the Finnish Graduate School in Chemical Engineering.

References

[1] M. A. Duran and I. E. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36(3):307–339, 1986.

[2] J. Kronqvist, A. Lundell, and T. Westerlund. The extended supporting hyperplane algorithm for convex mixed-integer nonlinear programming. *Journal of Global Optimization*, 64(2):249–272, 2015.

[3] A. Lundell, J. Kronqvist, and T. Westerlund. An extended supporting hyperplane algorithm for convex MINLP problems. In L.G. Casado, I. García, and E.M.T. Hendrix, editors, *Proceedings of the XII Global Optimization Workshop*, 2014.

[4] MINLP Library 2, 4 2016.

[5] R. Pörn. *Mixed Integer Non-Linear Programming: Convexification Techniques and Algorithm Development*. PhD thesis, Åbo Akademi University, 2000.

[6] A. F. Veinott Jr. The supporting hyperplane method for unimodal programming. *Operations Research*, 15(1):147–152, 1967.

[7] T. Westerlund and F. Petterson. An extended cutting plane method for solving convex MINLP problems. *Computers & Chemical Engineering*, 19:131–136, 1995.

Proceedings of GOW'16, pp. 105 – 108.

Multiobjective Based Scoring Function for Ligand Based Virtual Screening*

Savíns P. Martín[1], Juana L. Redondo[1], Helena den-Haan A.[2], Horacio Pérez-Sánchez[2] and Pilar M. Ortigosa [1]

[1] *University of Almería, Agrifood Campus of International Excellence, ceiA3, Almería, Spain*
{savinspm,jlredondo,ortigosa}@ual.es

[2] *Bioinformatics and High Performance Computing Research Group (BIO-HPC), Universidad Católica San Antonio de Murcia (UCAM), Murcia, Spain*
{hdenhaan,hperez}@ucam.edu

Abstract The comparison of diverse descriptors among molecules forms the basis of Ligand Based Virtual Screening methods (LBVS). In these techniques, a query molecule is processed against large compound libraries containing up to millions of chemical compounds. After screening, subsets of compounds with descriptor values close to the ones of the query compound will be subjected to experimental characterization. This technique (LBVS) has shown success in many different drug discovery scenarios and it is nowadays a very active research field within computational drug discovery. Nonetheless, many of these LBVS methods are limited by single property or molecular descriptor comparison between query and compound databases. This mono-objective optimization methodology imposes serious limitations to drug discovery methods since it has been demonstrated that many discovered drugs share simultaneously several descriptors values at once with query compounds. Therefore the implementation of multi-objective optimization methods in LBVS is mandatory and of outstanding interest on this area. In this work in progress, a multi-objective evolutionary algorithm has been designed. Its aim is to quickly obtain a fixed size set approximating the complete Pareto-front. It adapts ideas from different multi-objective optimization evolutionary algorithms, but also incorporates new devices. In order to analyze the algorithm performance, the new method has been compared to algorithms from the state-of-the-art. According to preliminary computational experiments, the performance of the new algorithm is promising.

Keywords: Multiobjective Optimization, Computational Chemistry, Drug discovery, Ligand Based Virtual Screening

1. Introduction

Virtual Screening (VS) techniques allow to provide predictions about which chemical compounds might interact with a given protein target in some specified way and thus achieving the desired biological function. VS techniques are mainly divided into Ligand Based Virtual Screening (LBVS) and Structure Based Virtual Screening (SBVS).

SBVS methods require detailed structural information about the target protein and can not be applied in situations where this data is not available. Unfortunately this issue is very common for instance in the case of membrane proteins such as GPCRs (G protein-coupled

*This work has been funded by grants from the Spanish Ministry of Economy and Competitiveness (TIN2015-66680-C2-1-R), Junta de Andalucía (P11-TIC7176 and P12-TIC301), Fundación Séneca–Agencia de Ciencia y Tecnología de la Región de Murcia under Projects 19419/PI/14 and 18946/JLI/13, and by the Nils Coordinated Mobility under grant 012-ABEL-CM-2014A, in part financed by the European Regional Development Fund (ERDF). Powered@NLHPC: This research was partially supported by the supercomputing infrastructure of the NLHPC (ECM-02). The authors also thankfully acknowledge the computer resources and the technical support provided by the Plataforma Andaluza de Bioinformática of the University of Málaga. This work was partially supported by the computing facilities of Extremadura Research Centre for Advanced Technologies (CETACIEMAT), funded by the European Regional Development Fund (ERDF). CETACIEMAT belongs to CIEMAT and the Government of Spain. Juana López Redondo is a fellow of the Spanish 'Ramón y Cajal' contract program, co-financed by the European Social Fund.

receptor), which are of the highest pharmacological relevance. In such cases one can recur to LBVS methods, where only information about known ligands (actives and inactives, agonists and antagonists, etc) is exploited in order to predict new bioactive compounds against selected protein targets. LBVS methods will therefore consider all existing available information (structural, physico-chemical parameters, binding affinities, etc) about known active and inactive compounds, and this information will be referred to as molecular descriptors. There exist a large number of molecular descriptors of potentials used to compare molecules, as for example *Shape similarity, Electrostatic similarity, Atomic property fields, Aromatic potential, Desolvation potential*, etc.

In practice, the comparison of diverse descriptors among molecules lies in the core of the diverse LBVS methods. In these techniques, a query molecule is processed against large compound libraries containing up to millions of chemical compounds. After screening, subsets of compounds with descriptor values close to the ones of the query compound will be considered for experimental studies. For these comparisons the properties in 3D space and ligand flexibility must be taken into account. This implies that in order to measure the descriptor similarity between the query and database molecule, the optimization algorithm must explore the translation and rotation of the query molecule in addition to the potential or descriptor similarity that the molecule may adopt in three-dimensional (3D) space. So, the algorithm must find the optimal alignment that shows the highest descriptor similarity. In several cases, the optimization algorithm approach has consisted on defining a multistart algorithm where each of the random generated points call to a good local optimizer.

Nonetheless, many of these LBVS methods are limited by single property or molecular descriptor comparison between query and compound databases. This mono-objective optimization methodology imposes serious limitations to drug discovery methods since it has been demonstrated that many discovered drugs share simultaneously several descriptors values at once with query compounds. Therefore the implementation of multi-objective optimization methods in LBVS is mandatory and of outstanding interest on this area.

In this work in progress, a multi-objective evolutionary algorithm has been designed; we deal with a bi-objective problem, where the Three-Dimensional Shape [1, 2] and the Electrostatic Similarity [3] are the two objective functions which are optimized simultaneously. Our main aim is to quickly obtain a fixed size set approximating the complete Pareto-front. This approach adapts ideas from different multi-objective optimization evolutionary algorithms, but also incorporates new devices.

2. The Multi-objective evolutionary algorithm

In this paper, we deal with a bi-objective problem. A general nonlinear bi-objective problem can be formulated as follows:

$$\begin{array}{ll} \min & \{f_1(y), f_2(y)\} \\ \text{s.t.} & y \in S \subseteq \mathbb{R}^n \end{array} \tag{1}$$

where $f_1, f_2 : \mathbb{R}^n \longrightarrow \mathbb{R}$ are two real-valued functions. Let us denote by $f(y) = (f_1(y), f_2(y))$ the vector of objective functions and by $Z = f(S)$ the image of the feasible region.

When dealing with multi-objective problems we need to clarify what 'solving' a problem means. Some widely known definitions to explain the concept of solution of (1) follow.

Definition 1. *A feasible vector $y^* \in S$ is said to be* efficient *iff there does not exist another feasible vector $y \in S$ such that $f_l(y) \leq f_l(y^*)$ for all $l = 1, 2$, and $f_j(y) < f_j(y^*)$ for at least one index j ($j = 1$ or 2). The set S_E of all the efficient points is called the* efficient set *or* Pareto-set. *If y_1 and y_2 are two feasible points and $f_l(y_1) \leq f_l(y_2), l = 1, 2$, with at least one of the inequalities being strict, then we say that y_1* dominates *y_2.*

Efficiency is defined in the decision space. The corresponding definition in the criterion space is as follows:

Definition 2. *An objective vector $z^* = f(y^*) \in Z$ is said to be* non-dominated *iff y^* is efficient. The set Z_N of all non-dominated vectors is called the* non-dominated set *or* Pareto-front. *If y_1 and y_2 are two feasible points and y_1 dominates y_2, then we say that $f(y_1)$ dominates $f(y_2)$.*

Solving (1) means obtaining the whole efficient set, that is, all the points which are efficient. To this aim, exact general methods should be used [4, 5]. However, considering the high computational cost of the problem to solve as well as the large memory requirements of exact methods, heuristic algorithms seem to be the best choice.

The (meta)heuristic methods and, in particular the evolutionary algorithms, allow to obtain good approximations of the efficient set. This is due to their ability to find multiple efficient solutions in one single simulation run. In this work in progress, a multiobjective evolutionary algorithm is presented to deal with this hard-to-solve optimization problem. It adapts some concepts from other evolutionary algorithms (EAs) devised to cope with single-objective optimization problems. Therefore, a niching technique aiming at the preservation of solution diversity in objective space has been implemented. It is remarkable that the issue of population diversity in parameter space has been gaining attention because several scientific domains, including ligand-virtual screening and drug discovery, are interested in producing solutions that differ [6]. Our method also combines evolutionary algorithms properties with local search techniques that are applied to accelerate the convergence to different local optima. Our algorithm also includes ideas from other typical MOEAs, as the establishment of Pareto-ranks, the determination of Non-dominated solutions, and the use of distance metrics to compute the estimation of the density of solutions during the selection procedure (as in [7]).

A comprehensive computational study is in progress to compare the new multiobjective algorithm with the well-known NSGA-II [7] and SPEA2 [8] algorithms, which have become the reference algorithms in the multi-objective evolutionary computation community. Additionally, two other state-of-the-art algorithms have been included in the comparison: the algorithms MOEA/D [11] and SMS-EMOA [10], which have proved to be very competitive in different studies. The implementation of those four algorithms in the platform j-Metal [9] has been used for the evaluation. Following the existing performance indicators in literature, the comparisons have been accomplished in terms of effectiveness, i.e. in terms of *quality* of the obtained approximations of the Pareto-front. The *modus operandi* has been to provide the algorithms with a budget in the number of function evaluations and to obtain a fixed size approximation of the Pareto-front.

For the assessment and comparison of Pareto-set approximations, global indicators (hypervolume), proximity indicators (average distance and $I^1_{\epsilon+}$) and also dispersion indicators (spread and spacing) have been computed.

3. Summary

In this work, a multi-objective evolutionary algorithm will be presented. It is designed to deal with a bi-objective problem, where the Three-Dimensional Shape and the Electrostatic Similarity of two molecules must be optimized simultaneously. The performance of the algorithm has been analyzed through an extensive computational study. The preliminary results are promising.

References

[1] X. Yan, J. Li, Z. Liu, M. Zheng, H. Ge, and J. Xu. Enhancing Molecular Shape Comparison by Weighted Gaussian Functions. *Journal of Chemical Information and Modeling*, 53(8):1967–1978, 2013.

[2] I.S. Haque, and V.S. Pande. PAPER-Accelerating parallel evaluations of ROCS. *Journal of Computational Chemistry*, 31(1):117–132, 2010.

[3] S.W. Muchmore, A.J. Souers and I. Akritopoulou-Zanze. The Use of Three-Dimensional Shape and Electrostatic Similarity Searching in the Identification of a Melanin-Concentrating Hormone Receptor 1 Antagonist. *Chemical biology & drug design*, 67(2): 174–176, 2006.

[4] J. Fernández, and B. Tóth. Obtaining an outer approximation of the efficient set of nonlinear biobjective problems. *Journal of Global Optimization*, 38(2):315–331, 2007.

[5] J. Fernández, and B. Tóth. Obtaining the efficient set of nonlinear biobjective optimization problems via interval branch-and-bound method. *Computational Optimization and Applications*, 42(3):393–419, 2009.

[6] C.A. Nicolaou, J. Apostolakis and C.S. Pattichis. De novo drug design using multi-objective evolutionary graphs. *Journal of Chemical Information and Modeling* 49: 295–307, 2009.

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[8] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In K. C. Giannakoglou, D. T. Tsahalis, J. Périaux, K. D. Papailiou, and T. Fogarty, editors, *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pages 95–100, Athens, Greece, 2002. International Center for Numerical Methods in Engineering (CIMNE).

[9] J.J. Durillo, and A.J. Nebro. jMetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, 42:760–771, 2011.

[10] N. Beume, B. Naujoks, and M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.

[11] Q. Zhang, and H. Li. MOEA/D: A Multi-objective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

Proceedings of GOW'16, pp. 109 – 112.

`lsmear`: A Variable Selection Strategy for Interval Branch and Bound Solvers

Ignacio Araya[1] and Bertrand Neveu[2]

[1] *Pontificia Universidad Católica de Valparaíso, Escuela de Ingeniería Informática, Chile,* ignacio.araya@pucv.cl

[2] *Imagine LIGM Université Paris–Est, France,* Bertrand.Neveu@enpc.fr

Abstract Smear-based variable selection strategies are well-known and commonly used by branch-and-prune interval-based solvers. They estimate the impact of the variables on each constraint of the system by using the partial derivatives and the sizes of the variable domains. Then they aggregate these values, in some way, to estimate the impact of each variable on the whole system. The variable with the greatest impact is then selected. A problem of these strategies is that they, generally, consider all constraints equally important.

In this work, we propose a new variable selection strategy which first weights the constraints by using the optimal Lagrangian multipliers of a linearization of the original problem. Then, the impact of the variables is computed with a typical smear-based function but taking into account the weights of the constraints. The strategy is tested on classical benchmark instances outperforming significantly the classical ones.

Keywords: Global optimization, Branch and bound, Variable selection, Lagrangian multipliers, Smear function

1. Introduction

This paper deals with continuous global optimization (nonlinear programming) deterministically handled by interval branch and bound (B&B). The problem is defined by: $\min_{x\in \mathsf{x}} f(x)\ s.t.\ g(x) \leq 0$, where $f : \mathbb{R}^n \to \mathbb{R}$ is the real-valued objective (non convex) function and $g : \mathbb{R}^n \to \mathbb{R}^m$ is a vector-valued (non convex) function.[1] $x = (x_1, ..., x_i, ...x_n)$ is a vector of variables varying in a domain (i.e., a box) x[2]. For performance and simplicity considerations, a variable x_o, with initial domain $\mathsf{x}_o = [-\infty, +\infty]$ is included in the set of variables x and an additional constraint $f(x) = x_o$ is included in the set of constraints (actually, functions $f(x) - x_o$ and $x_o - f(x)$ are included in g). Finally, we solve an equivalent problem:

$$\min_{x\in \mathsf{x}} x_o \ \ s.t.\ \ g(x) \leq 0 \tag{1}$$

Several works have been proposed for finding good branching strategies ([2, 3, 4, 5, 7]). Smear-based methods [3, 4] use information on the system to obtain the variable with the greatest *impact*. The impact of a variable x_i on a function g_j is computed by means of the *smear value*. Consider that the current node is associated with box x; the smear value is given by: $\text{smear}(x_i, g_j) = |\boldsymbol{J}_{ji}| * \text{wid}(\mathsf{x}_i)$, where $\boldsymbol{J}_{ji}$ is an interval overestimate of the range of the partial derivative $\frac{\partial g_j}{\partial x_i}$ in x. $|\boldsymbol{J}_{ji}|$ is the *magnitude* of the interval $\boldsymbol{J}_{ji}$, i.e., $|\boldsymbol{J}_{ji}| = \max(|\underline{J_{ji}}|, |\overline{J_{ji}}|)$.

Selection methods based on the smear value select the variable that maximizes an aggregation of this value in the whole system.

Tawarmalani and Sahinidis [6] present an algorithm called *ViolationTransfer*, to estimate the impact of a variable on the problem. *ViolationTransfer* works with the Lagrangian function of

[1] The branching strategies proposed in this paper can also apply to problems having equality constraints.

[2] An interval $\mathsf{x}_i = [\underline{x_i}, \overline{x_i}]$ defines the set of reals x_i s.t. $\underline{x_i} \leq x_i \leq \overline{x_i}$. A *box* x is a Cartesian product of intervals $\mathsf{x}_1 \times ... \times \mathsf{x}_i \times ... \times \mathsf{x}_n$.

a relaxation of the problem and an optimal solution of the relaxation $x^\star$. For each variable, an interval $\mathsf{x}_i^v \subset \mathsf{x}_i$ is defined. x^v is the smallest box such that it contains $x^\star$ and each univariate constraint $g_j(x_1^\star, ..., x_{i-1}^\star, x_i, x_{i+1}^\star, x_n^\star) \leq 0$ $(j = 1..m)$ is feasible for at least one value in x_i^v.

Then, for each variable x_i, the difference between the bounds of the image of the Lagrangian function over the interval x_i^v is estimated. In each estimation all the variables are fixed except x_i. The assumption is that branching on the variable maximizing the image width is likely to improve the lower bound of the objective function in the subproblems.

In this article we propose `lsmear`, a new variable selection strategy for interval B&B solvers. In a few words, the method selects the variable maximizing the smear value of the Lagrangian function of the problem. In the Lagrangian function, the Lagrange multipliers are replaced by the dual optimal of a linear approximation of the problem. Related to the *ViolationTransfer* strategy our approach has some important differences:

1. `lsmear` uses a simple *linear approximation* of the original problem instead of sophisticated convex relaxation techniques.
2. `lsmear` estimates the impact of each variable in the Lagrangian function of the *original problem*. The estimated impact is computed by using the smear value.
3. The computation of x^v requires the solver to use a reformulated problem in which multidimensional functions are replaced with either univariate or bilinear functions [6]. For the moment, and in order to maintain the simplicity and generality of the approach, `lsmear` uses directly x instead of x^v.

2. `lsmear`, a Smear-based strategy using optimal Lagrange multipliers

A main issue related to the smear-based strategies is that these strategies consider all the constraints equally important. To overcome this issue we propose to estimate the impact of the constraints in the system by using the optimal Lagrange multipliers of a linear approximation of the original problem.

The `lsmear` method works in two phases. First, a linearization of the global optimization problem is generated. Each function $g_j(x)$ is approximated by using the first order term of its Taylor expansion around the midpoint of the box, i.e.,

$$gl_j(x) = g_j(\text{mid}(\mathsf{x})) + \sum_{i=1}^{n} \text{mid}(\boldsymbol{J}_{ji}).(x_i - \text{mid}(\mathsf{x}_i))$$

where $\boldsymbol{J}_{ji}$ is an interval overestimate of the image of $\frac{\partial g_j}{\partial x_i}$ over x. Note that instead of using the partial derivatives in the midpoint of the box we use the midpoint of the overestimate of the partial derivatives (i.e., $\text{mid}(\boldsymbol{J}_{ji})$).

The generated linear optimization problem includes the bound constraints, i.e., $\underline{x_i} \leq \mathsf{x}_i \leq \overline{x_i}$, and it is solved by using the simplex method[3]. If an optimum exists, then a second phase is carried out. In this phase, the strategy computes the smear value of the following function:

$$L(x) = x_o + \sum_{i=1}^{m} \lambda_j^* g_j(x)$$

where λ^* corresponds to the dual optimal solution of the linear problem. The function L is equivalent to the Lagrangian of (1) but the Lagrange multipliers have been replaced by the

[3] Actually we need to solve the dual problem, however we use a linear solver which solves both, the primal and the dual problems

optimal Lagrange multipliers of the linear approximation. Thus, the problem of minimizing $L(x)$ can be seen as a rough approximation of the original optimization problem. The interesting thing about L is that it aggregates the objective function and the constraints of the problem in only one function, thus, computing the smear value of each variable in L offers an estimation of the impact of the variable in the original problem. We believe that this estimation is fairer than the one computed by the classical smear-based strategies because each constraint is, in a certain way, being weighted according to its influence on the optimal value. Finally, as well as the other Smear-based strategies, the variable with the greatest impact is selected for bisection.

Input: $(\mathsf{x}, \boldsymbol{J}, g)$
`// Phase 1: linearization and solving the linear program`
$gl \leftarrow g(\text{mid}(\mathsf{x})) + \text{mid}(\boldsymbol{J}).(x - \text{mid}(\mathsf{x}))$
$\mathsf{t} \leftarrow x_o$; $\mathsf{x}_o \leftarrow [-\infty, \infty]$
$\lambda^* \leftarrow$ optimize($\min x_o$, subject to: $\underline{x_i} \leq x_i \leq \overline{x_i},\ gl_j(x) \leq 0$)
$\mathsf{x}_o \leftarrow \mathsf{t}$
if $\lambda^* \neq \varnothing$ **then**
  `// Phase 2: computing the impact of` $L(x)$
  for $i \in \{1..n\}$ **do**
    $\boldsymbol{D} \leftarrow \lambda^*_i$
    for $j \in \{1..m\}$ **do**
      $\boldsymbol{D} \leftarrow \boldsymbol{D} + \lambda^*_{n+j}.\boldsymbol{J}_{ji}$
    $I \leftarrow |\boldsymbol{D}.\text{wid}(\mathsf{x}_i)|$
    if $I >$ *max_impact* **then**
      max_impact $\leftarrow |\mathcal{I}|$
      var $\leftarrow i$
  return var
else
  return smearsum$(\mathsf{x}, \boldsymbol{J})$

Algorithm 1: `lsmear`

Algorithm 1 shows our approach. $\boldsymbol{J}$ corresponds to the Jacobian matrix which contains the interval overestimates of the partial derivatives over x. In the linear program (Phase 1), the interval related to the objective variable is unbounded to enhance the chances for successfully finding an optimal solution. In Phase 2, for each variable x_i we first compute $\boldsymbol{D}$, which is an interval overestimate of $\frac{\partial L}{\partial x_i}$ over x. The overestimate is obtained by adding the products of the interval partial derivatives on each constraint ($\boldsymbol{J}_{ji}$) and the corresponding dual optimal value (λ^*_{n+j}). $\boldsymbol{D}$ is initialized with the dual optimal value related to the bounded constraint, i.e., λ^*_i (the partial derivative related to the i-th bound constraint over the variable x_i is 1). Then, the smear impact of the variable is computed as the magnitude of the product of the interval partial derivative and the width of the related interval. The variable with the maximum impact is saved and returned. If the linear program does not have solutions or if the optimal value is unbounded, then the `smearsum` method is launched instead. This method uses the same Jacobian matrix received by the `lsmear` one.

3. Experiments

In order to validate our approach, we implemented `lsmear` in IbexOpt, a state-of-the-art optimizer of the Interval-Based EXplorer library (Ibex ([1])). All the experiments were run on a server PowerEdge T420, with 2 quad-processors Intel Xeon, 2.20 GHz and 8 GB RAM.

The instances were selected from the series 1 and 2 of the COCONUT constrained global optimization benchmarks.[4] We selected all the problems solved by some strategy in a time comprised between 2s and 3600s (66 instances). Each strategy was run 5 times on each instance and the average CPU time was reported.

We compared the proposed strategy `lsmear` with some of the classical variable selection strategies: `round-robin` (rr), `largest-first` (lf), `smearsum` (ssum), `smearmax` (smax) and `smearsumrel` (ssr). Figure 1 summarizes the comparison among the six strategies.

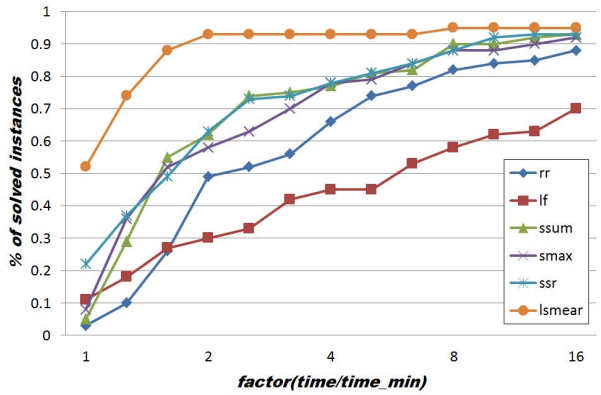


Figure 1. Performance profile.

Each curve reports the percentage of instances solved by the corresponding strategy in less than *factor* times the best reported CPU time. From the results we observe that `lsmear` clearly outperforms all the classical variable selection strategies. Also note that more than 90% of the instances are solved by `lsmear` in less than twice the best CPU time reported by all the strategies.

References

[1] G. Chabert and L. Jaulin. Contractor Programming. *Artificial Intelligence*, 173:1079–1100, 2009.

[2] Tibor Csendes and Dietmar Ratz. Subdivision direction selection in interval methods for global optimization. *SIAM Journal on Numerical Analysis*, 34(3):922–938, 1997.

[3] Eldon Hansen and G William Walster. *Global optimization using interval analysis: revised and expanded*, volume 264. CRC Press, 2003.

[4] R Baker Kearfott and Manuel Novoa III. Algorithm 681: INTBIS, a portable interval Newton/bisection package. *ACM Transactions on Mathematical Software (TOMS)*, 16(2):152–157, 1990.

[5] R Moore. Interval analysis, vol. 60, 1966.

[6] Mohit Tawarmalani and Nikolaos V Sahinidis. Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical programming*, 99(3):563–591, 2004.

[7] G. Trombettoni, I. Araya, B. Neveu, and G. Chabert. Inner regions and interval linearizations for global optimization. In *AAAI Conference on Artificial Intelligence*, pages 99–104, 2011.

[4] `www.mat.univie.ac.at/~neum/glopt/coconut/Benchmark/Benchmark.html`

Proceedings of GOW'16, pp. 113 – 116.

Merging Flows in Terminal Manoeuvring Areas via Mixed Integer Linear Programming

Mohammed Sbihi[1], Marcel Mongeau[1], Ma Ji[1] and Daniel Delahaye[1]

[1] *ENAC- Université de Toulouse, 7 avenue Edouard Belin, 31055 Toulouse cedex 4, France*
[mohammed.sbihi|marcel.mongeau|ji.ma|daniel.delahaye]@recherche.enac.fr

Abstract This paper focuses on the aircraft merging and sequencing problem at Terminal Manoeuvring Areas through the use of Controlled Time of Arrival (CTA). A Mixed-Integer Linear Programming formulation is proposed in order to minimize the number of non achievable CTAs while maintaining separation between aircraft with regard to the horizontal, wake-turbulence, and runway occupancy time constraints. Computational experiments performed on real-world case studies of Paris Charles De-Gaulle (CDG) airport show that the approach is viable.

Keywords: Aircraft merging and sequencing, Controlled time of arrival, Optimization model, MILP.

1. Introduction

The Terminal Manoeuvring Area (TMA) is a designated area of controlled airspace surrounding one or several airports where there is a high volume of traffic. It is designed to handle aircraft arriving to and departing from airports. TMA is identified by many researchers as one of the most critical parts of the air transportation system. Therefore, there is a need for improving efficiency and increasing capacity by using efficient approaches and algorithms. In this paper, we focus on the problem of merging flight flows into the TMA. During the transition from the en-route to the terminal airspaces, aircraft arriving from different entry points must be merged and organized into an orderly stream while maintaining a safe separation between them. In moving to the future SESAR concept of Trajectory Based Operations, Air Traffic Controllers (ATC) can merge arrival traffic streams into sequences by the use of so-called *Controlled Times of Arrival* (CTA) at the TMA entry point, called *Initial Approach Fix* (IAF). CTA can be achieved using the airborne *Required Time of Arrival* (RTA) functionality, a feature of modern Flight Management Systems designed to calculate and adjust the speed of the aircraft to arrive at a given point in space at a defined target time. CTAs are determined by ATC (typically using an arrival manager tool) and set when the aircraft is around 150-200NM from touchdown. Such calculations might take into account, among other things, downlinked aircraft *Estimated Time of Arrival* (ETA) (or a time-window $[\mathrm{ETA}_{\min}, \mathrm{ETA}_{\max}]$). In [1], De Smedt *et al.* investigated the application of RTA to a real sequence of arriving aircraft into Melbourne, Australia. They found that pressure on the terminal area would sometimes require aircraft to lose more time than what is possible through the RTA capability, and hence require additionally a recourse to other conventional sequencing techniques to provide a sequence resolution.

In this paper, we consider the problem of assigning CTAs to arriving aircraft in order to reduce the number of CTAs that fall outside the $[\mathrm{ETA}_{\min}$,$\mathrm{ETA}_{\max}]$ windows subject to operational constraints related to wake turbulence, horizontal separation, and runway occupancy time. This problem is very close to the problem of minimizing the number of late jobs on one machine, which is known to be NP-hard in the strong sense [2]. We propose a Mixed-Integer Linear Programming (MILP) formulation of this problem and report computational experiments on real-world case studies from Paris CDG airport using Gurobi optimization solver.

2. MILP formulation

2.1 Given data

In TMA the traffic is arranged so that there is a basic segregated and separated Terminal Area flight path structure with arriving traffic coming through one of a number of IAF. For example, Figure 1 displays the arriving network structure at Paris CDG airport to runway 26L. In this arrival procedure, four routes, originating from IAF MOPAR, LORNI, OKIPA and BANOX, fuse into one single route towards the runway.

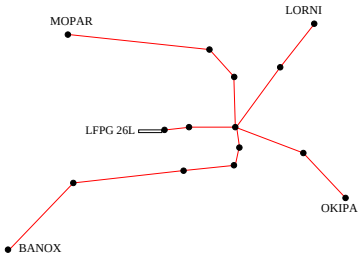


Figure 1. The route structure model for LFPG runway 26L.

We model the route structure as a graph, or to be more precise as a tree here, $G = (V, E)$ in which the aircraft are allowed to fly. The vertex set V is the set of way-points and E is the set of the arcs interconnecting these way-points by a straight-line segments . The arcs are assigned the natural orientation towards the root r, which plays the role of the runway threshold. Each other leaf $e \in V$ corresponds to an entry point and admits a unique corresponding path (route) r_e leading to the root. We are given a set of flights (or aircraft), $\mathcal{F} = \{1, \dots |\mathcal{F}|\}$, and for each flight $f \in \mathcal{F}$ the following data is also given:

- e_f: entry way-point at TMA (this determines the route $r_f := r_{e_f}$ of flight f)
- t_f: ETA at the entering point $e_f \in V$
- s_f^u: speed (supposed constant) of f on the arc $u \in E$
- $[t_f - \underline{\Delta}_f, t_f + \overline{\Delta}_f]$: the [ETA$_{\min}$,ETA$_{\max}$] window.

2.2 Optimization variables

For each flight f we associate the continuous variable x_f, representing its assigned CTA, and the binary variable y_f indicating whether the assigned CTA falls outside the [ETA$_{\min}$,ETA$_{\max}$] window ($y_f = 1$) or not ($y_f = 0$). Considering two flights f and g, we have to decide which lands first. Thus, we further introduce the decision variable

$$\delta_{f,g} = \begin{cases} 1 & \text{if } f \text{ lands before } g \\ 0 & \text{otherwise} \end{cases}$$

Remark that $\delta_{f,g}$ decides also the passing order on any node $v \in r_f \cap r_g$. We also introduce auxiliary variable t_f^v, representing the passing time through node v. It is connected to x_f by $t_f^v = x_f + \sum_{u \in r_f^v} \frac{d_u}{s_f^u}$, where the r_f^v contains the arcs of r_f before v, and d_u is the length of u.

2.3 Objective function

The aim is to minimize the total number of non achievable CTAs:

$$\sum_{f \in \mathcal{F}} y_f. \tag{1}$$

2.4 Constraints

The first set of constraints indicates the decision interval for the CTA. Ideally we would ask each x_f to be an achievable CTA, *i.e.* $t_f - \underline{\Delta}_f \leq x_f \leq t_f + \overline{\Delta}_f$. However, such a requirement, will in general render the problem infeasible. Thus, at the price of possibly invoking other conventional sequencing techniques, we relax this constraint by

$$t_f - \underline{\Delta}_f \leq x_f \leq t_f + R_{\max}, \qquad f \in \mathcal{F} \tag{2}$$

where $R_{\max}$ is the maximum moving-backward value of CTA, a user-defined parameter.

Now, considering pairs (f, g) of flights, we have that

$$\delta_{fg} + \delta_{gf} = 1, \qquad f, g \in \mathcal{F}, g > f. \tag{3}$$

In words, either flight f must land before ($\delta_{fg} = 1$) or after ($\delta_{gf} = 1$) flight g. It is trivial to see that, for certain pairs (f, g) of flights, we can decide whether $\delta_{fg} = 1$ or whether $\delta_{gf} = 1$ based on the particular input data in a preprocessing step. The link between x_f and y_f is given by

$$x_f - y_f(R_{\max} - \overline{\Delta}_f) \leq t_f + \overline{\Delta}_f, \qquad f \in \mathcal{F}. \tag{4}$$

Indeed, if the CTA is not achievable, then this constraint implies that $y_f = 1$. Otherwise, both values $y_f = 0$ and $y_f = 1$ are feasible, but in the minimal solution y_f will necessarily be 0.

Operational constraints. In this problem, we consider three separation requirements.

Runway separation constraints. For each ordered pair of flights (f, g) a minimum separation of $\tau_{f,g}$ units must be maintained between the landing times t_f^r and t_g^r of f and g. This minimum separation $\tau_{f,g}$ depends on the wake turbulence categories of f and g. This separation is insured by the following constraint (M denotes a sufficiently large positive constant)

$$t_g^r - t_f^r \geq \tau_{f,g} - (1 - \delta_{f,g})M, \qquad f, g \in \mathcal{F}, f \neq g. \tag{5}$$

Weak-turbulence constraints. For each pair of successive aircraft (f, g), the International Civil Aviation Organization regulates the minimum spacing between them to avoid the danger of wake turbulence. It is a distance-based separation $w_{f,g}$. As the speed is assumed to stay constant throughout one arc, it is sufficient to check this separation constraint at the nodes $v \in r_f \cap r_g$. This is achieved by imposing the following constraint

$$t_g^v - t_f^v \geq \max\left(\frac{w_{f,g}}{s_g^{u_{g,v}^-}}, \frac{w_{f,g}}{s_f^{u_{f,v}^+}}\right) - (1 - \delta_{f,g})M, \qquad v \in r_f \cap r_g; f, g \in \mathcal{F}, f \neq g \tag{6}$$

where $u_{g,v}^-$ (resp. $u_{f,v}^+$) is the arc of r_g incoming to (resp. the arc of r_f outgoing from) node v.

Horizontal separation constraints. Aircraft must satisfy a minimum given horizontal separation, d_h, based on radar (typically $d_h = 3$ NM in the TMA). In order to give a necessary and sufficient condition for the horizontal separation, we need the following assumptions:

(H1) The distance between any two non-adjacent arcs u_1 and u_2 is greater than or equal to d_h.

(H2) For any two distinct adjacent arcs $u_1 = (v_1, v)$ and $u_2 = (v_2, v)$ (or $u_2 = (v, v_2)$), the distance between v_2 and the line segment $[v_1, v]$ and the distance between v_2 and the line segment $[v2, v]$ are greater than or equal to d_h.

Assumption (H1) implies that a *conflict* (the distance between two aircraft is less than d_h) can only occur between two aircraft flying on the same arc or two adjacent arcs. Moreover, in the later case, Assumption (H2) guarantees that a conflict can only occur near the common node. The following lemma takes care of the case where a pair of flights travel on arcs that are adjacent to a common node.

Lemma 1. *Let $u_1 = (v_1, w_1)$, $u_2 = (v_2, w_2)$ two arcs adjacent to a common node $v \in V$. Let θ_{u_1,u_2} be the angle between the vectors $\overrightarrow{v_1w_1}$ and $\overrightarrow{v_2w_2}$. Assume that flight f passes node v before g. Then, there is no conflict between f and g, when f is flying on u_1 while g is flying on u_2 if and only if*

$$t_g^v - t_f^v \geq \Delta_{u_1,u_2}^{f,g}, \tag{7}$$

where $\Delta_{u_1,u_2}^{f,g}$ is defined as follows:

1. *If $u_1 = (v_1, w_1)$ and $u_2 = (v_2, w_2)$ are converging arcs (i.e. $v = w_1 = w_2$), then*

$$\Delta_{u_1,u_2}^{f,g} := \begin{cases} \frac{d_h}{s_{u_2}^g} & \text{if } s_{u_1}^f \cos(\theta_{u_1,u_2}) \leq s_{u_2}^g \\ \frac{d_h \sqrt{(s_{u_1}^f)^2 + (s_{u_2}^g)^2 - 2 s_{u_1}^f s_{u_2}^g \cos(\theta_{u_1,u_2})}}{|\sin(\theta_{u_1,u_2})| s_{u_1}^f s_{u_2}^g} & \text{otherwise.} \end{cases}$$

2. *If $u_1 = (v_1, w_1)$ and $u_2 = (v_2, w_2)$ are serial arcs (i.e. $v = v_1 = w_2$), then*

$$\Delta_{u_1,u_2}^{f,g} := \begin{cases} \max\left(\frac{d^h}{s_{u_1}^f}, \frac{d_h}{s_{u_2}^g}\right) & \text{if } s_{u_1}^f \cos(\theta_{u_1,u_2}) \geq s_{u_2}^g \text{ or } s_{u_2}^g \cos(\theta_{u_1,u_2}) \geq s_{u_1}^f \\ \frac{d^h \sqrt{(s_{u_1}^f)^2 + (s_{u_2}^g)^2 + 2 s_{u_1}^f s_{u_2}^g \cos(\theta_{u_1,u_2})}}{|\sin(\theta_{u_1,u_2})| s_{u_1}^f s_{u_2}^g} & \text{otherwise.} \end{cases}$$

Consequently, the horizontal separation constraint on node v reads

$$t_g^v - t_f^v \geq \Delta_{u_1,u_2}^{f,g} - (1 - \delta_{f,g})M. \tag{8}$$

This constraint must be satisfied for each pair f, g of aircraft, for each node $v \in r_f \cap r_g$, and for each arcs $u_1 \in r_f$, $u_2 \in r_g$ adjacent to v.

3. Computational experiments

We test our approach on real traffic data sample recorded on 5th May 2015 at Paris CDG Airport on runway 26L. We apply our algorithm first at once on the 24-hour data and then on different 2-hour time windows. The MILP model is solved with Gurobi 5.6.3. Computations are performed on an Intel(R) Core(TM) i5-3210M with 2.5GHz and 4Go RAM memory. The results (number of non-achievable CTAs) are given in Table 1 and show that the problem can be efficiently solved.

Table 1. Results obtained on real traffic of the Paris CDG Airport runway 26L (May 5, 2015)

Time windows	0:00 - 24:00	3:00 - 5:00	5:00 - 7:00	7:00 - 9:00	9:00 - 11:00	11:00 - 13:00	13:00 - 15:00	15:00 - 17:00	17:00- - 19:00	19:00 - 21:00
Number of flights	417	32	63	39	51	54	42	48	48	37
Optimal solution	29	3	9	3	8	3	0	0	1	1
Computation time (sec.)	28.6	0.05	31.04	0.03	5.66	0.01	0.03	0.05	0.03	0.03

4. Summary

The problem of minimizing the number of CTAs falling outside the [$\text{ETA}_{\min}$,$\text{ETA}_{\max}$] windows subject to operational constraints is investigated. A MILP formulation and promising preliminary computational experiments on real traffic data were presented.

References

[1] D. De Smedt, J. Bronsvoort, and G. McDonald. Controlled Time of Arrival Feasibility Analysis. Tenth USA/Europe Air Traffic Management Research and Development Seminar, Chicago, Illinois USA , June 2013.

[2] J.K. Lenstra, A.H.G. Rinnooy Kan, and P. Brucker. Complexity of machine scheduling problems. *Annals of Discrete Mathematics*, 1:343-362, 1977.

Proceedings of GOW'16, pp. 117 – 120.

Lifted Polyhedral Approximations in Convex Mixed Integer Nonlinear Programming

Jan Kronqvist[1], Andreas Lundell[2] and Tapio Westerlund[3]

[1] *Åbo Akademi University, Faculty of Science and Engineering, Process Design and Systems Engineering, 20500 Åbo, Finland,* jan.kronqvist@abo.fi

[2] *Åbo Akademi University, Faculty of Science and Engineering, Mathematics and Statistics, 20500 Åbo, Finland,* andreas.lundell@abo.fi

[3] *Åbo Akademi University, Faculty of Science and Engineering, Process Design and Systems Engineering, 20500 Åbo, Finland,* tapio.westerlund@abo.fi

Abstract Here we illustrate the advantage of utilizing a lifted polyhedral approximation, resulting in a tighter polyhedral approximation. The tighter approximation can greatly improve the performance of solvers based on polyhedral approximations, and the benefits are demonstrated by a numerical comparison.

Keywords: Lifted polyhedral approximation, Separable functions, Linear approximation, MINLP

1. Introduction

In the following study we will focus on a specific type of convex mixed integer nonlinear programming (MINLP) instances, where all nonlinear functions are convex and can be separated into convex functions consisting only of a subset of the original variables. For such MINLP instances it is possible to utilize a simple reformulation, here referred to as a lifted polyhedral approximation, where the problem is lifted to an higher dimensional space to obtain a tighter linear approximation which can significantly enhance the performance of some MINLP solution techniques. The benefits of the lifted polyhedral approximation have, for example, been discussed in [5, 6] and Hijazi et al. demonstrated in [3] the benefits of combining lifted polyhedral approximations with outer approximation. Here the main interest has been to study the benefits of combining lifted polyhedral approximations with a new solver called the supporting hyper plane optimization toolkit (SHOT) [4]. As will be demonstrated in this paper, the reformulation is beneficial for solution techniques based on a polyhedral outer approximation of the feasible region. Solvers utilizing a polyhedral approximation are for example SHOT, ALPHAECP and DICOPT, as well as BONMIN with certain settings, *e.g.*, see [4, 1].

In a recent benchmark on convex MINLP instances presented in [4] the previously mentioned solvers preformed well and the overall best performance was obtained with SHOT closely followed by BARON and ALPHAECP, thus motivating the use of solvers based on polyhedral approximations. However, even though there are several commercial and academic solvers available for convex MINLP, it should be noted that convex MINLP is still a challenging type of optimization problems. For example in the benchmark about 3% of the problem instances remains unsolved, even if we combine the effort of all the solvers used in the benchmark. Hence further research is still motivated within the field of convex MINLP.

2. Divided linear approximations

Here we will consider the linear approximation of completely additively separable and almost separable functions. We will illustrate the benefits of approximating each component separately compared to a linear approximation of the original function by two simple examples. A completely additively separable function f can be decomposed into univariate functions g_i according to

$$f(x_1, x_2, ..., x_n) = \sum_{i=1}^{n} g_i(x_i). \tag{1}$$

In case the function f is convex, it is clear that all the functions g_i are also convex. A partially additively separable function, sometimes also referred to as additively decomposed, cannot be divided into univariate functions. However, it can be divided into functions h_i consisting of only a subset of the original variables. Let N be an index set containing the index of each variable, *i.e.*, $N = \{1, 2, ..., n\}$ and let $N_i \subset N$ be the index set of the variables defining h_i. Then a partially additively separable function can be defined as

$$f(x_1, x_2, ..., x_n) = \sum_{i=1}^{n} h_i\left(x_j \mid j \in N_i\right). \tag{2}$$

The definition of a partially additively separable function require the subsets N_i to be disjoint sets. However, here we only require that the sets are not equal and that they are proper subsets of N. These functions will here be referred to as almost separable. Now it should be noted that a variable *e.g.*, x_1 may appear in several of the functions h_i. Thus, even if the original function f is convex it does not guarantee that the separated functions h_i are convex. Here we will only consider such functions f that can be divided into convex functions h_i. Now, consider how to generate a polyhedral approximation of the convex function f. The intuitive way is to utilize a first order Taylor series expansion at the linearization point $\mathbf{x}^0$ according to

$$f(\mathbf{x}) \approx \hat{f}(\mathbf{x}) = f(\mathbf{x}^0) + \nabla f(\mathbf{x}^0)^T(\mathbf{x} - \mathbf{x}^0). \tag{3}$$

If we utilize several linearizations the approximation is given by

$$f(\mathbf{x}) \approx \hat{f}_1(\mathbf{x}) = \max_j \left\{ f(\mathbf{x}^j) + \nabla f(\mathbf{x}^j)^T(\mathbf{x} - \mathbf{x}^j) \right\}. \tag{4}$$

However, it is also possible to generate individual linearizations for each component h_i of the function f

$$f(\mathbf{x}) \approx \hat{f}_2(\mathbf{x}) = \sum_{i=1}^{n} \max_j \left\{ h_i(\mathbf{y}_i^j) + \nabla h_i(\mathbf{y}_i^j)^T(\mathbf{y}_i - \mathbf{y}_i^j) \right\}, \tag{5}$$

where $\mathbf{y}_i$ is a vector, containing a subset of the variables $\mathbf{x}$, defined by the index set N_i. Provided that each component h_i is a convex function, the approximation given by $\hat{f}_2$ will always underestimate the function f.

Now, to illustrate the benefits of the latter approximation let ut consider a simple example. Suppose we wish to approximate the function $f(x, y) = x^2 + y^2$ within the box defined by $-3 \leq x \leq 3$ and $-3 \leq y \leq 3$. We then generate the approximations $\hat{f}_1$ and $\hat{f}_2$ according to equations (4) and (5) by generating linearizations at the points $(-3, 0), (0, 0), (3, 0), (0, -3)$ and $(0, 3)$. From Figure 1 it is clear that $\hat{f}_2$ gives a tighter approximation of f compared to $\hat{f}_1$. From the figure it can also be seen that the greatest approximation error of $\hat{f}_1$ is obtained at the corners of the box. At these corner points $\hat{f}_2$ actually gives an exact approximation. This is due to the fact that the individual components $h_1(x) = x^2$ and $h_2(y) = y^2$ are approximated by linearizations at the extreme values and at the center value. Hence, each component have an

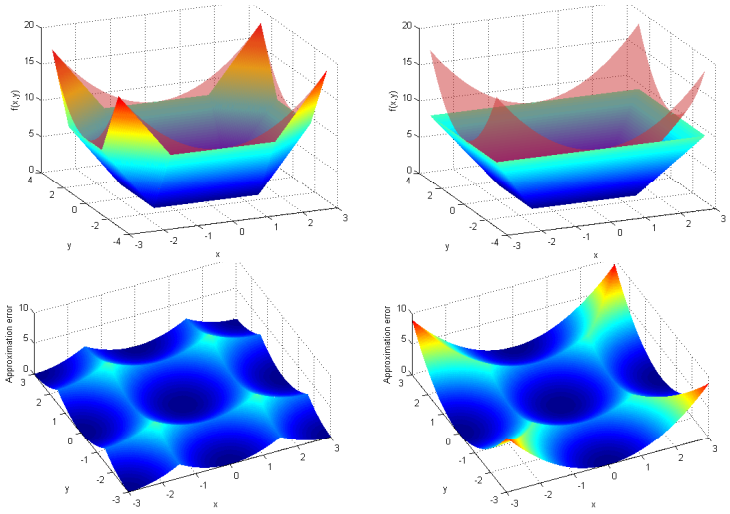


Figure 1. The figures on top shows the approximations and in light red the real function f. The upper left figure shows the approximation according to equation (5) and the upper right figure shows the approximation according to equation (4). The figures below shows the corresponding approximation error.

exact approximation at the extreme values and at the center resulting in an exact approximation at each combination of these points. For an optimization method based on a polyhedral approximation, the tighter approximation gives a great advantage, usually reducing the number of iterations required significantly.

In the previous example the function f was completely additively separable. Now let us consider the almost separable function $f(x,y,z) = x^{-1.2}y^{-0.8} + e^{0.5(y+z)}$. Suppose we wish the approximate the function f within the three dimensional box X, defined by $X = \{x,y,z \mid 0.1 \leq x \leq 3, 0.1 \leq y \leq 3, 0.1 \leq z \leq 3\}$. Here both components $h_1(x,y) = x^{-1.2}y^{-0.8}$ and $h_2(y,z) = e^{0.5(y+z)}$ are convex, thus the approximation according to equation (5) will result in an underestimation of f. Suppose we generate the approximations $\hat{f}_1$ and $\hat{f}_2$ by generating linearizations at all corners of the box X. To compare the approximations we can integrate the difference between the approximation a $\hat{f}_k$ and f to obtain an estimation error volume V_k. The estimation error volumes are given by

$$V_1 = \int_{0.1}^{3}\int_{0.1}^{3}\int_{0.1}^{3} \left(f(x,y,z) - \hat{f}_1(x,y,z)\right) dxdydz$$
$$V_2 = \int_{0.1}^{3}\int_{0.1}^{3}\int_{0.1}^{3} \left(f(x,y,z) - \hat{f}_2(x,y,z)\right) dxdydz.$$

In case linearizations are added for all corners of the box X, we obtain $V_1 \approx 37.9$ and $V_2 \approx 34.7$. More interestingly, in case we leave out linearizations at 3 cornes of X we obtain $V_1 \approx 216.2$ and still $V_2 \approx 34.7$. Hence, in this case the approximation $\hat{f}_2$ results in a tighter approximation even with fewer linearizations points.

3. Lifted polyhedral approximations in MINLP

In the previous section we showed that the approximation $\hat{f}_2$ can give a significantly tighter underestimator than $\hat{f}_1$. By introducing auxiliary variables μ_i in a convex MINLP instance, the problem can be written in such a form that solvers based on a polyhedral approximation can benefit from the separate approximation of the components h_i as in equation (5). This is

Table 1. Comparison of solution times of original problem and the reformulated with different solvers.

MINLP Instance	*AlphaECP* *Org./ Ref.*	*BONMIN OA* *Org./ Ref.*	*DICOPT* *Org./ Ref.*	*SHOT* *Org./ Ref.*
Netmod_dol1	>7200 s/ 3794 s	>7200 s/ 1135 s	>7200 s/ 3462 s	>7200 s/ 1199 s
Portfolio_classical_050_1	>7200 s/ 6 s **	>7200 s/ 4 s	>7200 s/ 8 s	2513 s/ 15 s
Slay10h	>7200 s/ 93 s	>7200 s/ 15s	>7200 s/ 51 s	1649 s/ 22 s
Stockcycle	>7200 s/ 1 s	>7200 s/ 1 s	>7200 s/ 1 s	>7200 s/ 3 s
Tls4	142 s/ 26 s	59 s/ 3 s	>7200 s/ >7200 s	52 s/ 9 s
Tls5	>7200 s/ >7200 s	>7200 s/ >7200 s	>7200 s/ >7200 s	>7200 s/ 4403 s
Tls7	>7200 s/ >7200 s	>7200 s/ >7200 s	7200 s/ >7200 s	>7200 s/ >7200 s

The problem instances were solved to a desired tolerance of $\pm 1\%$ of the optimal solution, which in practice was specified with the optcr option in GAMS. All calculations where done on a desktop computer with an quad-core Intel Xeon 3.6 GHz processor. In the instance marked ** AlphaECP did not return a solution within the desired tolerance. This is due to the fact that the solver's termination criterion is not based on the duality gap. All solvers have used CPLEX as subsolver.

done by dividing each constraint $f(\mathbf{x}) = \sum_{i=1}^{n} h_i\left(x_j \mid j \in N_i\right) \leq 0$ according to

$$h_i\left(x_i \in N_i\right) \leq \mu_i \quad \forall i, \quad \sum_{i=1}^{n} \mu_i \leq 0. \tag{6}$$

A solution satisfying these new constraints, will obviously satisfy the original constraint. Note that this transformation requires each component h_i to be convex, which does not follow from the convexity of f. Next we will consider 7 convex MINLP instances taken from MINLPLib2 [2], where some of the nonlinear functions are completely or almost additively separable. We have solved the problems both in their original form and with the transformation according to equation (6). SHOT, as well as the GAMS solvers AlphaECP, BONMIN-OA and DICOPT were used in the comparison, whom all utilize a polyhedral approximation and should hence benefit from the transformation. For all solvers we have used the parameters recommended for convex problems and the time limit was set to 7200 s. From Table 1 it can be seen that the reformulation, resulting in a lifted polyhedral approximation, in several cases drastically improved the performance of the solvers. The results strongly motivates the use of a lifted polyhedral approximation, and it will be investigated further how to incorporate it in SHOT.

Acknowledgments

The authors wish to thank GAMS Development Corporation for support. J.K. is also grateful for the financial support from the Finnish Graduate School in Chemical Engineering.

References

[1] M.R. Bussieck and S. Vigerske. MINLP solver software. In J.C. Cochran, L.A. Jr. Cox, P. Keskinocak, L.A. Kharoufeh, and J.C. Smith, editors, *Wiley Encyclopedia of Operations Research and Management Science*. Wiley Online Library, 2010.

[2] GamsWorld. Mixed-integer nonlinear programming library, 2016. [Online; accessed 1-February-2016].

[3] H. Hijazi, P. Bonami, and A. Ouorou. An outer-inner approximation for separable MINLPs. *LIF, Faculté des Sciences de Luminy, Université de Marseille, Tech. Rep*, 2010.

[4] J. Kronqvist, A. Lundell, and T. Westerlund. The extended supporting hyperplane algorithm for convex mixed-integer nonlinear programming. *Journal of Global Optimization*, 64(2):249–272, 2016.

[5] M. Lubin, E. Yamangil, R. Bent, and J.P. Vielma. Extended formulations in mixed-integer convex programming. *arXiv preprint arXiv:1511.06710*, 2015.

[6] M. Tawarmalani and N.V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103(2):225–249, 2005.

Proceedings of GOW'16, pp. 121 – 124.

Preference-based Multi-Objective Single Agent Stochastic Search

Algirdas Lančinskas and Julius Žilinskas

Vilnius University, Vilnius, Lithuania algirdas.lancinskas@mii.vu.lt, julius.zilinskas@mii.vu.lt

Abstract The research is focused on preference-based multi-objective optimization. The algorithm based on controlled random search which includes preference information given by the decision maker is proposed and experimentally investigated thus highlighting its advantages.

Keywords: Multi-objective optimization, Random search, Preference information

1. Introduction

A global optimization problem can be described as

$$\min_{\mathbf{x}\in D} f(\mathbf{x}), \tag{1}$$

where $f(\mathbf{x})$ is the *objective function*, $\mathbb{D}$ is the *search space*, and $\mathbf{x}$ is the *decision vector*. Real-world optimization problems usually require to consider more than one criteria thus translating the above global optimization problem to the *multi-objective optimization problem* where a set of $m \geq 2$ objective functions $f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})$ must be taken into account when determining the optimal decision vector of the problem.

Due to conflicts between objectives often encountered in real-world situations, usually it is impossible to determine a single decision vector $\mathbf{x}^*$, which would be the best according to all objectives of the problem – the best decision vector by one objective can be worse or even the worst by another one. Therefore the goal of solution of a multi-objective optimization problem is to find a set of compromising decision vectors which cannot be improved by any objective without deterioration of another one. Such a set of decision vectors is called *Pareto Set* and the corresponding set of the objective functions values – the *objective vectors* – is called *Pareto Front*. For more details on multi-objective optimization and Pareto optimality we refer to [1].

Determination of the Pareto front is the main goal of multi-objective optimization. However, determination of the exact Pareto front of a real-world problem usually is complex and time consuming task and sometimes impossible within a reasonable time. On the other hand, usually it is not necessary to find the exact Pareto front when dealing with real-world optimization problems – it is good enough to provide a discrete approximation of the true Pareto front.

The algorithms based on controlled random search are popular to tackle practical optimization problems as they usually are easy to implement, robust, and require a little knowledge on the problem being solved – the only requirement is to be able to evaluate objective values for every feasible decision vector. The well-known algorithms such as NSGA-II [2], SPEA2 [10], MOEA/D [5, 7], NSGA-III [3] etc. can be given as an examples of controlled random search algorithms for approximation of the Pareto front.

The set of objective vectors approximating the Pareto front can be very large (or even infinite) and hard to process especially if a decision maker have to select one or several the most suitable solutions. On the other hand the decision maker might declare a *region of interest* of the Pareto front and guide the search towards it. Based on whether the preference information

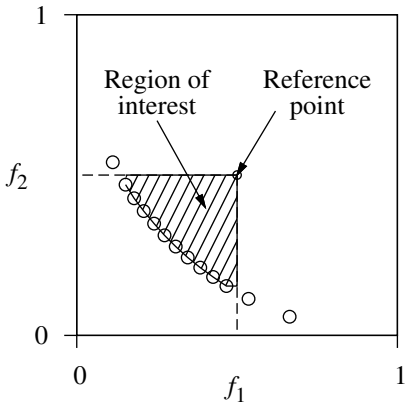


Figure 1. Illustration of the region of interest defined by the reference point for the problem of two objectives both subject to minimize

is provided before, after or during the optimization process, preference-based multi-objective optimization methods can be classified into priori, posteriori, and interactive ones [8].

Our research is focused on incorporation of a decision maker preferences into previously developed Multi-Objective Single Agent Stochastic Search (MOSASS) thus developing Preference-based MOSASS – PreMOSASS.

2. PreMOSASS

MOSASS has been developed from Single Agent Stochastic Search (SASS) [6] and used in conjunction with NSGA-II in [4].

MOSASS is based on generation of new decision vectors in the neighborhood of the initial decision vector, which is updated if the decision vector dominating it is found. The perturbation in random generation of coordinates of the new decision vector is dynamically adjusted according to the repetitive successes and failures in generation the new decision vectors, thus giving more attention to the global or the local search. For the detailed description of MOSASS we refer to [4].

PreMOSASS extends the MOSASS by incorporating the reference point as the preference information given by decision maker. Thus the PreMOSASS focuses on the approximation of the Pareto front within the region of interest (see Figure 1).

The PreMOSASS algorithm begins with an initial decision vector $\mathbf{x} = (x_1, x_2, \dots x_n)$ and an empty archive A for storing non-dominated decision vectors will be found at runtime of the algorithm. A new decision vector $\mathbf{x}'$ is generated by changing values of some variables of $\mathbf{x}$. Each variable x_i ($i = 1, 2, \dots, n$) is modified with probability $1/n$ by adding a random value ξ_i generated following Gaussian distribution. In general, the new decision vector can be expressed mathematically as

$$\mathbf{x}' = \mathbf{x} + \xi, \tag{2}$$

where $\xi = (\xi_1, \xi_2, \dots, \xi_n)$ and

$$\xi_i = \begin{cases} \mathcal{N}(b_i, \sigma), & \text{if} \quad r_i \leq 1/n, \\ 0, & \text{if} \quad r_i > 1/n, \end{cases} \tag{3}$$

where $\mathcal{N}(b_i, \sigma)$ is random number generated following Gaussian distribution with the bias b_i and the standard deviation σ, r_i – a random number uniformly generated within interval $[0, 1]$, and n – the number of problem variables. Such a probabilistic approach of generation of

new decision vector leads to the change of a single variable in average; see [4] for details and advantage of the approach.

If the generated decision vector $\mathbf{x}'$ dominates its precursor $\mathbf{x}$, then A is updated by removing all decision vectors dominated by $\mathbf{x}'$, and $\mathbf{x}$ is replaced by $\mathbf{x}'$. The iteration is then assumed to be successful and the algorithm continues to the next iteration.

If $\mathbf{x}'$ does not dominates its precursor $\mathbf{x}$ and is not dominated by it, then the dominance relation of $\mathbf{x}'$ against reference point and decision vectors in A is evaluated. If $\mathbf{x}'$ dominates the reference point (belongs to the region of interest) and is not dominated by any decision vector in A, then A is updated by removing all decision vectors dominated by $\mathbf{x}'$ and including $\mathbf{x}'$ as a member of the archive. Otherwise, if $\mathbf{x}'$ is dominated by the reference point or by any decision vector from A, then $\mathbf{x}'$ is rejected and the opposite decision vector $\mathbf{x}'' = \mathbf{x} - \xi$ is investigated.

If decision vector $\mathbf{x}$ is updated or archive A is supplemented by $\mathbf{x}'$, then iteration is assumed to be successful and the biases of Gaussian perturbation is updated by

$$b_i \leftarrow 0.2b_i + 0.4\xi_i, \quad i = 1, 2, \ldots, n. \tag{4}$$

If decision vector $\mathbf{x}$ is updated or archive A is supplemented by $\mathbf{x}''$, then iteration is assumed to be successful and the biases of Gaussian perturbation is updated by

$$b_i \leftarrow b_i - 0.4\xi_i, \quad i = 1, 2, \ldots, n. \tag{5}$$

If both $\mathbf{x}'$ and $\mathbf{x}''$ are rejected, then iteration is assumed to be failed and the biases of Gaussian perturbation is updated by

$$b_i \leftarrow 0.5b_i, \quad i = 1, 2, \ldots, n. \tag{6}$$

The standard deviation σ of the Gaussian perturbation is dynamically adjusted with respect to the repetitive success and failed iterations. If the number $scnt$ of repetitive successful iterations reaches the predefined number $Scnt$, then the standard deviation is increased twice. Analogous, if the number $fcnt$ of repetitive failed iterations reaches the predefined number $Fcnt$, then the standard deviation is reduced by a half. If the standard deviation becomes smaller than its lower bound σ_{min}, then σ is set to its lower bound ($\sigma = \sigma_{min}$).

The iterative process is continued till a stopping criterion is not satisfied, which is usually based on the maximum number of iterations or the maximum number of functions evaluations. The algorithm returns a set $A \cup \{\mathbf{x}\}$ of non-dominated decision vectors found during the runtime of the algorithm.

3. Computational experiments

PreMOSASS has been experimentally investigated by solving multi-objective optimization problem ZDT1 [9] with 10 variables and 2 objectives – both subject to minimize. The reference point has been set to $(0.5, 0.5)$ and 1000 function evaluations have been used for the approximation of the Pareto front.

The results obtained by MOSASS and PreMOSASS are illustrated in Figure 2. One can see from the figure that approximation of the Pareto front obtained by MOSASS covers the whole Pareto front, whereas PreMOSASS focuses on exploration of the regenio defined by the reference point.

4. Summary

The previously proposed algorithm MOSASS has been improved by incorporating the preference information thus developing the preference-based multi-objective optimization algorithm PreMOSASS. The proposed algorithm has been experimentally investigated by solving

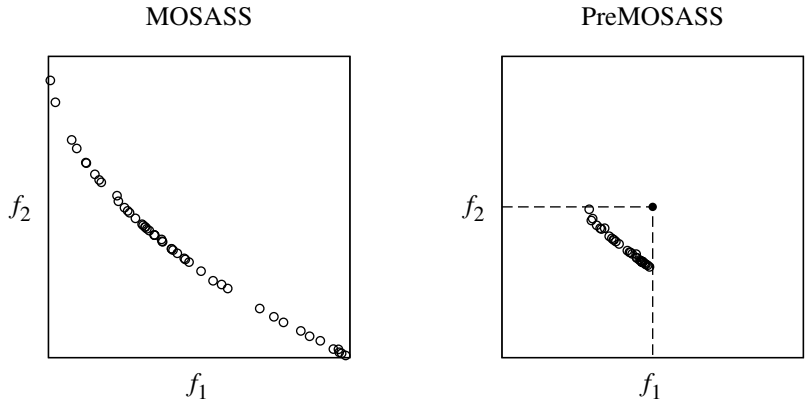


Figure 2. Pareto front approximation obtained by MOSASS and by PreMOSASS with reference point $(0.5, 0.5)$.

a well-known multi-objective optimization test problem thus highlighting advantages of incorporation of the preference information into the random search technique.

References

[1] Carlos Coello Coello, Gary B Lamont, and David A. van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation. Springer, Berlin, Heidelberg, 2nd edition, 2007.

[2] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[3] Himanshu Jain and Kalyanmoy Deb. An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part II: Handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation*, 18(4):602–622, 2014.

[4] Algirdas Lančinskas, Pilar Martinez Ortigosa, and Julius Žilinskas. Multi-objective single agent stochastic search in non-dominated sorting genetic algorithm. *Nonlinear Analysis: Modelling and Control*, 18(3):293–313, 2013.

[5] Tapabrata Ray, Md Asafuddoula, and Amitay Isaacs. A steady state decomposition based quantum genetic algorithm for many objective optimization. In *2013 IEEE Congress on Evolutionary Computation (CEC)*, pages 2817–2824, 2013.

[6] F.J. Solis and B. Wets. Minimization by random search techniques. *Mathematics of Operations Research*, 6(1):19–30, 1981.

[7] Qingfu Zhang and Hui Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

[8] Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Qingfu Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, 2011.

[9] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.

[10] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm. Technical report, 2001.

Proceedings of GOW'16, pp. 125 – 128.

A Quadratic Approach to the Maximum Edge Weight Clique Problem*

Seyedmohammadhossein Hosseinian[1], Dalila B.M.M. Fontes[1,2] and Sergiy Butenko[1]

[1]*Texas A&M University, Department of Industrial and Systems Engineering, College Station, TX 77843, United States*
hosseinian@tamu.edu, dfontes@tamu.edu, butenko@tamu.edu

[2]*Faculdade de Economia and LIAAD - INESC TEC, Universidade do Porto, Rua Dr. Roberto Frias, 4200 Porto, Portugal*

Abstract This work addresses the maximum edge weight clique problem, a generalization of the well-known maximum clique problem. We propose a heuristic approach based on the optimization of a quadratic function over a sphere. Preliminary computational results are reported for a set of benchmark problem instances derived from the DIMACS maximum clique instances.

Keywords: Maximum weighted clique, Quadratic optimization, Heuristics

1. Introduction

The maximum weight clique problem is an important generalization of the maximum clique problem (MC) and it has been considered in three different versions in the literature: the maximum vertex weight clique, the maximum edge weight clique, and the maximum edge and vertex weight clique. In all three versions a clique with maximum weight is sought. The difference being the way in which the clique weight is calculated, namely as the sum of the weights of its vertices, the sum of the weights of its edges, and the sum of the weights of both its vertices and edges, respectively. In this work, we focus on the maximum edge weight clique (MEWC) problem. If all edge weights are equal, then finding the MEWC is equivalent to finding the MC. Therefore, the MEWC problem has, at least, the same computational complexity as the MC problem, which is known to be $\mathcal{NP}$-hard [6].

Many applications can be found for these cliques in facility location and dispersion [11], molecular biology [5], broadband network design [8], and bioinformatics [3]. The first work on this problem appears to have been that of Späth [11]. The author addresses a facility selection-location problem and finds heuristic solutions through iterative improvements of some initial clique by successively exchanging two vertices. Since then several works have been reported in the literature. Exact branch-and-cut algorithms, based on linear programming have been proposed by [7, 8, 10], among others. These works first introduce facet-defining inequalities which are then used as cutting planes. More recently, heuristics have been proposed: Tabu Search [1], phased local search [9], and several linear programming relaxations [4].

All authors, except for [4, 9], consider complete networks, although many practical applications are defined on sparse graphs, for example in protein interaction networks [3]. Furthermore, except for [1] all the works on the MEWC use linear representations of the problem. This preference for linearity is understandable, since many methods that take advantage of linearity have been proven to be successful and efficient. However, linearizing combinatorial problems may not always be the best choice from a computational point of view.

In this paper, we propose a heuristic that uses a continuous quadratic formulation to find solutions to the MEWC problem. Section 2 provides a quadratic discrete formulation, as well

*The financial support of project UID/EEA/50014/2013 and NSF/CMMI–1538493 is appreciated.

as an equivalent quadratic continuous formulation. The heuristic proposed in this work is presented in Section 3, which also reports on preliminary computational results. Finally, Section 4 concludes this work.

2. Problem definition and formulation

Given a simple edge-weighted graph $G = (V, E)$, where V is the set of vertices and E is the set of edges, a clique in G is a subset of vertices $C \subseteq V$, such that all vertices in C are pairwise adjacent. The weight of a clique C is defined as $W(C) = \sum_{(i,j)\in E(C)} w_{ij}$, where $w_{ij} > 0$ denotes the weight associated with edge (i, j) and $E(C)$ is the set of edges connecting vertices of C. The MEWC problem seeks a clique C in G which maximizes $W(C)$.

A straightforward binary quadratic formulation for the MEWC problem is:

$$\max \left\{ \sum_{i,j\in V} w_{ij}x_ix_j : x_ix_j = 0 \,\forall\, (i,j) \notin E; x_v \in \{0,1\} \,\forall\, v \in V \right\}. \tag{1}$$

The following result can be derived from (1) :

Proposition 1. *The MEWC problem can be equivalently formulated as a continuous quadratic problem, as shown in (2), where M denotes a sufficiently big number.*

$$\max_{x_v\in[0,1]\,\forall\, v\in V} \left(\sum_{(i,j)\in E} w_{ij}x_ix_j - \sum_{(i,j)\notin E} Mx_ix_j \right). \tag{2}$$

3. Heuristic approach

Problem (2) is $\mathcal{NP}$-hard. However, optimizing a quadratic function subject to ellipsoid constraints has been proved to be polynomially solvable [12]. In this work, following the work in [2] the MEWC problem is approximated by substituting the unit hypercube constraint by a unit hypersphere.

Let A_G denote the weighted adjacency matrix of graph G, where $A_G(i, j)$ is set to w_{ij} for $(i, j) \in E$ and 0 otherwise, and $\bar{A}_G$ the adjacency matrix of $\bar{G}$, the complement of G, where $\bar{A}_G(i, j)$ is set to 1 for $(i, j) \notin E$ and to 0 otherwise. The quadratic model in (2) can then be written as:

$$\max_{X\in[0,1]^n} \left(\frac{1}{2}X^TA_GX - \frac{1}{2}X^T(M\bar{A}_G)X \right), \tag{3}$$

which is approximated by:

$$\begin{aligned} &\text{Maximize } \frac{1}{2}X^TQX \\ &\text{Subject to} \\ &\qquad ||X||_2^2 = 1, \end{aligned} \tag{4}$$

where $X \in \mathbb{R}^n$ is the vector of variables, $Q = A_G - M\bar{A}_G$, and $||X||_2^2 = X^TX$.
The stationary points of this problem are given by the eigenpairs (μ^*, X^*) of Q. Thus, a heuristic solution to the MEWC can be obtained by enumerating the stationary points, i.e., eigenvectors of Q, and extracting the corresponding clique in G, as in the algorithm below.

ALGORITHM

1. **for** each eigenvector X^e of matrix $Q = A_G - M\bar{A}_G$ **do**
   (a) obtain a clique C by performing the following procedure:
      i. Sort components of X^e in a nonincreasing order (i.e. $x_1^e \geq x_2^e \geq \cdots \geq x_n^e$)
      ii. $C \leftarrow v(x_1^e)$
      iii. **for** $j = 2$ to n **do**
         if $v(x_j^e) \in \bigcap_{i \in C} N(i)$ **then** $C \leftarrow C \cup \{v(x_j^e)\}$
   (b) obtain $W(C)$ the weight of the incumbent clique
   (c) **if** $W(C) > W(C^*)$ **then** $C^* \leftarrow C$
2. Return the best known clique C^* and the corresponding weight $W(C^*)$

Here, $v(x)$ denotes the vertex corresponding to variable x and $N(i)$ denotes the set of vertices adjacent to vertex i in G (i.e. neighbors of u).
Preliminary results are reported in Table 1 for the DIMACS-EW benchmark instances, proposed by Pullan [9], which were derived from the DIMACS MC instances.

The first four columns characterize the problem instances in terms of name, number of vertices, number of edges, and edge density, while the fifth reports the best known solution due to Pullan [9]. In columns six and seven we report the clique weight $W(C)$ and clique size $|C|$, obtained by the heuristic proposed here, and in column eight the computational time required to obtain such solutions. Finally, column nine reports on the quality of solutions by providing the percentage gap between our solutions and those of Pullan, i.e. $\frac{W'(C)-W(C)}{W'(C)} \times 100$.

Table 1. Computational results for the proposed algorithm.

Instances	$\|V\|$	$\|E\|$	$d(G)$	$W'(C)$	$W(C)$	$\|C\|$	*Time(s)*	Gap(%)
johnson8-2-4	28	210	0.556	192	192	4	0.01	0.00
MANN-a9	45	918	0.927	5,460	5,445	16	0.04	0.27
hamming6-2	64	1,824	0.905	32,736	32,736	32	0.13	0.00
hamming6-4	64	704	0.349	396	396	4	0.08	0.00
johnson8-4-4	70	1,855	0.768	6,552	6,552	14	0.11	0.00
johnson16-2-4	120	5,460	0.765	3,808	3,766	8	0.46	1.10
C125.9	125	6,963	0.898	66,248	60,095	31	0.54	9.29
keller4	171	9,435	0.649	6,745	6,175	11	1.29	8.45
brock200-1	200	14,834	0.754	21,230	21,230	21	2.35	0.00
brock200-2	200	9,876	0.496	6,542	6,542	12	2.02	0.00
brock200-3	200	12,048	0.605	10,303	10,303	15	2.03	0.00
brock200-4	200	13,089	0.658	13,967	13,736	17	2.05	1.65
c-fat200-1	200	1,534	0.077	7,734	7,734	12	2.65	0.00
c-fat200-2	200	3,235	0.163	26,389	26,389	23	2.14	0.00
c-fat200-5	200	8,473	0.426	168,200	168,200	58	2.76	0.00

As it can be seen, for most problem instances the best known result was matched, nonetheless this was not the case for five problem instances. For two problem instances the optimality gap is unexpectedly large.

4. Conclusions

This work introduces a heuristic for the MEWC problem that approximates the optimal solution by maximizing a continuous quadratic program over an unit hypersphere centered at the

origin rather than over the unit hypercube. The preliminary results obtained are very encouraging as for many of the problem instances we were able to obtain a solution as good as the best known solution. Other approximations and corresponding heuristics can be derived by considering spheres centered at points other than the origin and different radius r, that is by generalizing the sphere equation to $||X - X^0||_2^2 = r^2$.

References

[1] Bahram Alidaee, Fred Glover, Gary Kochenberger, and Haibo Wang. Solving the maximum edge weight clique problem via unconstrained quadratic programming. *European Journal of Operational Research*, 181(2):592–597, 2007.

[2] Stanislav Busygin, Sergiy Butenko, and Panos M Pardalos. A heuristic for the maximum independent set problem based on optimization of a quadratic over a sphere. *Journal of Combinatorial Optimization*, 6(3):287–297, 2002.

[3] Sergiy Butenko and Wilbert E Wilhelm. Clique-detection models in computational biochemistry and genomics. *European Journal of Operational Research*, 173(1):1–17, 2006.

[4] Luis Gouveia and Pedro Martins. Solving the maximum edge-weight clique problem in sparse graphs with compact formulations. *EURO Journal on Computational Optimization*, 3(1):1–30, 2015.

[5] Marcel Hunting, Ulrich Faigle, and Walter Kern. A lagrangian relaxation approach to the edge-weighted clique problem. *European Journal of Operational Research*, 131(1):119–131, 2001.

[6] Richard M Karp. *Reducibility among combinatorial problems*. Springer, 1972.

[7] Elder Magalhães Macambira and Cid Carvalho de Souza. The edge-weighted clique problem: valid inequalities, facets and polyhedral computations. *European Journal of Operational Research*, 123:346–371, 2000.

[8] Kyungchul Park, Kyungsik Lee, and Sungsoo Park. An extended formulation approach to the edge-weighted maximal clique problem. *European Journal of Operational Research*, 95(3):671–682, 1996.

[9] Wayne Pullan. Approximating the maximum vertex/edge weighted clique using local search. *Journal of Heuristics*, 14(2):117–134, 2008.

[10] Michael M. Sørensen. New facets and a branch-and-cut algorithm for the weighted clique problem. *European Journal of Operational Research*, 154:57–70, 2004.

[11] H Späth. Heuristically determining cliques of given cardinality and with minimal cost within weighted complete graphs. *Zeitschrift für Operations Research*, 29(3):125–131, 1985.

[12] Yinyu Ye. A new complexity result on minimization of a quadratic function with a sphere constraint. In *Recent Advances in Global Optimization*, pages 19–31. Princeton University Press, 1992.

Proceedings of GOW'16, pp. 129 – 132.

Runtime Landscape Analysis for Global Optimization using Memetic Approaches

Federico Cabassi[1] and Marco Locatelli[2]

[1]*Dipartimento di Ingegneria dell'Informazione, Università di Parma, Parma, Italy,* federico.cabassi@studenti.unipr.it

[2]*Dipartimento di Ingegneria dell'Informazione, Università di Parma, Parma, Italy,* marco.locatelli@unipr.it

Abstract The aim of this paper is threefold. First, given the objective function of a global optimization problem, we propose techniques to identify some of its properties, on which the difficulty of the optimization depends. Second, we propose a new class of test functions, which allows to modify the difficulty of the global optimization task by varying a couple of parameters and which can be employed, together with other already existing test functions, to evaluate the previously mentioned techniques. Third, we show how the outcome of our analysis can be employed to make appropriate algorithmic choices for the global optimization task through memetic approaches. The discussion will put in evidence the advantages but also the limits of the proposed techniques.

Keywords: Global optimization, Funnel Landscapes, Memetic Algorithms

1. Introduction

The global optimization task asks for the minimization of a continuous function f over a compact domain $D \subset \mathbb{R}^n$. Depending on the function to be optimized, the problem of finding the global optimum of the objective function can be either relatively simple, *e.g.*, when f is a convex function and D a convex domain, or very hard, *e.g.*, when f is a highly multimodal function. In the existing literature various techniques have been proposed to solve global optimization problems. For highly structured problems with a moderate dimension it is possible to apply branch-and-bound methods which return a certified globally optimal solution. But for poorly structured problems exact methods are doomed to failure and only heuristic methods can return an estimate of the global optimum within a reasonable time. An exhaustive discussion about different global optimization approaches can be found in [5]. Our focus in this paper is on problems which feasible region is a box or a polytope, and which objective function is highly multimodal. We also assume that function evaluation and local searches are relatively cheap tasks. As already commented in [4], the difficulty of these problems is not only connected to the modality of the functions, *i.e.*, to the number of local minima, but also to the way these minima are placed within the feasible domain. In order to formalize this aspect, the concept of *level 2 local minimum* has been introduced in [4] (*level 1 local minima* are the usual local minima of the function). These are the local minima over a graph which nodes are the level 1 local minima (assumed to be in a finite number). The set of nodes will be denoted by V in what follows. The edges of the graph depend on some neighborhood structure. Edges are always oriented from a local minimum with higher function value towards a local minimum with lower function value. Level 2 local minima always include the global minimum, which will be assumed to be unique. But other level 2 local minima may be present, depending on the choice of the neighborhood. One possible way to define the neighborhood is through *mountain passes* (*i.e.*, stationary points at which the Hessian has a single negative eigenvalue). Given two local minima $\mathbf{x}$ and $\mathbf{y}$, these are connected through an edge if there exists a mountain pass $\mathbf{z}$ and a continuous curve $\mathbf{s}(t)$, $t \in [0,1]$, such that: $\mathbf{s}(0) = \mathbf{x}$, $\mathbf{s}(1) = \mathbf{y}$, $\mathbf{s}(\bar{t}) = \mathbf{z}$, for some $\bar{t} \in [0,1]$, and, moreover, $f(\mathbf{s}(t))$ increases over $[0, \bar{t}]$ and decreases over $[\bar{t}, 1]$.

While rigorous, this definition has the difficulty that the existence of the curve $\mathbf{s}(t)$ is hard to establish. An alternative, simpler, definition is based on the Euclidean distance between local minima. In particular, we use the *normalized* Euclidean distance, *i.e.*, the ratio between the Euclidean distance and the diameter of the feasible domain. In what follows we denote such distance with $D(\cdot,\cdot)$. Normalized distances range between 0 and 1. For a given function f and a threshold $d \in [0,1]$, the oriented graph is $G_d = (V, E_d)$, where

$$E_d = \{(\mathbf{x},\mathbf{y}) \,:\, \mathbf{x},\mathbf{y} \in V,\ \mathbf{x} \neq \mathbf{y},\ D(\mathbf{x},\mathbf{y}) \leq d,\ f(\mathbf{y}) \leq f(\mathbf{x})\}.$$

Local minima over graph G_d are all nodes without outgoing edges. We can define a function $WN : [0,1] \to \{1,\dots,|V|\}$ as follows. First we introduce the subset

$$V(d) = \{\mathbf{x} \in V \,:\, \nexists\, \mathbf{y} \in V \,:\, (\mathbf{x},\mathbf{y}) \in E_d\} \subset V,$$

i.e., the subsets of local minima with no outgoing edges (these are the level 2 local minima identified over graph G_d). Next, let $f_{min} = \min_{\mathbf{x}\in V} f(\mathbf{x})$ and $f_{max} = \max_{\mathbf{x}\in V} f(\mathbf{x})$ be the minimum and maximum objective function value at local minima of the problem (in fact, the former is the global minimum value). Finally,

$$WN(d) = \sum_{\mathbf{x}\in V(d)} \left(\frac{f_{max} - f(\mathbf{x})}{f_{max} - f_{min}}\right),$$

i.e., $WN(d)$ is a weighted sum of the level 2 local minima identified over graph G_d, with weights belonging to the interval $[0,1]$ and larger at local minima with lower function value. It holds that $WN(d) \geq 1$ for all $d \in [0,1]$ since the global minimum always belongs to $V(d)$ and has weight equal to one. As d increases, function WN decreases (more precisely, it is a non increasing step function). Graph G_1 is the complete one and $WN(1) = 1$ (recall we are assuming that the global minimum is unique). The value d_f defined as follows

$$d_f = \min\{d \,:\, WN(d) = 1\},$$

is quite significant. This is the smallest value at which the graph has a single local minimum which necessarily corresponds to the global minimum of f. If d_f is small, this means that we can reach the global minimum by exploring small neighborhoods of the local minima, while a large d_f value means that we may need to explore large neighborhoods in order to reach the global minimum. Since the smaller the neighborhood, the more efficient is its exploration, functions with a small d_f value tend to be easier to optimize. These functions are also called in the literature *single-funnel* ones, while functions for which WN decreases slowly to 1 are called *multi-funnel* ones. In fact, some care is needed. *E.g.*, if one or few local (but not global) minima of f are far away from all the other local minima, their presence will cause a large d_f value, while when restricting the attention to low-level local minima, the d_f value would be much lower. In other words, some care is needed when "outliers" are present. Moreover, the value d_f and the function WN may not be enough to establish the difficulty of a global optimization problem. This will be made clearer through some examples later on, showing that the difficulty of optimizing functions with similar information from WN and d_f can be rather different. All the same the information returned by function WN and by value d_f should be taken into account when evaluating the difficulty of a global optimization problem.

At this point we should remark that the exact knowledge of WN and d_f requires the knowledge of all the local minima, which, of course, would make trivial the detection of the global minimum. Thus, what we can do is to make an estimate. Possible estimates will be given in Section 3.

2. Test functions

In the existing literature there are different test functions which can be classified either as single-funnel or as multi-funnel ones. Examples of single-funnel functions are the Rastrigin

[8], Ackley [1], and Levy n.13 [2] functions. Examples of multi-funnel functions are the Schwefel [7] and Lunacek bi-Rastrigin [6] functions. In our study we considered all these functions. More precisely, we considered rotated versions of these functions in order to remove the simplifying feature of separability of the basic versions. In this paper we also propose a new class of test functions, called MF and parameterized with respect to two parameters $\alpha_l \in [-10, 0]$ and $\alpha_r \in [0, 10]$. The MF functions are defined as:

$$MF_{\alpha_l,\alpha_r}(\mathbf{x}) = \sum_{i=1}^{n} \min\{\theta(x_i - \alpha_l)^2, (x_i - \alpha_r)^2 + 1\} - 10(\cos(2\pi(x_i - \alpha_l)) - 1),$$

$$\text{where} \qquad \theta = \frac{\alpha_r^2 + 1}{\alpha_l^2},$$

$$\text{and} \qquad x_i \in [-10, 10],\ x_i^* = \alpha_l,\ i = 1, \ldots, n,\ f(\mathbf{x}^*) = 0,$$

where $\mathbf{x}^*$ denotes the global minimum. For $\alpha_l, \alpha_r \to 0$, the function MF becomes equivalent to the single-funnel Rastrigin function, while the d_f value of these functions is controlled through the difference $\alpha_r - \alpha_l$. The d_f value for these functions is strictly related to $\alpha_r - \alpha_l$. In the one-dimensional case parameter α_l represents the position of the global minimum, while α_r is the position of another level 2 local minimum over graphs G_d (when d is smaller than $\alpha_r - \alpha_l$). Finally, similarly to what we have done with the other test functions, in order to destroy separability, we considered rotated versions of the MF functions, *i.e.*, we considered functions $MF_{\alpha_l,\alpha_r}(\mathbf{Wx})$, minimized over rotated domains $\{\mathbf{x} \in \mathbb{R}^n : \mathbf{Wx} \in [-10, 10]^n\}$, where $\mathbf{W}$ is an orthonormal matrix.

3. How to establish whether a function is single- or multi-funnel

The construction of the graphs G_d requires the knowledge of all the local minima of the function but this is a task at least as difficult as the original global optimization problem. Thus, we propose to consider a subset of local minima denoted by $\overline{L} \subseteq V$ and the subgraph induced by this subset. The resulting graphs are $\overline{G}_d = (\overline{L}, E_d(\overline{L}))$, where $E_d(\overline{L})$ is the set of edges which nodes both belong to the subset $\overline{L}$. In particular, in order to consider a subset of "good" local minima, we used the population members at different generations of a simple memetic approach called MDE (see [3]), and we computed at each generation an estimate $\overline{WN}$ of function WN (the estimates for f_{min} and f_{max} are the minimum and maximum objective function values within the population at the current generation), and an estimate $\bar{d}_f$ of the value d_f. The functions $\overline{WN}$, after five generations, are displayed in Figure 1. All the curves are obtained averaging the values over 10 random trials.

It is clearly seen that functions $\overline{WN}$ decrease much more rapidly for single-funnel functions. Moreover the $\bar{d}_f$ values of the single-funnel functions are much smaller. Function $MF_{-2.5,2.5}$, which is indeed the one with smallest d_f values among the MF functions we considered, lies at the border between single- and multi-funnel functions but it is more clearly distinguished with respect to single-funnel functions like the Levy and Rastrigin ones, and the graph of the corresponding $\overline{WN}$ function lies above that of the three single-funnel functions. Thus, we have seen that it is possible to identify some indicators which allow to recognize whether a function is single- or multi-funnel. Interestingly, it also holds that the three $\overline{WN}$ curves which decrease more slowly (namely, Schwefel, $MF_{-8,5,8.5}$ and $MF_{-2,5,8.5}$) will turn out to be also those of the most challenging tests according to our experiments.

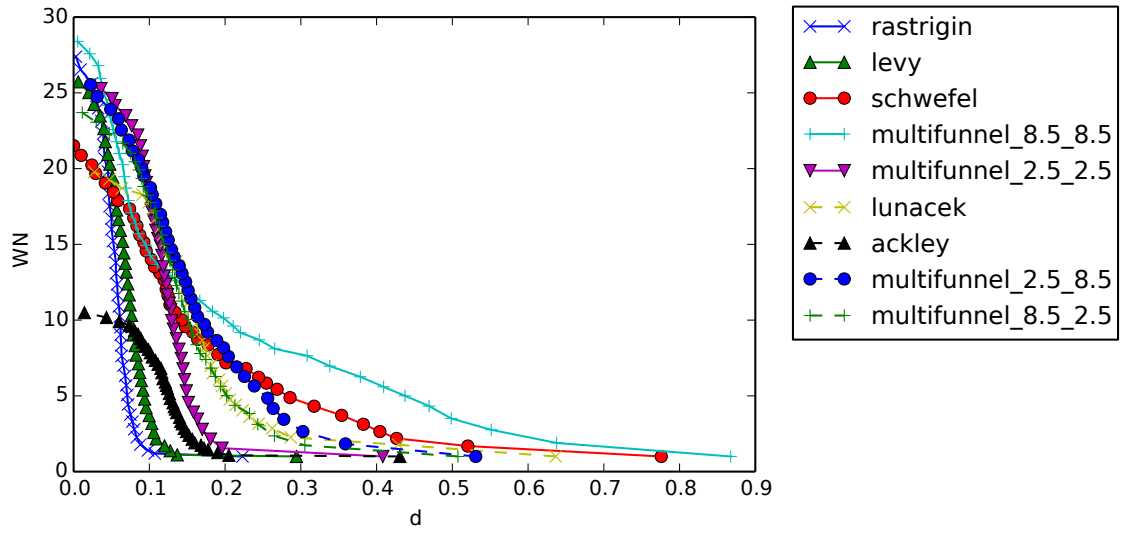


Figure 1. Represents the WN function after the 5-th generation using MDE.

4. Computational experiments

We have performed some preliminary computational experiments which primary goal is to verify that the difficulty indicator estimated in Section 3 is coherent with the real function difficulty. The tests have been executed averaging the values over 25 trials and using different memetic approaches based on Differential Evolution. These approaches can be divided in three categories. The first category uses a *greedy* search policy that allows a fast convergence of the algorithm. The second category try to maintain *diversity* between elements of the population causing a slower convergence (and consequently a larger number of local searches) but a more accurate exploration. The last category has an *intermediate* behaviour between the previous two. The computational experiments have shown that the functions with a low $\bar{d}_f$ value and fast decreasing $\overline{WN}$ function can be efficiently solved using a greedy strategy. However, as the $\bar{d}_f$ value increases, the results obtained with a greedy strategy become poorer in terms of successes. Functions with a high $\bar{d}_f$ value and slowly decreasing $\overline{WN}$ function require approaches which favor global exploration by enhancing diversity within the population.

References

[1] D.H. Ackley. *A Connectionist Machine for Genetic Hillclimbing*. Kluwer Academic Publishers, Boston, 1987.

[2] Alejandro Velasco Levy and A. Montalvo. The tunneling method for global optimization. *SIAM Journal on Scientific and Statistical Computing*, 1:15–29, 1985.

[3] M. Locatelli, M. Maischberger, and F. Schoen. Differential evolution methods based on local searches. *Computers and Operations Research*, 43:169–180, 2014.

[4] Marco Locatelli. On the multilevel structure of global optimization problems. *Computational Optimization and Applications*, 30(1):5–22, 2005.

[5] Marco Locatelli and Fabio Schoen. *Global Optimization: Theory, Algorithms, and Applications*. SIAM, 2013.

[6] Monte Lunacek, Darrell Whitley, and Andrew Sutton. *Parallel Problem Solving from Nature – PPSN X: 10th International Conference, Dortmund, Germany, September 13-17, 2008. Proceedings*, chapter The Impact of Global Structure on Search, pages 498–507. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[7] H.P. Schwefel. *Numerical Optimization of Computer Models*. John Wiley & Sons, 1981.

[8] A. Törn and A. Zilinskas. *Global Optimization*. Springer-Verlag, Berlin, 1989.

Proceedings of GOW'16, pp. 133 – 136.

An Interval Branch and Bound Algorithm for Parameter Estimation

Bertrand Neveu[1], Martin de la Gorce[1] and Gilles Trombettoni[2]

[1]*LIGM, Ecole des Ponts Paristech, Université Paris Est, France,* Bertrand.Neveu@enpc.fr

[2]*LIRMM, Université de Montpellier, France,* Gilles.Trombettoni@lirmm.fr

Abstract The parameter estimation problem is a widespread and challenging problem in engineering sciences consisting in computing the parameters of a parametric model that fit observed data. The computer vision community has proposed the RANSAC algorithm to deal with outliers in the observed data. This randomized algorithm is efficient but non-deterministic and therefore incomplete. Jaulin et al. propose a branch-and-contract algorithm that returns all the model instances fitting at least q observations. Assuming that at least q observed data are inliers, this algorithm achieves on the observations a relaxed intersection operator called q-intersection. First, this paper presents several improvements to Jaulin et al.'s algorithm. Second, an interval branch and bound algorithm is designed to produce a model that can explain the maximum number of observations within a given tolerance. Experiments are carried out on computer vision and image processing problems. They highlight a significant speedup w.r.t. Jaulin et al.'s interval method in 2D and 3D shape recognition problems. We have also investigated how the approach scales up in dimensions up to 7 for stereovision (estimation of essential and fundamental matrices).

Keywords: Interval branch and bound, Shape detection, Parameter estimation

1. Parameter estimation

Parameter estimation is a difficult problem widely studied by engineering sciences. It consists in determining the n numerical parameters of a model based on m observations. Calibration or geolocation can be viewed as specific parameter estimation problems. A parameterized model is defined by an implicit equation $f(\mathbf{x}, \mathbf{p}) = 0$, $\mathbf{p} = (p_1, \ldots, p_n)$ being the vector of parameters to be determined. Given a finite set of observations $\{\mathbf{o_1}, \ldots, \mathbf{o_i}, \ldots, \mathbf{o_m}\}$, we search for all the parameters vectors that are compatible with at least q of these observations. We generally have $n \leq q \leq m$. An observation $\mathbf{o}_i$ is a d-dimensional vector of observed data. It is said compatible with the parameters vector $\mathbf{p}$, using a tolerance value τ, when it satisfies an *observation constraint* $-\tau \leq f(\mathbf{o}_i, \mathbf{p}) \leq +\tau$. The *consensus set* $C(\mathbf{p})$ is the set of observations compatible with $\mathbf{p}$.

$$C(\mathbf{p}) = \{\mathbf{o}_i | -\tau \leq f(\mathbf{o}_i, \mathbf{p}) \leq \tau\} \tag{1}$$

This parameter estimation problem becomes challenging when the function f used to define the parametric model is not linear and/or in presence of outliers. Outliers can have numerous origins, including extreme values of the noise, erroneous measurements and data reporting errors. In order to cope with outliers we search for model instances whose consensus set contains at least q elements.

This problem can be formulated as a numerical constraint satisfaction problem with n variables $\mathbf{p} = (p_1, \ldots, p_n)$ having a real interval domain, and a single constraint stating that at least q observations are compatible with the model:

$$card(C(\mathbf{p})) \geq q \tag{2}$$

The optimization version of this problem simply consists in maximizing the cardinality of the consensus set, i.e. q.

RANSAC: parameter estimation heuristic coping with outliers

The random sample consensus algorithm (RANSAC) [1] has become a state-of-the-art tool in the computer vision and image processing communities to achieve parameter estimation robust to outliers. This stochastic algorithm proceeds by randomly sampling observations for determining a model (n observations for determining n parameters), before checking the number of other observations compatible with this model. A version of RANSAC presented in [2] is dedicated to the detection of *several* solutions (models), but it does not detect all of them. Indeed, when a solution is found, this non deterministic algorithm removes the observations involved in the consensus set before searching for a next solution.

Deterministic interval constraint programming approach

A deterministic parameter estimation method based on interval constraint programming and robust to outliers was described in [3, 4].

We denote by $[x_i] = [\underline{x_i}, \overline{x_i}]$ the interval/domain of a real-valued variable x_i, where $\underline{x_i}$, $\overline{x_i}$ are floating-point numbers. A Cartesian product of intervals $[\mathbf{x}] = [x_1] \times ... \times [x_n]$ is called a (parallel-to-axes) *box*. The width of a box is given by the width $\overline{x_k} - \underline{x_k}$ of its largest dimension x_k. Interval methods also provide *contracting operators* (called contractors), i.e. methods that can reduce the variable domains involved in a constraint or a set of constraints without loss of solutions. In particular, a simple forward-backward (also called HC4-revise) algorithm traverses twice the expression tree corresponding to a given constraint to contract the domains of its variables [5, 6].

The deterministic parameter estimation algorithm performs a tree search to exhaustively explore the parameter space.

- $[\mathbf{p}]$ is recursively subdivided: one variable p_i in $\mathbf{p}$ is selected, its domain $[p_i]$ is bisected into two sub-intervals and the two corresponding sub-boxes are explored recursively. The combinatorial process stops when a precision is reached, i.e. when the width of the current box is inferior to ϵ_{sol}.

- At each node of the tree, a box $[\mathbf{x}]$ is handled:

  1. A contraction is achieved using each of the m observation constraints by the forward-backward procedure, which produces an m-set S of sub-boxes of $[\mathbf{x}]$.
  2. The q-intersection box $\cap^q S$ of these contracted boxes is returned.

The q-intersection operator relaxes the (generally empty) intersection of m boxes by the union of all the intersections obtained with q boxes. More formally:

Definition 1. *Let S be a set of boxes. The q-intersection of S, denoted by $\cap^q S$, is the box of smallest perimeter that encloses the set of points of $\mathbb{R}^n$ belonging to at least q boxes.*

For instance, the box in dotted lines in Figures 1–2 is the 4-intersection of the $m = 10$ two-dimensional boxes (in plain lines).

The q-intersection of boxes is a difficult problem that has been proven DP-complete in [7] and we have resorted to a non optimal projection heuristic that reasons on each dimension independently [8]. This algorithm is time $O(nm\log(m))$.

2. Improvements

We have proposed several generic improvements to the deterministic parameter estimation code, and several improvements specific to shape recognition problems.

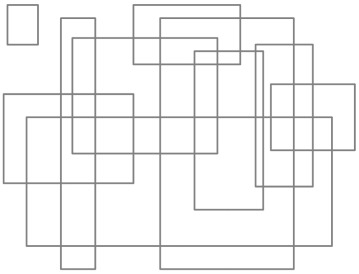


Figure 1. A set S of 2-dimensional boxes

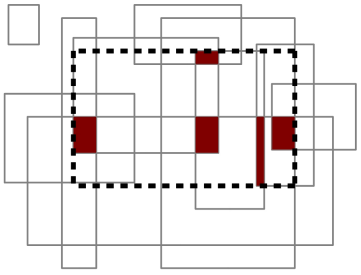


Figure 2. The dashed box is $\cap^4 S$. Zones that belong to at least 4 boxes are darkened. Illustration of q-intersection for $q = 4$, $m = 2$.

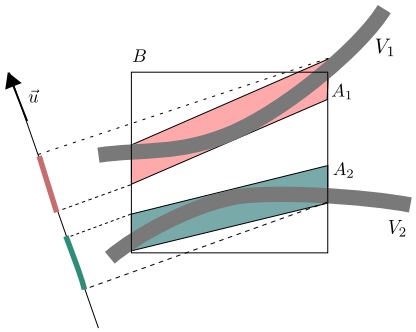


Figure 3. Approximation of the observation constraints V_i by parallelograms A_i, and projection on the normal vector $\underline{u}$. Q-intersection in an additional direction.

2.1 Generic improvements

Possible and valid observations. In the search tree, two data structures are maintained. First, the set of *possible* observations: if an observation constraint leads to an empty box using a (forward-backward or q-intersection) contraction at a given node, this observation will be removed from the possible observations in the subtree.

Second, the number of *valid* observations is maintained by testing every possible observation (using (1)) at given punctual parameters vectors inside the studied box. The valid observations form a subset of the possible observations. A stopping condition in the current branch of the search is reached when the two sets are the same.

Q-intersection in an additional direction. The q-intersection algorithm achieves a projection on each dimension (called q-projection) of the boxes obtained by contraction using every observation constraint.

We also perform a q-projection on an additional direction where we hope to obtain small intervals, thus favoring a failure of the q-intersection. To this end, we linearize and relax every observation constraint, and project the parallelograms obtained on the direction corresponding to the mean normal vector of the "parallelogram" gradients. See Figures 1–3 for a 2D illustration.

In an improved version, the q-projection is achieved only in the additional direction, except in the lowest part of the search tree where all the dimensions are handled.

2.2 Improvements specific to applications

Dedicated contraction. Instead of running a general forward-backward contraction algorithm using a library for interval arithmetic computations and backward projections (e.g., implemented in `Ibex`), we can rewrite interval computations dedicated to the analytical form of observation constraints.

Bisection strategy. For 3D plane recognition, we bisect first the intervals of the variables corresponding to the plane normal vector and bisect the variable intervals modeling distances to the origin only when the plane normal vector intervals have reached a good precision. For 2D circle recognition, we first handle the circle center coordinates and terminate with the circle radius.

3. Parameter estimation fitting a maximum number of inliers

We have also designed an interval branch and bound algorithm for parameter estimation that computes a model maximizing the consensus, i.e. maximizing the number of valid observations (inliers) of a parameterized model. Several strategies have been designed: depth-first search (and a variant) and best-first search.

Contraction using forward-backward procedures and q-intersection is performed at each iteration (node). The lower bound q_{min} of this maximization problem is given by the number of observations that have been validated in past iterations. An upper bound q_{max} of the number of inliers in a node is given by the maximum number of intersected intervals found by the $(q_{min}+1)$-projection procedure in a dimension. If $q_{max} \leq q_{min}$, then the branch is pruned. q_{max} may be inferior to the number of possible observations in the box, in particular if the box contains several valid models.

4. Experiments

The algorithms are implemented in the Interval Based EXplorer (`Ibex`) [9], a free C++ library devoted to interval computing. The combination of the improvements described in Section 2 brings a significant speedup of two orders of magnitude on each tested instance of 3D plane and 2D circle detection problems and appears to be an interesting alternative to RANSAC in low dimension. These experiments suggest that our interval branch and bound algorithm can guarantee a model maximizing the number of inliers while ensuring a good performance.

5. Discussion

A question is whether the approach scales up in higher dimension. First experiments seem to show that the current interval branch and bound algorithm cannot cope with the fundamental matrix estimation problem (dimension 7) useful in stereovision. We will investigate whether the approach can handle parameter estimation problems of dimension 4 or 5 (essential matrix estimation [10]).

References

[1] M. A. Fischler and R. C. Bolles. Random Sampling Consensus: Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. of the ACM*, 24(6):381–395, 1981.

[2] R. Schnabel, R. Wahl, and R. Klein. Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum*, 26(2):214–226, 2007.

[3] L. Jaulin and E. Walter. Guaranteed Robust Nonlinear Minimax Estimation. *IEEE Transactions on Automatic Control*, 47, 2002.

[4] L. Jaulin and S. Bazeille. Image Shape Extraction using Interval Methods. In *Proc. of the 16th IFAC Symposium on System*, pages 378–383, 2009.

[5] F. Benhamou, F. Goualard, L. Granvilliers, and J.-F. Puget. Revising Hull and Box Consistency. In *Proc. of International Conference on Logic Programming (ICLP)*, pages 230–244, 1999.

[6] F. Messine. *Méthodes d'optimisation globale basées sur l'analyse d'intervalle pour la résolution des problèmes avec contraintes*. PhD thesis, LIMA-IRIT-ENSEEIHT-INPT, Toulouse, 1997.

[7] C. Carbonnel, G. Trombettoni, P. Vismara, and G. Chabert. Q-Intersection Algorithms for Constrained-Based Robust Parameter Estimation. In *Proc. of Conference of the Association for the Advancement of Artificial Intelligence (AAAI)*, pages 2630–2636, 2014.

[8] P. Chew and K. Marzullo. Masking failures of Multidimensional Sensors. In *Proc. of Reliable Distributed Systems*, pages 32–41, 1991.

[9] G. Chabert and L. Jaulin. Contractor Programming. *Artificial Intelligence*, 173:1079–1100, 2009.

[10] J. Yang, H. Li, and Y. Jia. Optimal Essential Matrix Estimation via Inlier-Set Maximization. In *Proc. of European Conference of Computer Vision (ECCV)*, 2014.

Proceedings of GOW'16, pp. 137 – 140.

Solving the 1-median Problem on a Network with Demand Surplus*

Kristóf Kovács[1], Rafael Blanquero[2], Emilio Carrizosa[2] and Boglárka G.-Tóth[1]

[1]*Budapest University of Technology and Economics, Budapest, Hungary,* kkovacs@math.bme.hu, bog@math.bme.hu

[2]*Universidad de Sevilla, Seville, Spain* rblanquero@us.es, ecarrizosa@us.es

Abstract Our aim is to find the optimal location of a facility on a network, where demand is distributed on the nodes and edges, while only a given percent of the demand has to be covered. An edge is either covered entirely or not at all. This inner knapsack problem has to be solved as well as the location of the facility.

Keywords: Global optimization, Network location, Facility location, Knapsack problem, Branch and bound

1. Introduction

Location problems on networks have been around since the 60s of the last century. In most cases demand is concentrated on the nodes of the network, while facilities can be located on the edges. Although several researchers have proposed models where the edges contain demand as well, usually the demand is uniformly distributed along the edge. There are only a few papers dealing with arbitrary distributions, e.g. [1, 2, 3, 4].

In our work, we considered a single facility location problem with demand distributed at the nodes and along the edges of the network. The distribution of the demand on the edges is arbitrary. Contrary to the planar 1-median problem, known to be convex, the objective of this problem with continuous demand is much more complex.

2. Problem formulation

Let $N = (A, E)$ be a connected and undirected network, with node set $A = \{a_1, a_2, \dots a_n\}$ and edge set $E = \{e_1, e_2, \dots e_m\}$, where $|A| = n$ and $|E| = m$. Furthermore let l_{ij} denote the length of edge $(a_i, a_j) \in E$ and $d(x, y)$ denote the distance between two points on the network $x, y \in N$. The distance is understood as the length of the shortest path from x to y.

We assume the demand is distributed at the nodes as well as along the edges of the network. The demand of node a is denoted by $w_a \geq 0$, while the total demand distributed on a given edge e is $p_e \geq 0$. The distribution of the demand on edge e is given by a random variable with cumulative distribution function (cdf) F_e. Lastly the sum of the demand on the whole network is denoted by $D = \sum_{a \in A} w_a + \sum_{e \in E} p_e$.

Our objective is to minimize the distance of a facility to a set of the costumers weighted by their demand. At least a fraction α of the overall demand has to be covered. The inclusion of edges and nodes is binary, thus, if an edge or node is included, its whole demand has to be served.

*This work is funded by the Hungarian National Research, Development and Innovation Office - NKFIH, OTKA grant PD115554 and by research grants and projects MTM2015-65915-R (Ministerio de Economía y Competitividad, Spain), P11-FQM-7603 and FQM-329 (Junta de Andalucía, Spain). This work was also supported by an STSM Grant from COST Action TD1207.

$$\min_{A^*\subseteq A, E^*\subseteq E, x\in N} G(x) := \sum_{a\in A^*} w_a d(x,a) + \sum_{e\in E^*} p_e \int_{b\in e} d(x,b)\mathrm{d}F_e(b) \tag{1}$$

$$\text{s.t.} \sum_{a\in A^*} w_a + \sum_{e\in E^*} p_e \geq \alpha D \tag{2}$$

To solve this problem we propose a Branch and Bound algorithm. Branching is done on the location and the cover variables simultaneously, while bounds are computed using the relaxed solution of the cover problem.

3. The optimal cover of the demand for a fixed location

To find the optimal solution of (1)-(2), we need to find a cover of the edges that satisfies (2) and minimizes (1) for a given facility location x. The edges and nodes are either fully covered or not at all. Hence, for a location x fixed, an optimal cover can be obtained by solving a knapsack problem, where the demand of the edges and nodes are the weights of the items, while the value of the items is the cost of serving them, denoted by u_a for node a and v_e for edge e.

We approach the problem from the other way around, trying to find the edges and nodes which should be excluded from the cover. This way the problem resembles the usual knapsack problem. Thus we have to find a subset of the edges and nodes, that has the greatest cost, while the sum of their demand does not exceed the limit of the knapsack, which is $(1-\alpha)D$. The complementary set of this one will be an optimal cover for the original problem. The knapsack problem in formula is

$$\max_{y,z} \sum_{a\in A} u_a y_a + \sum_{e\in E} v_e z_e$$
$$\text{s.t.} \sum_{a\in A} w_a x_a + \sum_{e\in E} p_e y_e \leq (1-\alpha)D,$$
$$y_a, z_e \in \{0,1\} \quad \forall a \in A, e \in E.$$

The variables y, z are binary vector variables that represent the inclusion of the items in the knapsack. An example of a cover can be seen on Figure 1.

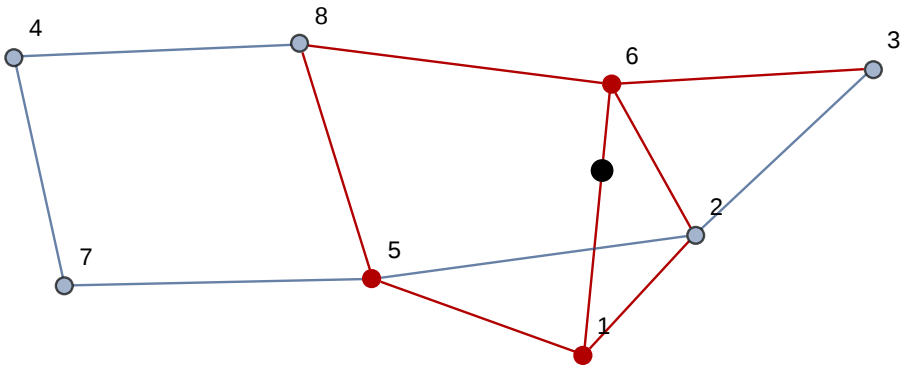


Figure 1. Example of a cover, where the facility is placed at the black circle. Included nodes and edges are shown in red.

The difficulty in solving the knapsack problem comes when the facility location is not given, and instead, it is known to belong to a segment of an edge, an interval. In this case, the cost

values, u_a, v_e are also interval valued, and an upper bound of the objective function is needed. If two cost intervals intersect we can not make a definitive choice in including one or the other, thus the knapsack problem given as such does not have an exact solution unless there is an ordering on the cost intervals. This ordering exists if and only if the cost intervals are disjoint. Luckily, every edge can be split into segments, such that within each segment an ordering on the corresponding cost intervals exist. Calculating these segments is computationally expensive, but once such a segment is obtained its knapsack problem can be solved to optimality.

4. Relaxation of the knapsack problem

As the exact solution of the knapsack problem is expensive, it is natural to use cheaper lower bounds, such as the continuous relaxation, which can be calculated directly and provides only one fractional variable (item) in the solution, called the split item: The items have to be ordered in decreasing order by their value to weight ratio; if we put the items into the knapsack in this order, the first item that do not fit is the split item.

5. Computational observations

Unfortunately working on the knapsack problem alongside the location problem is not effective due to the previously mentioned integrity gap. Thus the branching should be done on the location variable until it is narrowed down to a small enough segment where solving the knapsack problem to optimality is possible.

We propose two branching strategies. The first one relies on branching along the midpoint of the location segment, while the second strategy branches so that the resulting knapsack problems can be solved exactly. The second method allows us to start solving the knapsack problem when the exact solution can be found without any unnecessary location branches.

Due to the effect of the integrity gap on the bounds sharpness, the first strategy is expected to work better for large networks, while the second strategy would be more competitive for smaller ones.

6. Summary

We proposed a solution to the 1-median problem on a network with demand surplus. A Branch and Bound method is used with two branching strategies. The difficulty of the problem comes from the inner knapsack problem. Its exact solution for the upper bound is difficult to find due to the interval nature of the item costs. If the exact solution of the knapsack problem is not calculated, the lower bound does not converge to the optimal value.

The implementation of the algorithm utilizes a sorting method to decrease the computational cost of calculating the knapsack problem's relaxed solution. We believe this method will reliably solve small and medium sized networks, using the appropriate branching strategy according to the size of the network.

References

[1] Rafael Blanquero and Emilio Carrizosa. Solving the median problem with continuous demand on a network. *Computational Optimization and Applications*, 56(3):723–734, 2013.

[2] Samuel S Chiu. The minisum location problem on an undirected network with continuous link demands. *Computers & Operations Research*, 14(5):369 – 383, 1987.

[3] Ma Cruz López de los Mozos and Juan A. Mesa. The variance location problem on a network with continuously distributed demand. *RAIRO-Operations Research*, 34(2):155–182, 2000.

[4] Paul T. Nkansah and H. T. David. Network median problems with continuously distributed demand. *Transportation Science*, 20(3):213–219, 1986.

Proceedings of GOW'16, pp. 141 – 144.

MILP Model for Batch Scheduling on Parallel Machines

Renan S. Trindade[1], Olinto Araújo[2], Marcia Fampa[1] and Felipe Müller[3]

[1]*Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brazil,* trindade@cos.ufrj.br, fampa@cos.ufrj.br

[2]*CTISM, Universidade Federal de Santa Maria, Santa Maria, RS, Brazil,* olinto@ctism.ufsm.br

[3]*Universidade Federal de Santa Maria, Santa Maria, RS, Brazil,* felipe@inf.ufsm.br

Abstract We consider a mixed integer linear programming model from the literature for the problem of minimizing the makespan on parallel batch processing machines with non-identical job sizes. Symmetric and equivalent solutions in the feasible set of the problem lead to an inefficient application of branch-and-bound algorithms. We present valid inequalities to discard these solutions and show through computational results that our strengthened model clearly outperforms the model from the literature. With the model proposed in this work, we prove for the first time optimality for instances of this problem with more than 25 jobs.

Keywords: Batch scheduling, Mixed integer linear programming, Symmetry, Valid inequality

1. Introduction

Scheduling problems on batch processing machines have been widely exploited in the literature, mainly motivated by the vast number of applications in industries. The idea is to group jobs in batches and process all jobs on the same batch at the same time in a machine, avoiding setups and facilitating the material manipulation.

We address the problem of minimizing the makespan on parallel batch processing machines with non-identical job sizes, which models an important application in the semiconductor industry, namely the scheduling problem in the burn-in operation at the testing phase of semiconductors [9]. The problem is strongly NP-hard [1, 9] and has been categorized as $P_m|s_j, B|C_{\max}$ on the three-field system proposed in [4].

Several works have considered this specific batch scheduling problem, but mainly focusing on heuristic approaches (see, for example, [1, 2, 3, 5, 6]). In [1], the problem is formulated as a mixed integer linear program (MILP), and solved by a branch-and-bound (B&B) algorithm. Optimal solutions are only reported for instances with up to 25 jobs. The B&B, when applied to the MILP formulation proposed, solves several subproblems corresponding to symmetric and equivalent solutions, which makes it very inefficient. The difficulty related to symmetry in integer programming is well known [8], and the impact of symmetry breaking cuts on the solution of a particular application in software engineering [7], motivated us to develop valid inequalities to problem $P_m|s_j, B|C_{\max}$. The validity of the inequalities proposed in this work is supported by proven results about the optimal solution of problem $P_m|s_j, B|C_{\max}$, and their strength is demonstrated through computational experiments. With the new model proposed, we prove for the first time optimality for instances of this problem with more than 25 jobs.

In Section 2, we define problem $P_m|s_j, B|C_{\max}$ and describe the MILP formulation proposed in [1]. In Section 3, we propose valid inequalities for the problem and present the result that support their validity. A new strengthened model is proposed and in Section 4, computational results are used to compare the models and show the impact of the inequalities introduced.

2. A MILP model for minimizing makespan on parallel batch processing machines

On problem $P_m|s_j, B|C_{\max}$, a set J of jobs is considered. Each job $j \in J$ is characterized by a processing time p_j and a size s_j, and must be assigned to a batch $k \in K$, not exceeding the capacity B of the machine. The batches must be designed and scheduled on an $m \in M$ parallel machine. The objective is to minimize the makespan ($C_{\max}$), defined as the time required to finish the processing of the last batch processed on all the machines. The processing time of batch k in machine m is given by $P_{k,m} = \max\{p_j : j \text{ is assigned to batch } k \text{ and processed by machine } m\}$. As the machine cannot be interrupted during the batch processing, no job can be added to or removed from the machine until the processing is finished. Furthermore, jobs cannot be divided between batches.

In [1], $P_m|s_j, B|C_{\max}$ is formulated as the following MILP.

(P_1): $\min\ C_{\max}$: (1)

subject to :

$$\sum_{k\in K}\sum_{m\in M} x_{j,k,m} = 1 \qquad \forall j \in J \quad (2)$$

$$\sum_{j\in J}\sum_{m\in M} s_j x_{j,k,m} \leq B \qquad \forall k \in K \quad (3)$$

$$P_{k,m} \geq p_j x_{j,k,m} \qquad \forall j \in J, \forall k \in K, \forall m \in M \quad (4)$$

$$C_{\max} \geq \sum_{k\in K} P_{k,m} \qquad \forall m \in M \quad (5)$$

$$P_{k,m} \geq 0 \qquad \forall k \in K, \forall m \in M \quad (6)$$

$$C_{\max} \geq 0 \quad (7)$$

$$x_{j,k,m} \in \{0,1\} \qquad \forall j \in J, \forall k \in K, \forall m \in M \quad (8)$$

The variable $P_{k,m}$ represents the processing time of batch k on machine m, and $x_{j,k,m}$ indicates if job j is assigned to batch k and processed by machine m. The objective function value in (1) is the makespan. Constraints (2-3) state that each job is assigned to only one batch and one machine, all respecting the machine capacity. Constraint (4) sets the processing time of batch k assigned to machine m and (5) takes the time of the last machine to finish the processing. A batch is considered open if at least one job is assigned to it. As the number of open batches is bounded by $|J|$, $|K|$ is set equal to $|J|$.

Problem P_1 has a large number of symmetric solutions in its feasible set, defined as solutions with identical schedules for equal batches, but indexed differently. These solutions are considered distinct by a B&B algorithm, making it very inefficient, solving symmetric subproblems several times. Besides symmetric solutions, the feasible set of P_1 also contains equivalent solutions, contributing similarly to the inefficiency of the B&B. Equivalent solutions are defined as solutions where the jobs ordering in the batches is permuted, which has no effect on the makespan.

3. Valid inequalities

We propose valid inequalities to problem $P_m|s_j, B|C_{\max}$ that eliminate symmetric and equivalent solutions from the feasible set of P_1.

Firstly, we replace the variable $x_{j,k,m}$, which is responsible for both batch dimensioning and for machine allocation, with the two new variables: $x_{j,k}$, which indicates if job j is assigned to batch k, and $y_{k,m}$, which indicates batch k is assigned to machine m. This substitution reduces significantly the number of variables in the problem.

As mentioned in the previous section, many symmetries are allowed by the formulation P_1 because the ordering in which the jobs are placed in the batches does not interfere in the final optimal solution value. Let us consider, therefore, the jobs in J to be indexed in non-decreasing order of processing times ($p_1 \leq p_2 \leq ... \leq p_{|J|}$). We propose then the two following constraints:

$$\sum_{k \in K: j \leq k} x_{j,k} = 1 \qquad \forall j \in J \tag{9}$$

$$x_{j,k} \leq x_{k,k} \qquad \forall j \in J, \forall k \in K : j < k \tag{10}$$

Constraints (9) determine that a job j can only be assigned to a batch k, if $j \leq k$, and constraints (10) determine that batch k can only be open if job k is assigned to it. These constraints eliminate several solutions from the feasible set of P_1. Furthermore, (9) implies that $x_{j,k} = 0$, if $j > k$, allowing a significant further reduction on the number of binary variables in the model. As the jobs are sorted in non-decreasing processing time, and batch k can only be open if job k is assigned to it, the processing time of a existing batch k is fixed as p_k. That is, each batch k always have the job k as the highest index.

Thus, the solution of the formulation P_2 that we propose in the following, will always have a non-decreasing order of processing times of batches when assigned to machines. Proposition 1 ensures that no optimal solution is cut off when the problem is conceived in this way.

Proposition 1. *Any optimal solution for $P_m|s_j, B|C_{\max}$ can be represented considering batches sequenced in non-decreasing order of the processing times of batches.*

We propose, therefore, the following strengthened model to problem $P_m|s_j, B|C_{\max}$.

(P_2): $\min\ C_{\max}$ (11)

subject to

(9)(10)

$$\sum_{j \in J: j \leq k} s_j x_{j,k} \leq B x_{k,k} \qquad \forall k \in K \tag{12}$$

$$x_{k,k} \leq \sum_{m \in M} y_{k,m} \qquad \forall k \in K \tag{13}$$

$$C_m \geq \sum_{k \in K} p_k y_{k,m} \qquad \forall m \in M \tag{14}$$

$$C_{\max} \geq C_m \geq 0 \qquad \forall m \in M \tag{15}$$

$$x_{j,k} \in \{0, 1\} \qquad \forall j \in J, \forall k \in K : j \leq k \tag{16}$$

The objective function value in (11) is the makespan. Constraints (12) ensure that the machine capacity is respected. Constraints (13) state that if batch k is open, it should be assigned to a machine. Constraints (14-15) set $C_{\max}$ equal to the time of the last machine to finish the processing.

4. Computational results and conclusions

We analyze the strength of the valid inequalities proposed in this work, comparing the performance of the solver `CPLEX 12.5`, when applied to P_1 and P_2. All runs were conducted on an Intel Quad-Core Xeon X3360 2.83 GHz, 8GB, running under Linux, with a time limit of 30 minutes. The number of jobs in our tests varies from 10 to 100, the machine capacity is 10

and the number of parallel machines is 2, 4 or 8. The instances were created from the random discrete uniform distribution. The processing times are selected in two intervals: $p_1 : [1, 10]$ and $p_2 : [1, 20]$. The job sizes are selected in three intervals: $s_1 : [1, 10]$, $s_2 : [2, 4]$, $s_3 : [4, 8]$.

The smaller jobs in s_2 increase the number of possible jobs combinations in each batch and make the problems more difficult, in contrast with instances of type s_3. We solved 100 instances for each combination of job size, processing time and number of parallel machines, a total of 9600 instances. All the instances of type s_3 were solved to optimality with P_2 in an average time 0.14 seconds, while P_1 only can solve to optimality the instances with 10 jobs. With 20 jobs, P_1 reaches the time limit of 30 seconds in several instances of type s_3, and the final average duality gap was of 8.57%.

Concerning the instances of type s_1 optimality was reached with P_2 for the majority of instances with up to 100 jobs and with final duality gap not greater than 0.05%. P_2 proves the optimality of all instances of type s_1 with 20 jobs in an average time of 0.07 seconds, while P_1 has difficulty already in this case, with average final duality gap of 4,85%.

In case of instances of type s_3, P_2 still reaches optimal solutions in all the instances with up to 20 jobs in 0.18 seconds on average. Problem P_1 can only prove the optimality, in this case, when the number of machines is 8, in an average time of 2.78 seconds, while P_2 takes 0.08 seconds on average. When the number of machines is 2 or 4, P_1 presents final duality gap of 4,09%. For instances with up to 100 jobs, P_2 maintains its good performance keeping the average gap in 0.81% with 50 jobs and 1.50% with 100 jobs.

To our knowledge, this is the first time proven optimal solutions for instances with more than 25 jobs are reported, constituting the state of the art concerning results from exact methods to problem $P_m|s_j, B|C_{\max}$. Substantial gains were obtained with the inclusion of the valid inequalities proposed in the MILP formulation presented in the literature, decreasing the average running time for instances solved to optimality, and leading to a practical application of the B&B algorithm to the problem.

References

[1] P.-Y. Chang, P. Damodaran, and S. Melouk. Minimizing makespan on parallel batch processing machines. *International Journal of Production Research*, 42(19):4211–4220, oct 2004.

[2] B. Cheng, Q. Wang, S. Yang, and X. Hu. An improved ant colony optimization for scheduling identical parallel batching machines with arbitrary job sizes. *Applied Soft Computing*, 13(2):765–772, feb 2013.

[3] P. Damodaran, N. S. Hirani, and M. C. V. Gallego. Scheduling identical parallel batch processing machines to minimise makespan using genetic algorithms. *European J. of Industrial Engineering*, 3(2):187, 2009.

[4] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling : a survey. *Annals of Discrete Mathematics, Elsevier*, 5:287–326, 1979.

[5] Zhao-hong Jia and Joseph Y.-T. Leung. A meta-heuristic to minimize makespan for parallel batch machines with arbitrary job sizes. *European Journal of Operational Research*, 240(3):649–665, feb 2015.

[6] A. H. Kashan, B. Karimi, and M. Jenabi. A hybrid genetic heuristic for scheduling parallel batch processing machines with arbitrary job sizes. *Computers & Operations Research*, 35(4):1084–1098, apr 2008.

[7] V. Köhler, M. Fampa, and O. Araújo. Mixed-Integer Linear Programming Formulations for the Software Clustering Problem. *Computational Optimization and Applications*, 55(1):113–135, oct 2012.

[8] F Margot. Symmetry in integer linear programming. In *50 Years of Integer Programming 1958-2008*, pp 647–686. Springer, Berlin Heidelberg, 2010.

[9] R. Uzsoy. Scheduling a single batch processing machine with non-identical job sizes. *International Journal of Production Research*, 32:1615–1635, jul 1994.

Proceedings of GOW'16, pp. 145 – 148.

A Clustering-Based Algorithm for Aircraft Conflict Avoidance

Sonia Cafieri[1] and Emilio Carrizosa[2]

[1] *ENAC, MAIAA, F-31055 Toulouse, France*
University of Toulouse, IMT, F-31400 Toulouse, France, sonia.cafieri@enac.fr

[2] *IMUS-Instituto de Matemáticas de la Universidad de Sevilla*
Universidad de Seville, Spain, ecarrizosa@us.es

Abstract Aircraft conflict avoidance is a critical issue in Air Traffic Management, which can be addressed, among others, by means of Mixed Integer Non-Linear Programming (MINLP) techniques.

In this work we introduce a new approach to address, via velocity regulation, the problem of avoiding conflict for a set of aircraft flying in an air sector at cruise flight. Speed variations for all aircraft are to be minimized so that, at any time instant, the horizontal distance between each pair of aircraft is above a threshold security value. The problem can be expressed as a MINLP, solvable with standard MINLP solvers for a small number of aircraft, but intractable for instances of more realistic size. This motivates us the design of a cluster-based procedure.

In our approach, aircraft are clustered into groups, so that within each cluster the aircraft are conflicting, while conflicts do not exist (or they are at least less severe) between aircraft in different clusters. Then a MINLP solver is used sequentially on each cluster to minimally modify the aircraft speeds so that conflicts within the cluster are solved, not creating new conflicts with aircraft in other clusters.

Theoretical convergence of the procedure and preliminary numerical results will be discussed in the talk.

Keywords: Clustering, MINLP, Aircraft conflict avoidance, Subliminal speed control

1. Introduction

One of the main and most crucial tasks of air traffic control services is continuously monitoring aircraft trajectories, detecting potential *conflicts*, i.e., potential loss of separation between trajectories, and issuing appropriate conflict resolution maneuvers. The increasing air traffic on the world-scale has an immediate impact on air traffic controllers' workload, making it more and more difficult to handle aircraft conflicts. Hence, a higher level of automation in Air Traffic Management and Control urgently needs to be introduced, so that automatic aircraft conflict avoidance procedures have in particular received a growing attention over the past few years.

Mathematical optimization naturally arises in this context, as one usually aims at separating conflicting aircraft while optimizing a selected criterion (e.g., stay as close as possible to the original trajectory, or minimizing delays induced by separation maneuvers). Mixed-Integer Nonlinear Programming is attracting a growing attention for the considered real-world application, as it enables to model the complex nonlinear (nonconvex) aircraft separation constraints while considering mixed variables (continuous variables typically used for aircraft speeds, heading angles, etc., and integer ones typically used for logical choices). First approaches based on mixed-integer optimization date back to 2002 ([5, 8]), and more recently, were proposed in [1, 2, 3, 6, 7].

One can observe that the complexity of the problem under consideration is specially related to the nonlinear nonconvex separation constraints, that are indexed on all pairs of aircraft, and

whose number increases quadratically with the number of aircraft. Hence, exact solution algorithms easily turn to be computational demanding, specially when the number of aircraft considered simultaneously is large. On the other hand, in realistic situations, the airspace section under consideration is a quite large portion of the airspace where generally only small groups of aircraft with close trajectories may potentially be in conflict. This suggest to decompose the problem into smaller subproblems (clusters) of conflicting aircraft. Such an approach was introduced in [3], where however a procedure to create subgroups of aircraft was not proposed.

In the present work, we propose to decompose the overall problem into subproblems involving only a small number of aircraft and to perform conflict avoidance exactly on these subproblems, then combining all the obtained solutions. Subproblems are described through mixed-integer nonlinear programs, and are constructed by clustering aircraft using a suitable dissimilarity measure. Each cluster is solved to optimality, then clusters are modified if conflicts are still present, and the process is repeated until no conflicts occurr anymore.

2. Problem modeling

Let us consider a set A of n aircraft flying during their cruise flight in a given air sector, all at the same flight level. The horizontal separation only has to be satisfied. This means that, at any time instant $t \geq 0$, the distance between any pair of aircraft i, j should not be smaller than the horizontal separation standard d. Assuming that uniform motion laws apply, the position $\mathbf{x}_i(t)$ of aircraft i at time t is given by

$$\mathbf{x}_i(t) = \mathbf{x}_i(0) + t\mathbf{v}_i, \tag{1}$$

where both the initial position $\mathbf{x}_i(0)$ and velocity $\mathbf{v}_i$ are assumed to be known, and the distance $\|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|$ between i and j satisfies

$$\|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|^2 = t^2\|\mathbf{v}_i - \mathbf{v}_j\|^2 + 2t\,(\mathbf{v}_i - \mathbf{v}_j) \cdot \mathbf{x}_{ij}^{r0} + \|\mathbf{x}_{ij}^{r0}\|^2, \tag{2}$$

where $\|\cdot\|$ is the Euclidean norm in the two-dimensional space and $\mathbf{x}_{ij}^{r0} = \mathbf{x}_i(0) - \mathbf{x}_j(0)$ denotes the relative position at time $t = 0$ of aircraft i with respect to j.

Since, keeping the original trajectories, equation $\|\mathbf{x}_i(t) - \mathbf{x}_j(t)\| \geq d$ may not be fulfilled i.e., aircraft may be in conflict, suitable separation maneuvers, corresponding to trajectory deviations, have to be carried out. We consider aircraft speed deviations, following the concept of *subliminal* speed control, that was introduced as a promising method to improve traffic congestion while maintaining a low impact on controller's workload [4]. In this framework, we allow each aircraft i to (slightly) modify its speed from $\mathbf{v}_i$ to $q_i\mathbf{v}_i$, where the speed variations q_i are bounded in the interval $[-6\%v_i, +3\%v_i]$.

Speed variations q_i for all i, represent the main decision variables of our nonlinear optimization problem, that is formulated as follows:

$$\begin{array}{lll} \min & \sum_{i \in A}(q_i - 1)^2 & \\ \text{s.t.} & d_{ij}(q_i, q_j) \geq d & \forall i, j \in A, i < j \\ & q_{\min} \leq q_i \leq q_{\max} & \forall i \in A \end{array} \tag{3}$$

where $q_{\min}$ and $q_{\max}$ are lower and upper bounds on q_i, and $d_{ij}(q_i, q_j)$ represents the distance between aircraft i and j. It is a nonlinear nonconvex expression, as detailed below.

3. Aircraft clustering-based algorithm

For speed variations q_i, $i \in A$, the second-degree polynomial function in equation (2) attains its minimum in $[0, \infty)$ at time instant

$$\max\left\{0, t_{ij}^m(q_i, q_j)\right\}, \tag{4}$$

with

$$t_{ij}^m(q_i, q_j) = \frac{-(q_i\mathbf{v}_i - q_j\mathbf{v}_j)\cdot \mathbf{x}_{ij}^{r0}}{\|q_i\mathbf{v}_i - q_j\mathbf{v}_j\|^2}. \tag{5}$$

Hence, by substituting in (2), we have that the separation between aircraft i and j is

$$d_{ij}(q_i, q_j) = \begin{cases} \dfrac{\sqrt{\|\mathbf{x}_{ij}^{r0}\|^2\|q_i\mathbf{v}_i - q_j\mathbf{v}_j\|^2 - \left(\mathbf{x}_{ij}^{r0}\cdot(q_i\mathbf{v}_i - q_j\mathbf{v}_j)\right)^2}}{\|q_i\mathbf{v}_i - q_j\mathbf{v}_j\|}, & \text{if } t_{ij}^m(q_i, q_j) \geq 0 \\ \|\mathbf{x}_{ij}^{r0}\|, & \text{else} \end{cases} \tag{6}$$

Equation (6) gives nonlinear nonconvex constraints for problem (3). Furthermore, the condition "if $t_{ij}^m(q_i, q_j) \geq 0$" leads us to introduce for each $i, j \in A, i < j$, the binary variables y_{ij} as

$$y_{ij} = \begin{cases} 1, & \text{if } t_{ij}^m \geq 0 \\ 0, & \text{else,} \end{cases} \tag{7}$$

or, equivalently, as

$$y_{ij} = \begin{cases} 1, & \text{if } (q_i\mathbf{v}_i - q_j\mathbf{v}_j)\cdot \mathbf{x}_{ij}^{r0} \leq 0 \\ 0, & \text{else,} \end{cases} \tag{8}$$

i.e., the constraint

$$\left((q_i\mathbf{v}_i - q_j\mathbf{v}_j)\cdot \mathbf{x}_{ij}^{r0}\right)(2y_{ij} - 1) \geq 0. \tag{9}$$

We decompose the overall problem into subproblems involving only a small number of aircraft and perform aircraft conflict resolution exactly on these subproblems, then combining all the obtained results. To do so, clustering is performed to make groups of aircraft. As a dissimilarity measure, we consider the pairwise critical distance for aircraft separation $d_{ij}(q_i, q_j)$ as defined in (6). The smaller the value of $d_{ij}(q_i, q_j)$, the more critical the conflict is between aircraft i and j, i.e., the more similar i and j are. We propose a hierarchical agglomerative clustering procedure using single linkage (closest distance) to merge groups, and imposing an upper bound σ on each cluster size in order to avoid clusters to be unbalanced in size.

Given aircraft with speeds $(\overline{q}_i\mathbf{v}_i)_{i\in A}$, the above clustering procedure yields a list of clusters $A_1, \ldots, A_m$. Such a list is sequentially inspected, and, for each cluster A_c, $c = 1, 2 \ldots, m$, an optimization problem is solved, in which the speeds of all aircraft j outside A_c is considered to be fixed to $\overline{q}_j\mathbf{v}_j$. More precisely, we seek the values $(q_i)_{i\in A_c}$ as close as possible to 1 so that

1. inside A_c, all conflicts are solved, i.e., $d_{ij}(q_i, q_j) \geq d$ for all $i, j \in A_c$, $i \neq j$,
2. for a given set of aircraft pairs $B_c \subset \left\{(i, j) : i \in A_c, j \notin A_c, d_{ij}(\overline{q}_i, \overline{q}_j) \geq d\right\}$, no conflict appears, i.e., $d_{ij}(q_i, \overline{q}_j) \geq d$.

Formally, the subproblem to be addressed for cluster A_c can be stated as

$$\begin{aligned} \min \quad & \textstyle\sum_{i\in A_c}(q_i - 1)^2 && && (10)\\ \text{s.t.} \quad & d_{ij}(q_i, q_j) \geq d && \forall (i, j) \in A_c, i < j && (11)\\ & d_{ij}(q_i, \overline{q}_j) \geq d && \forall (i, j) \in B_c && (12)\\ & q_{\min} \leq q_i \leq q_{\max} && \forall i \in A_c. && (13) \end{aligned}$$

Convergence issues and empirical performance on test instances will be presented.

Acknowledgments

Financial support by French National Research Agency (ANR) through grant ANR 12-JS02-009-01 "ATOMIC" is gratefully acknowledged. The research of Emilio Carrizosa is supported by projects P11-FQM-7603 (Junta de Andalucía) and MTM2015-65915-R (Ministerio de Economía y Competitividad), both enjoying EU ERDF funds.

References

[1] A. Alonso-Ayuso, L. F. Escudero, and F. J. Martín-Campo. On modeling the air traffic control coordination in the collision avoidance problem by mixed integer linear optimization. *Annals of Operations Research*, 222:89–105, 2014.

[2] A. Alonso-Ayuso, L. F. Escudero, P. Olaso, and C. Pizarro. Conflict avoidance: 0-1 linear models for conflict detection & resolution. *TOP*, 21(3):485–504, 2013.

[3] S. Cafieri and N. Durand. Aircraft deconfliction with speed regulation: new models from mixed-integer optimization. *Journal of Global Optimization*, 58(4):613–629, 2014.

[4] C. Dupré, D. Bonini, 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*, 2009.

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

[6] D. Rey, C. Rapine, V. V Dixit, and S. T. Waller. Equity-oriented aircraft collision avoidance model. *IEEE Transactions on Intelligent Transportation Systems*, 16(1):172–183, 2015.

[7] D. Rey, C. Rapine, R. Fondacci, and N.-E. El Faouzi. Subliminal speed control in air traffic management: Optimization and simulation. *Transportation Science*, 240–262, 2015.

[8] A. Richards and J.P. How. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *Proceedings of the American Control Conference 2002*, volume 3, pages 1936–1941 vol.3, 2002.

Proceedings of GOW'16, pp. 149 – 152.

On Solving Aircraft Conflict Avoidance Using Deterministic Global Optimization (sBB) Codes*

Sonia Cafieri[1], Frédéric Messine[2] and Ahmed Touhami[1,3]

[1]*ENAC, MAIAA, F-31055 Toulouse, France*
University of Toulouse, IMT, F-31400 Toulouse, France, sonia.cafieri@enac.fr

[2]*LAPLACE-ENSEEIHT-INPT, University of Toulouse, France,* frederic.messine@n7.fr

[3]*Faculty of Sciences and Technologies. Hassan I University, Settat, Morocco,* ahmed.touhami@gmail.com

Abstract In this paper, some improvements of spatial Branch and Bound (sBB) algorithms are discussed to solve aircraft conflict avoidance problems formulated as MINLP. We propose a new quadratic convex relaxation technique based on affine arithmetic. Moreover, a branching strategy is also proposed for the considered problem. Preliminary numerical results validates the proposed approach.

Keywords: Aircraft conflict avoidance, Interval Branch and Bound, Convex relaxation, Affine forms.

1. Introduction

In this work, we deal with the use of deterministic global optimization to solve the aircraft conflict avoidance problem by means of aircraft speed changes. Specifically, we focus on exact global solvers based on Branch and Bound methods. The selected solvers are `Couenne` and `IBBA`, the first based on convex relaxations and the latter based on rigorous interval computations and linear relaxations. Both codes also include interval constraint propagation techniques.

Two aircraft are said *in conflict* when the horizontal distance between them and their altitude distance are smaller than standard safety distances. In this paper, we consider aircraft in their *en route* cruise phase, all at the same altitude, so that only their horizontal distances have to be handled through appropriate *separation constraints*. Aircraft are monitored and suitable separation maneuvers are issued if in the observed time window conflicts may potentially occur. The separation maneuver considered here is aircraft speed deviations, while the directions of motions are kept fixed. Aircraft speed changes may not be able to solve all possible conflict situations, like in the case of two aircraft flying face-to-face; such an approach is however considered very promising to reduce the complexity of air traffic. *Subliminal* speed control is in particular interesting: it is a speed control where aircraft speeds are changed in a very tight range around original speeds, namely between -6% and 3%. In this work we further consider speeds between -12% and 6% of the original speeds as a second range for testing.

The optimization model for aircraft conflict avoidance based on speed changes considered in this work is described in Section 2. In Section 3 we briefly recall on the main characteristics of the two global optimization solvers `Couenne` and `IBBA`. A new convex relaxation based on *affine arithmetic*, to be used within `IBBA`, is proposed in Section 4 for the quadratic convex objective function of the considered model. In Section 5, some numerical tests are discussed. Some conclusions are given in Section 6.

*The authors gratefully acknowledges financial support under grant ANR 12-JS02-009-01 "ATOMIC".

2. MINLP model

In this section, we recall the main elements of a model developed and described in [1]. It is a MINLP model, where aircraft can change their speed once during the observed time window in order to get a conflict-free configuration. The main decision variables are $q_i, i \in A$ (A being the set of aircraft), representing the aircraft speed variations. For each aircraft i, $q_i = 1$ means that there is no change, $q_i > 1$ means that aircraft i accelerates and $q_i < 1$ that it decelerates. The optimization criterion is $\sum_{i=1}^{n}(q_i - 1)^2$, where n is the number of aircraft. The main difficulty is represented by the separation constraints, for each pair of aircraft i and j:

$$\|x_i(t) - x_j(t)\| \geq d \qquad \forall t \in (0, T), \tag{1}$$

where T is the time horizon, d is the minimum required separation distance (5 Nautic Miles), and $x_i(t)$ is the position of aircraft i. Letting $\mathbf{x}_{ij}^r(t) = x_i(t) - x_j(t)$ be $\mathbf{x}_{ij}^{r0} + \mathbf{v}_{ij}^r\, t$, with $\mathbf{x}_{ij}^{r0}$ the relative initial position of aircraft i and j, and $\mathbf{v}_{ij}^r$ be their relative speed, one obtains $||\mathbf{x}_{ij}^{r0} + \mathbf{v}_{ij}^r\, t||^2 \geq d^2\, \forall t \in (0, T)$, and therefore

$$\|\mathbf{v}_{ij}^r\|^2\, t^2 + 2(\mathbf{x}_{ij}^{r0} \cdot \mathbf{v}_{ij}^r)\, t + (\|\mathbf{x}_{ij}^{r0}\|^2 - d^2) \geq 0 \qquad \forall t \in (0, T).$$

By computing the minimum t_{ij}^m of the above quadratic convex function, and by introducing binary variables y_{ij} to check the sign of t_{ij}^m, following the procedure of Cafieri et al. [1], one can reformulate the constraints above by eliminating the dependence on t. More precisely, the following model $(\mathcal{P})$ is obtained (see [1] for details):

$$(\mathcal{P}) \begin{cases} \min\limits_{q_i, t_{ij}^m, y_{ij}} & \sum\limits_{i=1}^{n}(q_i - 1)^2 & \\ s.t. & & \\ & t_{ij}^m \|\mathbf{v}_{ij}^r\|^2 + \mathbf{x}_{ij}^{r0} \cdot \mathbf{v}_{ij}^r = 0, & \forall (i,j) \in \{1, \cdots, n\}^2, i < j \\ & -t_{ij}^m\, (2y_{ij} - 1) \leq 0, & \forall (i,j) \in \{1, \cdots, n\}^2, i < j \\ & -y_{ij} \left(\|\mathbf{v}_{ij}^r\|^2 (\|\mathbf{x}_{ij}^{r0}\|^2 - d^2) - (\mathbf{x}_{ij}^{r0} \cdot \mathbf{v}_{ij}^r)^2 \right) \leq 0, & \forall (i,j) \in \{1, \cdots, n\}^2, i < j \\ & q_i \in \mathbf{q_i} = [\underline{\mathbf{q_i}}, \overline{\mathbf{q_i}}], & \forall i \in \{1, \cdots, n\} \\ & t_{ij}^m \in]-\infty, \infty[, & \forall (i,j) \in \{1, \cdots, n\}^2, i < j \\ & y_{ij} \in \{0, 1\}, & \forall (i,j) \in \{1, \cdots, n\}^2, i < j \end{cases}$$

Remark 1. *The optimization criterion in* $(\mathcal{P})$ *is* **quadratic** *and* **convex**.

3. sBB solvers: `Couenne` and `IBBA`

The deterministic global optimization solvers `Couenne` and `IBBA`, that we consider for the present work, are both based on a *spatial Branch-and-Bound (sBB)* method. Its main characteristics include Bissection/Branching techniques and Constraint Propagation techniques (named HC4 or FBBT) [2, 4]. Further characteristics of `Couenne` and `IBBA` are summarized in Table 1. Note that for computing bounds, `Couenne` uses convex relaxations (denoted by $(\mathcal{P}_{conv})$ in Table 1) [2], while `IBBA` uses linear relaxations (denoted by $(\mathcal{P}_{lin}^{AF})$ in Table 1) based on affine and interval arithmetics [3, 5].

Remark 2. `IBBA` *is numerically reliable (because it is mainly based on interval arithmetic).*

Table 1. Main characteristics of Couenne and IBBA

Couenne (Belotti et al.) [2]	IBBA (Messine and Ninin) [3, 5]
- $(\mathcal{P}) \geq (\mathcal{P}_{conv})$	- $(\mathcal{P}) \geq (\mathcal{P}_{lin}^{AF}) + E_r$ (where E_r is a constant)
- Formal Preprocess	- Interval and Affine Arithmetic
- $w_{ij} = x_i x_j$ and $w_{ii} = x_i^2$ with McCormick constraints.	- $x_i = \mathtt{mid}(\mathbf{x_i}) + \mathtt{rad}(\mathbf{x_i})\varepsilon_i$, with $\mathbf{x_i} = [\underline{\mathbf{x_i}}, \overline{\mathbf{x_i}}], \varepsilon_i \in [-1, 1]$.
- Relaxation $\Rightarrow$ + new variables and constraints.	- Relaxation $\Rightarrow$ same nb of variables and constraints.
- Use of IPOPT.	- Use of CPLEX.

4. Quadratic convex relaxation and branching strategy for IBBA

An idea to improve IBBA code is to keep the quadratic convex criterion instead of linearizing it. This yields a new automatic way to make a convex relaxation of problem $(\mathcal{P})$ by using affine arithmetic. More specifically, the constraints, which are mainly concave, are linearized directly using affine arithmetic, and the criterion is just rewritten by employing a change of variables from q_i to $\varepsilon_i \in [-1, 1]$ as $q_i \longrightarrow \mathtt{mid}(\mathbf{q_i}) + \mathtt{rad}(\mathbf{q_i})\varepsilon_i$, where $\mathbf{q_i} = [\underline{\mathbf{q_i}}, \overline{\mathbf{q_i}}]$, $\mathtt{mid}(\mathbf{q_i}) = \frac{\underline{\mathbf{q_i}} + \overline{\mathbf{q_i}}}{2}$ and $\mathtt{rad}(\mathbf{q_i}) = \frac{\overline{\mathbf{q_i}} - \underline{\mathbf{q_i}}}{2}$. Thus, the criterion of problem $(\mathcal{P})$ becomes:

$$\begin{aligned}\sum_{i=1}^{n}(q_i - 1)^2 \quad &\longrightarrow \sum_{i=1}^{n}\left(\mathtt{mid}(\mathbf{q_i}) + \mathtt{rad}(\mathbf{q_i})\varepsilon_i - 1\right)^2 \\ &= \sum_{i=1}^{n}(\mathtt{rad}(\mathbf{q_i}))^2{\varepsilon_i}^2 + 2(\mathtt{mid}(\mathbf{q_i}) - 1)\mathtt{rad}(\mathbf{q_i})\varepsilon_i + (\mathtt{mid}(\mathbf{q_i}) - 1)^2\end{aligned}$$

The quadratic part of the reformulated criterion reads $\varepsilon^T A_\varepsilon \varepsilon$, with A_ε a diagonal matrix having elements $\mathtt{rad}^2(\mathbf{q_i})$; A_ε is a matrix of size $n \times n$.

Remark 3. *A_ε is positive semidefinite and then the criterion reformulated in terms of ε is kept convex. Note that this property is independent on the selected application.*

Proposition 4. *If $\mathtt{rad}(\mathbf{q_i}) \to 0$ then $A_\varepsilon \to 0$, thus the criterion reformulated in ε tends to be linear.*

Another idea to improve IBBA (and possibly Couenne) is related to the branching strategy. We remark that in the constraints of problem $(\mathcal{P})$, the variables t_{ij}^m and y_{ij} can be deduced from variables q_i. Thus, the idea is to branch only on variables q_i, and to use the HC4-constraint propagation technique to automatically reduce bounds on variables t_{ij}^m and y_{ij}.

5. Numerical solutions

We tested IBBA on 5 problem instances, detailed in [1]; the aircraft are positioned around a circle and all of them fly with the same speed $400NM$ towards the center of the circle. In the two following tables, n represents the number of aircraft and r the radius of the circle, and the time window is about 30 minutes. In Table 2, we solve problem $(\mathcal{P})$ by considering a speed variation $q_i \in [0.94, 1.03]$ (subliminal control), while in Table 3 we consider a larger range, $q_i \in [0.88, 1.06]$. In Table 2, we first report the numerical results in terms of computing time obtained in [1] by using Couenne. In the three last columns of Table 2 and Table 3, we provide the results obtained using IBBA in three different versions: (i)IBBA alone, (ii)IBBA using the quadratic convex relaxation detailed in the previous section, (iii)IBBA using the McCormick's linear relaxations on the quadratic convex program. In the three cases, we use the CPLEX software to solve the linear and convex quadratic programs. The number of iterations is also reported in some cases.

We first note that the gain obtained by using the branching strategy discussed above is very important: when this is not used, IBBA behaves not differently from Couenne (that so could

Table 2. Results obtained with Couenne and 3 versions of IBBA, using $q_i \in [0.94, 1.03]$

n	r	Couenne from [1] time (s)	IBBA time (s)	IBBA+Quad time (s)	IBBA+McCormick time (s)
2	1×10^2	0.11	**0.01**	0.19	**0.01**
3	2×10^2	0.98	**0.23**	1.31	**0.23**
4	2×10^2	8.43	0.89	2.70	**0.88**
5	3×10^2	469.86	41.37	80.79	**40.35**
6	3×10^2	46707.03	**395.87** (67287its)	618.54 (**66761its**)	403.03 (66366its)

Table 3. Results obtained with Couenne and 3 versions of IBBA, using $q_i \in [0.88, 1.06]$

n	r	IBBA time (s)/(#its)	IBBA+Quad time (s)/(#its)	IBBA+McCormick time (s)/(#its)
2	1×10^2	**0.04** / (94)	0.11 / **(89)**	**0.04** / (94)
3	2×10^2	0.43 / (416)	0.74 / **(383)**	**0.41** / (386)
4	2×10^2	4.36 / (2134)	5.82 / **(1915)**	**4.09** / (1930)
5	3×10^2	117.56 / (32151)	136.25 / **(29700)**	**111.57** / (29862)
6	3×10^2	2270.03 / (384552)	2489.33 / **(360233)**	**2188.75** / (361720)

be considerably improved by using this strategy of branching). The impact of the proposed quadratic convex relaxation is actually not very strong on the considered conflict avoidance problem: a reduction in the number of iterations does not correspond to a smaller CPU-time. This is due to the fact that solving a quadratic convex program with CPLEX is of course more expansive than solving a linear one. Therefore, for the considered application the use of the McCormick linear relaxation of the quadratic convex problem provides the most efficient results in terms of CPU-time. Note that the main variables q_i vary within very tight bounds, and therefore the quadratic part of the convex relaxation quickly disappears during the computation (A_ε tends to be 0). As a consequence, when the variable ranges are small, the quadratic relaxation is not efficient; with a larger variable range, $[0.88, 1.06]$ (Table 3), the gain in the number of iterations is indeed more important.

6. Conclusion

We have showed that we can obtain promising results using sBB global optimization solvers such as Couenne and IBBA on an aircraft conflict avoidance model. A suitable branching strategy and a new quadratic convex relaxation based on affine arithmetic, implemented in IBBA, associated with a McCormick linear reformulation, enable to significantly improve the efficiency of the solver.

References

[1] S. Cafieri, and N. Durand. Aircraft deconfliction with speed regulation: new models from mixed-integer optimization. *Journal of Global Optimization*, 58(4):613–629, 2014.

[2] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wachter. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, 24(4):597–634, 2009.

[3] 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*, Springer, 267–292, 2005.

[4] F. Messine. Deterministic Global Optimization using Interval Constraint Propagation Techniques. *RAIRO-Operations Research*, 38(4):277–294, 2002.

[5] J.Ninin, F. Messine, and P. Hansen. A reliable affine relaxation method for global optimization. *4OR - a Quarterly Journal of Operations Research*, 13(3):247–277, 2015.

Proceedings of GOW'16, pp. 153 – 154.

Improving Performance of DIRECT-type Algorithms

Remigijus Paulavičius and Julius Žilinskas

Institute of Mathematics and Informatics, Vilnius University, Akademijos 4, LT-08663 Vilnius, Lithuania,
{remigijus.paulavicius, julius.zilinskas}@mii.vu.lt

Abstract DIRECT (DIviding RECTangles) is one of the most known partitioning-based algorithms for global optimization. Recently several modifications of DIRECT algorithm have been proposed including different partitions, various sampling strategies, and improved balancing of global and local search. In this talk we overview some of them and discuss how the performance of these algorithms can be further improved.

Keywords: Global optimization, DIRECT

Summary

In this talk we consider global optimization algorithms derived from well known DIRECT (DIviding RECTangles) algorithm [6]. It is one of the most known partitioning-based algorithms that balances local and global search in an attempt to efficiently find the global minimizer. The main procedure of the DIRECT algorithm involves evaluation of the objective function at the centers of hyper-rectangles and trisection of potentially optimal ones.

Many modifications of DIRECT have been proposed, most of them use hyper-rectangular partitions, central sampling, and trisection [1, 2, 3, 4, 5, 8, 9, 10, 11, 12]. Trisection is used to enable reuse of the objective function value at the center in descendant subregions, the same center is of the middle descendant subregion. Diagonal approach samples at the endpoints of diagonal [7, 17, 18, 19] and for efficient reuse of sample points use trisection with different directions of diagonals and a special vertex database. Simplicial partitions are used in DIRECT-type DISIMPL algorithm and its modifications [13, 14, 15, 16] where central sampling and trisection (DISIMPL-c) or vertex sampling and bisection (DISIMPL-v) is used. Recently we developed another modification of DIRECT where a bisection of hyper-rectangles and a new sampling strategy on diagonal is used. Bisection can ensure better shapes of hyper-rectangles with smaller variation of sizes in different dimensions than trisection which produces sizes differing by three times. However, to be competitive to other subdivisions, efficient sampling strategy with reuse in descendant subregions is necessary. In this talk we discuss various sampling and subdivision strategies in DIRECT-type algorithms and their impact on the performance.

DIRECT-type algorithms often spend an excessive number of function evaluations exploring suboptimal local minima and delaying discovery of the global minimum. It was shown recently that the significant speed-up may be achieved for the DIRECT-type algorithms either with two-phase technique [13, 18] or combining DIRECT-type algorithms with the local searches [9, 11]. In this talk we present overview of such modifications and results of experimental investigation.

Acknowledgments

This research was funded by a grant (No. MIP-051/2014) from the Research Council of Lithuania.

References

[1] D. Di Serafino, G. Liuzzi, V. Piccialli, F. Riccio, and G. Toraldo. A modified DIviding RECTangles algorithm for a problem in astrophysics. *Journal of Optimization Theory and Applications*, 151(1):175–190, 2011.

[2] D. E. Finkel. *Global Optimization with the* DIRECT *Algorithm*. PhD thesis, North Carolina State University, 2005.

[3] D. E. Finkel and C. T. Kelley. Additive scaling and the DIRECT algorithm. *Journal of Global Optimization*, 36(4):597–608, 2006.

[4] J. M. Gablonsky. *Modifications of the* DIRECT *Algorithm*. PhD thesis, North Carolina State University, 2001.

[5] J. M. Gablonsky and C. T. Kelley. A locally-biased form of the DIRECT algorithm. *Journal of Global Optimization*, 21(1):27–37, 2001.

[6] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Application*, 79(1):157–181, 1993.

[7] D. E. Kvasov, C. Pizzuti, and Ya. D. Sergeyev. Local tuning and partition strategies for diagonal GO methods. *Numerische Mathematik*, 94(1):93–106, 2003.

[8] Q. Liu and W. Cheng. A modified DIRECT algorithm with bilevel partition. *Journal of Global Optimization*, 60(3):483–499, 2014.

[9] G. Liuzzi, S. Lucidi, and V. Piccialli. A DIRECT-based approach exploiting local minimizations for the solution for large-scale global optimization problems. *Computational Optimization and Applications*, 45(2):353–375, 2010.

[10] G. Liuzzi, S. Lucidi, and V. Piccialli. A partition-based global optimization algorithm. *Journal of Global Optimization*, 48(1):113–128, 2010.

[11] G. Liuzzi, S. Lucidi, and V. Piccialli. Exploiting derivative-free local searches in DIRECT-type algorithms for global optimization. *Computational Optimization and Applications*, pages 1–27, 2014.

[12] J. Mockus, R. Paulavičius, D. Rusakevičius, D. Šešok, and J. Žilinskas. Application of reduced-set Pareto-Lipschitzian optimization to truss optimization. *Journal of Global Optimization*, 2015. In press.

[13] R. Paulavičius, Ya. D. Sergeyev, D. E. Kvasov, and J. Žilinskas. Globally-biased DISIMPL algorithm for expensive global optimization. *Journal of Global Optimization*, 59(2-3):545–567, 2014.

[14] R. Paulavičius and J. Žilinskas. Simplicial Lipschitz optimization without the Lipschitz constant. *Journal of Global Optimization*, 59(1):23–40, 2013.

[15] R. Paulavičius and J. Žilinskas. *Simplicial Global Optimization*. SpringerBriefs in Optimization. Springer New York, New York, NY, 2014.

[16] R. Paulavičius and J. Žilinskas. Advantages of simplicial partitioning for Lipschitz optimization problems with linear constraints. *Optimization Letters*, 10(2):237–246, 2016.

[17] Ya. D. Sergeyev. An efficient strategy for adaptive partition of N-dimensional intervals in the framework of diagonal algorithms. *Journal of Optimization Theory and Applications*, 107(1):145–168, 2000.

[18] Ya. D. Sergeyev and D. E. Kvasov. Global search based on efficient diagonal partitions and a set of Lipschitz constants. *SIAM Journal on Optimization*, 16(3):910–937, 2006.

[19] Ya. D. Sergeyev and D. E. Kvasov. *Diagonal Global Optimization Methods*. FizMatLit, Moscow, 2008. In Russian.

Proceedings of GOW'16, pp. 155 – 158.

On the Beam Angle Optimization Problem in IMRT: Combinatorial vs Continuous Optimization

H. Rocha[1,2], J. M. Dias[1,2], T. Ventura[3], B. C. Ferreira[4,5] and M. C. Lopes[3,5]

[1]*FEUC, Universidade de Coimbra, 3004-512 Coimbra, Portugal,* hrocha@mat.uc.pt

[2]*INESC-Coimbra, Coimbra, Portugal,* joana@fe.uc.pt

[3]*IPOC-FG, EPE, Coimbra, Portugal,* tiagoventura@ipocoimbra.min-saude.pt

[4]*ESTSP, Porto, Portugal,* bcf@estsp.ipp.pt

[5]*I3N, Universidade de Aveiro, Aveiro, Portugal,* mclopes@ipocoimbra.min-saude.pt

Abstract The optimal selection of the irradiation directions – beam angle optimization (BAO) – in intensity-modulated radiation therapy (IMRT) treatment planning is seldom done in clinical practice as the corresponding problem is a highly non-convex multi-modal optimization problem. However, for some tumor sites, the advantage of considering optimal radiation incidences is well known. In this paper we present a multistart approach for the continuous BAO problem and compare it with typically used combinatorial approaches.

Keywords: IMRT, Beam angle optimization, Multistart, Derivative-free optimization

1. Introduction

The goal of radiation therapy is to eradicate all cancer cells by delivering a dose of radiation to the tumor volume while attempting to spare the surrounding healthy organs and tissues. Radiation is typically generated by a linear accelerator mounted on a gantry that can rotate along a central axis and can be delivered from almost any angle (direction) around the tumor. The choice of appropriate angle directions to irradiate the tumor volume can enhance a better sparing of the surrounding structures.

The BAO approaches can be separated into two different classes. The first class, prevalent in literature and single solution offered by some commercial treatment planning systems, addresses the BAO problem as a combinatorial optimization problem by considering a discrete sample of all possible beam irradiation directions. The best ensemble of beam angle directions cannot be obtained, in an acceptable computation time, by performing exhaustive searches. Thus, searches are typically guided by a variety of different methods including neighborhood search [1], genetic algorithms [5] or simulated annealing [6]. This combinatorial formulation of the BAO problem leads to an NP hard problem and thus there is no algorithm known able to find, in a polynomial run time, the optimal solution of the combinatorial BAO problem [2]. Another common and successful approach is iterative BAO [2, 3], where beams are sequentially added, one at a time, to a treatment plan, significantly reducing the number of beam combinations. In a second class, the continuous search space of the highly non-convex BAO problem is explored [8, 9, 10]. In this paper, we present a novel approach that explores thoroughly the continuous search space of the highly non-convex BAO problem using a multistart derivative-free framework.

2. BAO continuous formulation

In order to model the BAO problem as a mathematical optimization problem, a measure of the quality of the beam angles ensemble is required. The straightforward measure for driving the BAO problem is the optimal solution of the fluence map optimization (FMO) problem [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], the problem of finding the optimal radiation intensities. Let us consider n to be the fixed number of (coplanar) beam directions, i.e., n beam angles are chosen on a circle around the CT-slice of the body that contains the isocenter (usually the center of mass of the tumor). All continuous $[0°, 360°]$ gantry angles will be considered instead of a discretized sample. Since the angle $-7°$ is equivalent to the angle $353°$ and the angle $367°$ is the same as the angle $7°$, a bounded formulation can avoided. A simple continuous formulation for the BAO problem is obtained by selecting an objective function such that the best beam angle ensemble is obtained for the function's minimum:

$$\begin{aligned} &\min \quad f(\theta_1, \ldots, \theta_n) \\ &\text{s.t.} \quad (\theta_1, \ldots, \theta_n) \in \mathcal{R}^n. \end{aligned}$$

The objective function $f(\theta_1, \ldots, \theta_n)$ that measures the quality of the beam angle ensemble $\theta_1, \ldots, \theta_n$ is the optimal value of the FMO problem for each fixed set of beam angles. For this study, a multicriterial optimization based on a prescription called wish-list [3] was used to address the FMO problem. Nevertheless, as the FMO model is used as a black-box function, the conclusions drawn using this particular formulation of the FMO problem can be extended to different FMO formulations.

3. Multistart approach for the continuous BAO problem

Multistart methods, most of the time, randomly sample the search space. However, for search spaces with peculiar characteristics as the BAO continuous search space, other strategies must be adopted to spread the starting ensembles (solutions) in the search space $[0, 360]^n$ as well as possible. Since the order of the irradiation directions of a beam angle ensemble is irrelevant, the BAO continuous search space have symmetry properties. In terms of optimization, there are many different strategies to address problems with symmetry properties. For the BAO problem, if we keep the beam angles sorted, the symmetry problem is solved and that strategy allows a significative reduction of the search space. In general, for n-beam directions, by sorting the solution's directions, we reduce the search space by 2^n. However, the reduced search space takes a peculiar form, as illustrated in Figure 1 for a BAO search space with 3 beam directions where all possible cases of sorted 3-beam angle ensembles distributed by the four quadrants are displayed as well as the corresponding cubes in the search space $[0, 360]^3$. For n-beam angle ensembles, the total number of (hyper)cubes of the entire search space is 4^n while the number of (hyper)cubes of the reduced search space, which corresponds to the number of possible distributions of n sorted beam angles by the 4 quadrants is the combination with repetition of $\binom{n+4-1}{4} = \frac{(n+4-1)!}{4!(n-1)!}$. For example, for the 7-beam angle optimization problem, the reduced search space has 120 hypercubes compared to 16384 hypercubes for the entire search space. Thus, for a multistart framework, a possible choice of the starting sorted beam angle ensembles is the selection of one ensemble for each of the (hyper)cubes of the reduced search space, guaranteeing that the starting solutions belong to the reduced search space, they are well spread and most importantly they cover well all the reduced search space.

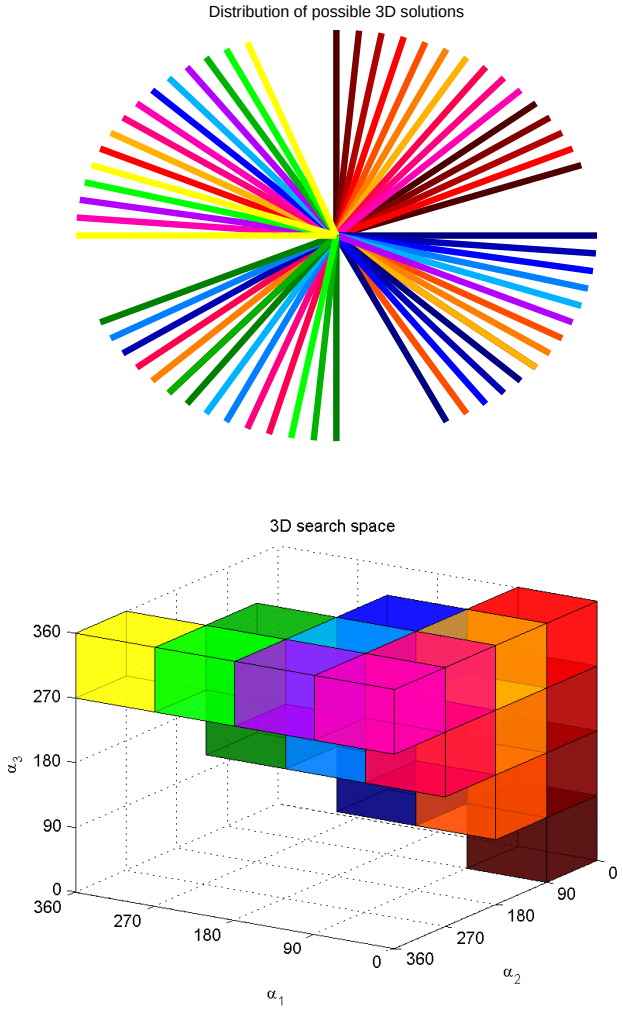


Figure 1. Three beam directions distribution by the four quadrants and the corresponding cubes in the search space $[0, 360]^3$.

4. Computational tests

Multistart is a two phase method that can be designated as global and local phases. In the global phase, the objective function is evaluated for all the starting solutions selected, as described in the previous section. Then, local search procedures seek to improve each of the starting points outcome in a local phase. We have showed in previous works that a beam angle set can be improved in a continuous manner using derivative-free algorithms. Thus, pattern search methods were selected as local procedure to tackle the BAO problem since they require few FMO problem computations and avoid local entrapment [8, 9, 10].

The multistart framework was tested for the optimization of the BAO problem using a set of twenty clinical examples of retrospective treated cases of nasopharyngeal tumors at the Portuguese Institute of Oncology of Coimbra (IPOC). Treatment plans with seven coplanar beam orientations were obtained using the multistart framework and compared with treatments plans obtained using iterative BAO and with the 7-beam equispaced coplanar treatment plans, typically used in clinical practice.

5. Summary

The BAO problem is an extremely challenging continuous global non-convex optimization problem. In clinical practice, most of the time, beam directions continue to be manually selected by the treatment planner in a time-consuming procedure without objective and rigorous criteria. Alternatively, combinatorial optimization strategies have been used to obtain beam directions, including iterative BAO. We propose a completely different methodological approach by addressing the BAO problem as a continuous optimization problem and exploring the reduced search space using a multistart framework. For the nasopharyngeal clinical cases retrospectively tested, the use of optimized beam ensembles obtained by the multistart framework enhanced the quality of the treatment plans obtained and clearly outperform the quality of treatment plans that considered equidistant beam ensembles or beam ensembles obtained using iterative BAO.

References

[1] D.M. Aleman, A. Kumar, R.K. Ahuja, H.E. Romeijn, and J.F. Dempsey. Neighborhood search approaches to beam orientation optimization in intensity modulated radiation therapy treatment planning. Journal of Global Optimization, 42:587–607, 2008.

[2] M. Bangert, P. Ziegenhein, and U. Oelfke. Characterizing the combinatorial beam angle selection problem. Physics in Medicine and Biology, 57:6707–6723, 2012.

[3] S. Breedveld, P. Storchi, P. Voet, and B. Heijmen. iCycle: integrated, multicriterial beam angle, and profile optimization for generation of coplanar and noncoplanar IMRT plans. Medical Physics, 39:951–963, 2012.

[4] D. Craft. Local beam angle optimization with linear programming and gradient search. Physics in Medicine and Biology, 52:127–135, 2007.

[5] J. Dias, H. Rocha, B.C. Ferreira, and M.C. Lopes. A genetic algorithm with neural network fitness function evaluation for IMRT beam angle optimization. Central European Journal of Operations Research, 22:431–455, 2014.

[6] J. Dias, H. Rocha, B.C. Ferreira, and M.C. Lopes. Simulated annealing applied to IMRT beam angle optimization: A computational study. Physica Medica, 31:747–756, 2015.

[7] G.J. Lim and W. Cao. A two-phase method for selecting IMRT treatment beam angles: Branch-and-Prune and local neighborhood search. European Journal of Operations Research, 217:609–618, 2012.

[8] H. Rocha, J. Dias, B.C. Ferreira, and M.C. Lopes. Selection of intensity modulated radiation therapy treatment beam directions using radial basis functions within a pattern search methods framework. Journal of Global Optimization, 57(4):1065–1089, 2013.

[9] H. Rocha, J. Dias, B.C. Ferreira, and M.C. Lopes. Beam angle optimization for intensity-modulated radiation therapy using a guided pattern search method. Physics in Medicine and Biology, 58(9):2939–2953, 2013.

[10] H. Rocha, J. Dias, B.C. Ferreira, and M.C. Lopes. Pattern search methods framework for beam angle optimization in radiotherapy design. Applied Mathematics and Computation, 219(23):10853–10865, 2013.

Proceedings of GOW'16, pp. 159 – 162.

Parallel Implementation of GLOBAL with Applications to Nanophotonical Detector Development*

Tibor Csendes[1], Balázs Bánhelyi[1], Mária Csete[2], Dániel Zombori[1], Gábor Szabó[2] and András Szenes[2]

[1]*Institute of Informatics, University of Szeged, H-6720 Szeged, Árpád tér 2, Hungary,* csendes@inf.szte.hu

[2]*Department of Optics and Quantum Electronics, University of Szeged, H-6720 Szeged, Dóm tér 9, Hungary,* mcsete@physx.szte.hu

Abstract We present a new, Java based implementation of the GLOBAL algorithm to enable it to run more efficiently in multicore computers. The direct motivation for this step were two applications requiring massive amounts of computation from the fields of surgical operation design and nanophotonical detector development.

Keywords: GLOBAL, Parallel implementation, Application, Java

1. Introduction

In recent years we were involved mostly in the solution of two sets of optimization application problems: optimal nanophotonical detector design [5] and surgical operation planning [3]. These were computationally demanding tasks that required a proper implementation allowing a quicker solution on everyday computers. The applied optimization platform was the GLOBAL algorithm [1, 4].

GLOBAL is a multistart procedure, that applies local search methods for finding local minimizer points. Two local search algorithms can be selected: the first is UNIRANDI, a direct search technique. It is a random walk method, that does not assume differentiability of the objective function, and requires only a subroutine for the calculation of the objective function value. The other method is BFGS, that is readily available in Matlab (even without the optimization packages). This assumes a smooth objective function, although it requires again only a subroutine for the calculation of the objective function value.

The framework multistart algorithm assumes that the relative size of the region of attraction of the global minimizer point(s) is not negligible, i.e. say larger than 0.00001. The GLOBAL procedure is in a schematic description:

Step 1: Draw N points with uniform distribution in X, and add them to the current cumulative sample C. Construct the transformed sample T by taking the γ percent of the points in C with the lowest function value.

Step 2: Apply the clustering procedure to T one by one. If all points of T can be assigned to an existing cluster, go to Step 4.

Step 3: Apply the local search procedure to the points in T not yet clustered. Repeat Step 3 until every point has been assigned to a cluster.

Step 4: If a new local minimizer has been found, go to Step 1.

*The research was supported by National Research, Development and Innovation Office-NKFIH through project "Optimized nanoplasmonics" K116362. Mária Csete acknowledges that the project was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences. Gábor Szabó acknowledges the support of the Hungarian Academy of Sciences.

Step 5: Determine the smallest local minimum value found, and stop.

2. A motivating problem from nanophotonical design

We intend to find favorable structures of tiny light sensors for sophisticated communication via encoded information. The parameters of the optimal structure of the detectors sensing quantum information holding infrared single-photons depend much on the wave length. Different kinds of plasmonic structures have relevant properties for controlling the nanophotonical properties of integrated detectors. We encounter many diverse optimization problems studying these integrated structures.

COMSOL is a Java based Matlab tool for multi-physics simulation (Figure 1). The RF module of the COMSOL is a well accepted tool for investigating the sensitivity of detectors in nanophotonics. It is cooperating well with other platforms such as Matlab, Java, Excel etc. Also a server-client type websocket based communication is available between the applications and the simulation engine. COMSOL can be driven by LiveLink (MATLAB):

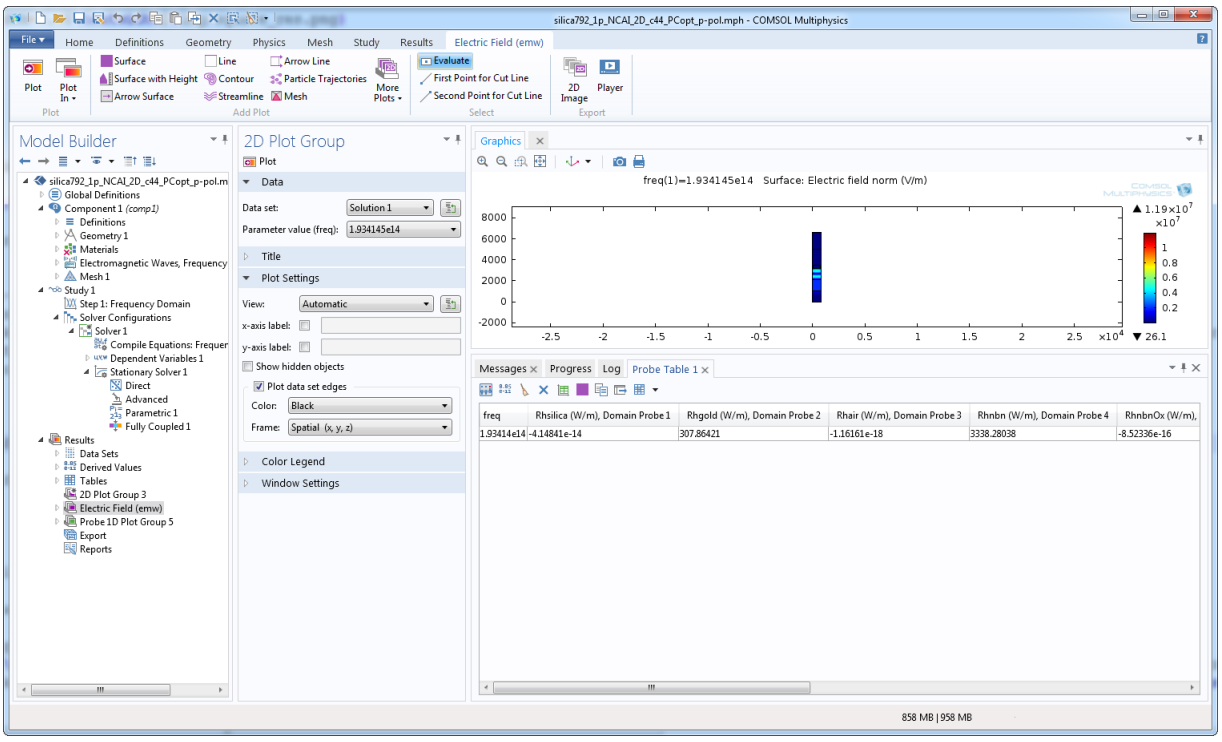


Figure 1. A typical COMSOL control page.

```
addpath '/n/comsol/COMSOL44/mli';
import com.comsol.model.*
import com.comsol.model.util.*
mphstart(pcName,2036);
model=mphload(mphFileName);
model.sol('sol1').run;
mphmean(model,'var4',1);
```

But COMSOL can also be controlled by LiveLink (JAVA):

```
System.setProperty("cs.root",
"C:\\Program Files\\COMSOL\\COMSOL44");
ModelUtil.initStandalone(false);
ModelUtil.connect(pcName, 2036);
model = ModelUtil.loadCopy("Model",mphFileName);
model.sol("sol1").run();
model.result().table("tbl1").getTableData(true)[0][0]
```

We studied the following optimization problems. What is the maximal detection efficiency for a given type detector? The available values in the literature are around 93%. What is the maximal detection efficiency for e.g. different polarization photons? The information carrying type photon should be absorbed with enhanced efficiency compared to other type photons. Can we improve the contrast without a major decrease in the detection efficiency?

The code was earlier available in Fortran, C, and Matlab. The reasons for the Java implementation:

- More efficient optimization, an object oriented version is also welcome.
- This language supports our parallelization aims as well, both the multi-machine and the multi-core versions (CUDA was neglected in the first row).
- Fits well our present problems.

The object oriented GLOBAL has three main components: sample generation, clustering, and local minimization. All the three are equipped with an interface class. All of them can be applied in a stand-alone way, other combinations can be formed. They have build methods.

3. Experimental results

Some representative efficiency results:
Nano-cavity-array-integrated detector (NCDAI)

Optimization method	Period (nm)	Absorption
Original (792, optA)	792.46	78.22%
GLOBAL (792, optC)	792.46	79.52%

Nano-cavity-double-deflector-array-integrated detector (NCDDAI)

Optimization method	Period (nm)	Absorption
Original (792, optA)	792.46	86.84%
GLOBAL (792, optC)	792.46	89.90%

Nano-cavity-trench-array-integrated detector (NCTAI)

Optimization method	Period (nm)	Absorption
Original (792, optA)	792.46	89.92%
GLOBAL (792, optC)	792.46	92.07%
GLOBAL (500-600, optG)	600.00	94.49%
GLOBAL (1000-1100, optI)	1056.24	95.05%

Optimized contrast results are given in the next table with fixed absorption rates (NCAI stands for Nano-cavity-array-integrated, NDAI for Nano-deflector-array-integrated):

System	Original	Optimized	90%	93%
NCAI	146.8800	219.9831	219.9679	199.461
NDAI	1.3420E+03	6.0424E+10	6.3289E+05	3.2113E+03
NCDDAI	1.9468E+03	4.6787E+11	2.1718E+08	1.9135E+07
NCTAI	49.9800	366.4346	69.8521	69.9753

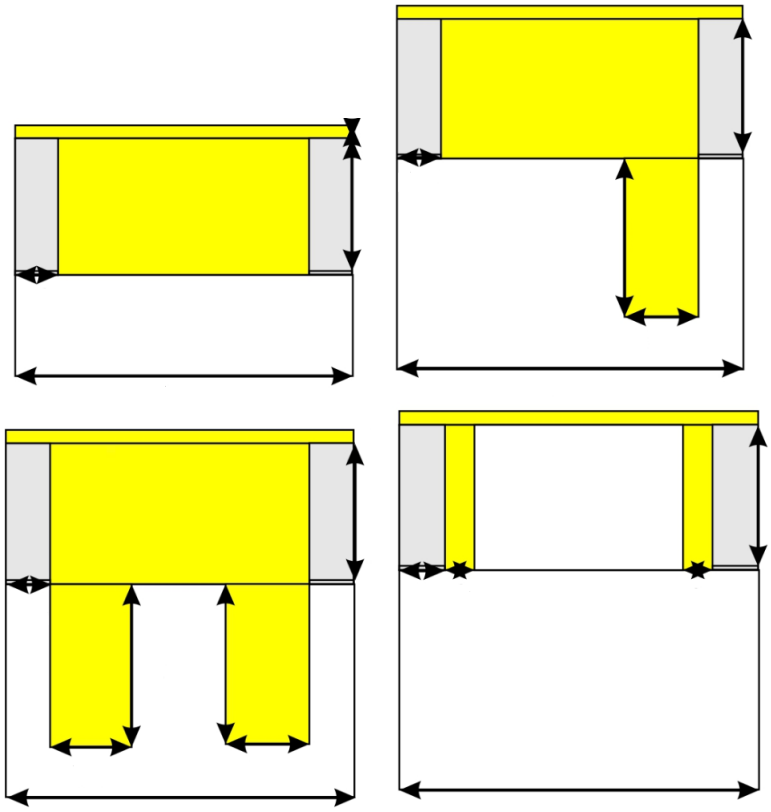


Figure 2. Detector types: NCAI, NCDAI, NCDDAI, and NCTAI, respectively. The arrows indicate the parameters to be optimized.

In the talk we shall report detailed computational test results on the Java version GLOBAL code, on the speed up achieved, and also on new, improved results on the surgical operation design problem [3] addressed in the last Global Optimization Workshop [2].

References

[1] T. Csendes. Nonlinear parameter estimation by global optimization – efficiency and reliability. *Acta Cybernetica* 8:361–370, 1988.

[2] T. Csendes and I. Bársony. Optimization in surgical operation design. Proceedings of MAGO 2014, Malaga, 145-148, 2014.

[3] T. Csendes, I. Bársony, and I. Szalay. Optimization in surgical operation design. Proceedings of the ICAI Conference, Eger, 2014.

[4] T. Csendes, L. Pál, J.O.H. Sendín, and J.R. Banga. The GLOBAL optimization method revisited. *Optimization Letters* 2:445–454, 2008.

[5] M. Csete, G. Szekeres, A. Szenes, B. Bánhelyi, T. Csendes, and G. Szabó. Optimized superconducting nanowire single photon detectors to maximize absorptance. *Progress In Electromagnetics Research B*, 65:81–108, 2016.

Proceedings of GOW'16, pp. 163 – 166.

On Regular Simplex Refinement in Copositivity Detection*

J.M.G. Salmerón[1], P. Amaral[2], L.G. Casado[1], E.M.T. Hendrix[3] and J. Žilinskas[4]

[1] *University of Almería,* leo@ual.es, josemanuel@ual.es

[2] *Universidade Nova de Lisboa,* paca@fct.unl.pt

[3] *Universidad de Málaga,* eligius@uma.es

[4] *Vilnius University,* julius.zilinskas@mii.vu.lt

Abstract Over the last decades checking copositivity of matrices by simplicial subdivision of the unit simplex has made a big progress. Recently it has been showing that surprisingly the use of regular simplicial subdivisions may have some advantage over traditional iterative bisection of simplices. In this contribution we pose the question whether regular subdivisions may provide opportunities in copositivity testing.

Keywords: Copositive matrix, Simplicial refinement, Unit simplex, Regular refinement

1. Copositivity Detection by Simplicial Refinement

Copositivity plays an important role in combinatorial and quadratic optimization. Setting up a linear optimization problem over the copositive cone leads to exact reformulations of combinatorial problems, for example, maximum clique [1]. Let $\omega(G)$ be the clique number of a graph G, E the identity matrix, $t \in \mathbb{N}$ a variable, and $Q = E - A_G$ a matrix derived from the adjacency matrix A_G of the graph G. The goal of this copositive programming problem is to find the smallest value of t such that $tQ - E$ is copositive, i.e. it is in set $\mathcal{C}$ of copositive matRices,

$$\omega(G) = \min\{t : \ tQ - E \in \mathcal{C}\}.$$

An $n \times n$ real symmetric matrix A is called copositive if $x^T Ax \geq 0$ for all $x \in \mathbb{R}^n_+$, where $\mathbb{R}^n_+ := \{x \in \mathbb{R}^n : x_i \geq 0 \text{ for all } i\}$ denotes the non-negative orthant. This means a matrix A is copositive if $e_i^T Ae_j = a_{ij} \geq 0$ for all i, j. Thanks to the thesis investigation of Bundfuss [2], it is known that iterative simplicial refinement of the unit simplex S, this sufficient condition eventually becomes closer and closer to a necessary condition. The unit simplex is defined as

$$S = \{x \in \mathbb{R}^n \mid \sum_{j=1}^{n} x_j = 1; \ x_j \geq 0, \ \forall j\}. \tag{1}$$

Let $v_1, \ldots, v_n$ denote the vertices of sub-simplex Δ in the refinement, then $v_i^T Av_j \geq 0$ shows that A is copositive over Δ. This condition gives rise to Algorithm 1 for copositivity detection. First of all, it should work with an accuracy ϵ in case $\min_{x \in S} x^T Ax = 0$. This means that the algorithm in fact presents a certificate of what could be called ϵ-copositivity, $x^T Ax \geq -\epsilon$.

The algorithm starts with the unit simplex, i.e. the vertices are the unit vectors $e_1, \ldots, e_n$. Simplices are subdivided until either the candidate list is empty, which certificates that A is copositive, or alternatively a point $x \in S$ is found for which $x^T Ax < 0$, which means

*This work has been funded by grants from the Spanish Ministry (TIN2015-66688) and Junta de Andalucía (P11-TIC-7176), in part financed by the European Regional Development Fund (ERDF) and by Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) through PEst-OE/MAT/UI0297/2011 (CMA)

Input: A: matrix, ϵ: accuracy.
$\Lambda := \{S\}$ `// Set of sub-simplices in tree`
certificate = true
while $\Lambda \neq \emptyset$ **do**
  Extract a simplex Δ from Λ and refine into a set Ω of sub-simplices
  for *all* $\Delta \in \Omega$ **do**
    Determine $L(\Delta)$
    if *found a point* x *with* $x^T Ax < 0$ `// A not copositive`
    then
      certificate = false goto 12
    if $L(\Delta) > -\epsilon$ **then**
      Store Δ in Λ.
return *certificate*

Algorithm 1: CoposTest(A, ϵ, certificate)

that A is not copositive. Like in branch and bound algorithms, it defines a lower bound $L(\Delta) = \min_{i,j} v_i^T A v_j \leq \min_{x \in \Delta} x^T Ax$ for each evaluated simplex Δ, such that each simplex Δ for which $L(\Delta) \geq -\epsilon$ can be discarded. Therefore, the search tree is pruned at the nodes corresponding to those simplices. Implicitly, the algorithm has a selection rule of which simplex to refine further first. Usually a depth first rule or best first rule based on the lowest lower bound value is used to prevent the tree to fill up all memory space in the computer.

Copositivity certification by simplicial refinement requires much more computation than verifying non-copositivity of a matrix. More recent implementations [4] that also include parallel computing show that verifying non-copositivity of matrices can go to a size up to several thousands. However, certifying copositivity of a copositive matrix is limited to a size up to 22 in a reasonable time. In the implemented simplicial partitioning algorithms [2, 4] the edge (i, j) with the most negative value $v_i^T A v_j$ is subdivided aiming to find a point with negative $v^T Av$ quickly in the case of a non-copositive matrix. The question is whether one could find a way to make certification of copositivity faster, although verification of non-copositivity is slower. This would also make copositive programming by simplicial partitioning faster.

Moreover, it is known that branch and bound techniques like Algorithm 1 require an efficient management of memory storage of the underlying search tree and its nodes. In order to make efficient use of memory and to facilitate the computational requirement, this contribution investigates whether the recently developed regular simplicial covering can be of help to achieve the aim of efficient copositivity detection.

2. Regular subdivision of the unit simplex by 2USC

For several years we investigated the concept of Uniform Simplex Cover (USC) where the simplex is covered by equally sized, equally oriented overlapping sub-simplices and we analysed its characteristics. Initially, the concept did not look very appealing compared to traditional bisection, as the refinement leads to overlapping sub-simplices, i.e. it is not a partition. Only recently has been found that in some aspects, where the complete tree is generated, like when the matrix is copositive, the method may have some advantage over bisection, see [3]. To express the idea of equally sized and oriented simplices we introduce the following concepts.

Each simplex Δ has a center c and a radius r which define its vertex matrix as

$$V = c\mathbf{1}^T + rD, \tag{2}$$

where $\mathbf{1}$ is the all ones vector and $E = (e_1, \ldots, e_n)$ represents the identity matrix. $D = (d_1, \ldots, d_n) = E - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ is a symmetric matrix with the directions from the center towards

the regular simplex vertices. Notice that the center of the unit simplex S is $c = \frac{1}{n}\mathbf{1}$, whereas its radius, that is the step size relative to the deviation matrix D is 1.

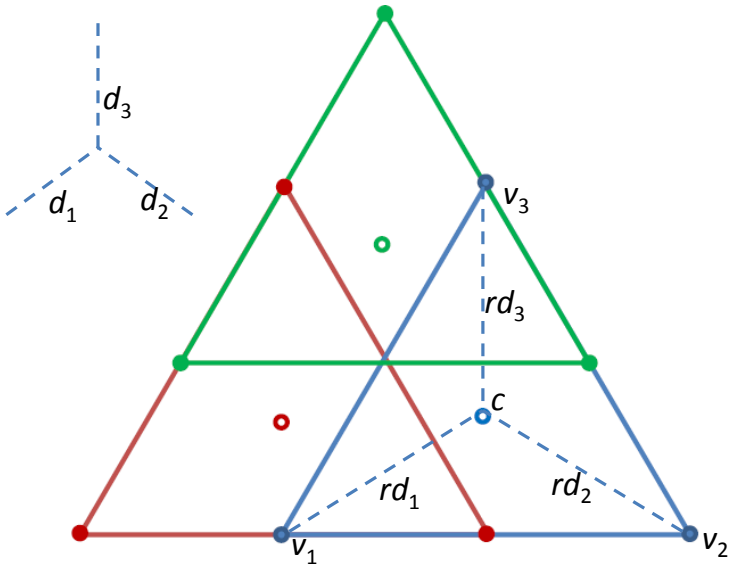


Figure 1. Uniform covering of a simplex by 2USC. The orientation provides fixed directions d_i, $i = 1, \ldots, n$.

The idea of 2USC, where each edge is covered by 2 overlapping simplices, is illustrated in Figure 1. Due to the orientation, the set of direction vectors in D is always the same. The reduction of the radius from parent simplex to a child in the corresponding search tree is at least $\frac{n-1}{n}$. Storage of the simplex in a tree requires only to store the center c and the radius r.

We now focus on the consequence of using such a refinement for the determination of a lower bound $L(\Delta)$ in Algorithm 1. For two vertices v_i and v_j of Δ we have

$$v_i^T A v_j = (c + rd_i)^T A(c + rd_j) = c^T Ac + rc^T A(d_i + d_j) + r^2 d_i^T A d_j. \tag{3}$$

For evaluating the lower bound, we have the evaluation $c^T Ac$ in c and, as will be shown, the minimum on a linear and quadratic term in the radius r with vectors b_{ij} and constants f_{ij}:

$$L(\Delta) = c^T Ac + \min_{i,j}\{rc^T b_{ij} + r^2 f_{ij}\}. \tag{4}$$

If the evaluated $c^T Ac$ is negative, non-positiveness has been proven in line 7 of the algorithm. If the term is nonnegative, the lower bounding term becomes of interest. Notice that term becomes smaller when we go deeper into the search tree where the radius r becomes small. The data for the evaluation of the lower bound is matrix A with columns a_i and average row (and column) vector $\overline{a} = \frac{1}{n}\sum_{i=1}^{n} A_{ij}$. Given that the direction vectors have the shape $d_i = e_i - \frac{1}{n}\mathbf{1}$, one can write the necessary coefficients to evaluate the lower bound in (4) as

$$b_{ij} = A(d_i + d_j) = a_i + a_j - 2\overline{a} \tag{5}$$

and

$$f_{ij} = d_i^T A d_j = A_{ij} - \overline{a}_i - \overline{a}_j + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}. \tag{6}$$

These coefficients can be calculated from the matrix A at the start of the algorithm preventing vertex evaluation and storage.

3. Conclusion

Copositivity detection using simplicial refinement is a computational challenge. Using regular simplex refinement with uniformly oriented simplices prevents evaluating and storing vertex evaluations.

References

[1] I. M. Bomze, M. Dür, E. de Klerk, C. Roos, A. J. Quist, and T. Terlaky. On copositive programming and standard quadratic optimization problems. *Journal of Global Optimization*, 18(4):301–320, 2000.

[2] S. Bundfuss and M. Dür. Algorithmic copositivity detection by simplicial partition. *Linear Algebra and its Applications*, 428(7):1511–1523, 2008.

[3] B. G.-Tóth, E. M. T. Hendrix, L. G. Casado, and I. García. On refinement of the unit simplex using regular simplices. *Journal of Global Optimization*, 64(2):305–323, 2016.

[4] J. Žilinskas and M. Dür. Depth-first simplicial partition for copositivity detection, with an application to maxclique. *Optimization Methods and Software*, 26(3):499–510, 2011.

Proceedings of GOW'16, pp. 167 – 170.

A Numerical Analysis of Flocking Based Distributed Stochastic Optimization Algorithms On Non-Convex Problems

Charles Hernandez and Alfredo Garcia

University of Florida, Gainesville, FL, cdhernandez@ufl.edu, alfredo.garcia@ufl.edu

Abstract We use numerical experiments to analyze the performance of a distributed stochastic optimization algorithm on a variety of non-convex problems. The algorithm consists of several independent threads each implementing a gradient descent procedure which is perturbed in a similar manner to mathematical models of flocking, swarming and other group formations. Through the experiments we learn how to effectively tailor the algorithm parameters to the form of the problem and identify where it can outperform traditional approaches.

Keywords: Stochastic optimization, Distributed optimization, Gradient descent, Non-convex, Flocking, Numerical analysis

1. Introduction

In recent years significant research attention has been focused on distributed optimization algorithms and their applications [4,6,9]. Though many disparate problems and algorithms have been analyzed, the vast majority of the analysis has focused on convex optimization. On the other hand, the majority of real life problems tend to be non-convex. Accordingly, there is significant benefit that can be derived by investigating these non-convex problems and finding effective algorithms to solve them. To this end we continue the development of the algorithm proposed in [8] and analyze its performance on a number of general non-convex problems. In addition, we seek to characterize its behavior as a function of the algorithm's parameters both in stationary and time-varying implementations. Lastly, we attempt to obtain an optimization scheme that can effectively be applied to real world problems where the shape of the optimization surface and the characterization of the noise are not known.

2. The Algorithm

We consider a distributed scheme for stochastic optimization with $N > 1$ independent computing threads each implementing a stochastic gradient algorithm wherein the gradient is further perturbed by a *flocking* potential with repulsive and attractive terms (a function of the relative distance between threads). Hence, the updating of individual threads is coupled in a similar manner to mathematical models of flocking, swarming and other group formations (see [2]). In our recent work (see [8]) we have shown that this coupling endows the scheme with a reduced sensitivity to noise. Noise realizations that induce trajectories differing too much from the group average are likely to cancel out with the attractive term which aims to maintain cohesion. This noise reduction property is fundamentally different to that attained by other *centralized* gradient estimation techniques. For example, with $N > 1$ samples (which could be obtained in parallel), a better estimate for gradient can be computed in a centralized manner (see for instance, [1,7] for a survey of gradient estimation techniques and [3,5] for recent applications in machine learning.). In contrast, for the flocking-based approach, each thread only needs to keep track of the current location of other threads. A flocking-based approach to stochastic optimization provides a novel way of reducing the effects of noise and

therefore provides a basis for a novel distributed computing approach to stochastic optimization.

To illustrate, consider the minimization of the function $f(x) = \log(||x||^2+1)$ where $x \in \mathbb{R}^2$. The unique optimal solution is $x^* = 0$. Suppose the gradient $\nabla f(x)$ is observed with noise so that the basic iteration in a traditional stochastic gradient descent algorithm can be written as:

$$x_{k+1} = x_k + \rho_k(-\nabla f(x_k)) + \xi_k)$$

where $\rho_k > 0$ is the step size and the collection $\{\xi_k : k > 0\}$ is the noise. Now we introduce an additional perturbation to the gradient so that the basic iteration for a thread i is:

$$x_{i,k+1} = x_{i,k} + \rho_k(-\nabla f(x_{i,k}) + \xi_{i,k} + \sum_{j=1,j\neq i}^{N} g(x_{i,k} - x_{j,k}))$$

where $g(x_{i,k} - x_{j,k})$ is the *flocking potential* between threads i and j, i.e. a combination of repulsive and attractive "forces" depending upon the relative distance $x_i - x_j$. The performance of the flocking-algorithm can be seen in Figure 1 ($N = 15$, $\xi_{i,k} \sim N(0, 4^2)$, $\rho_k = .3$ and $g(x) = -x[a - b/x^2]$ where $a = .2, b = .4$).

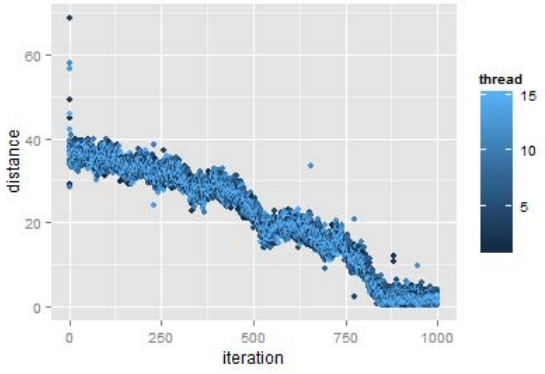


Figure 1a. Distance from the optimal solution over time for each thread

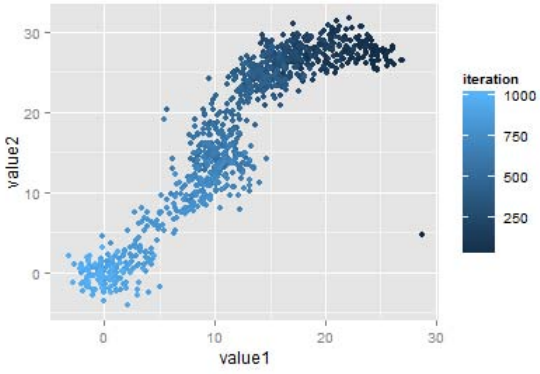


Figure 1b. Path of a single thread through the solution space

3. Numerical Experiments

The numerical experiments will be performed by testing several implementations of the flocking algorithm on a variety of non-convex optimization problems. The process will consist of several stages in order to first build intuition about the relationship between the algorithm's parameters and its behavior and then using this intuition to develop effective configurations for different non-convex problems.

3.1 Experiment Parameters

As referenced in Section 2, the algorithm has 3 components which can be modified to alter its behavior.

The first alterable component is the time scale or step size of the algorithm, which governs the rate at which the various threads move around the solution space. Previous analysis has shown that smaller step-sizes will result in the algorithm being less sensitive to noise, while larger step sizes allow the algorithm to converge to the optimal solution quicker. However, if the step size is made too large, it can result in the algorithm failing to converge at all, even among convex problems where we use many threads (see Figures 2, 3, 4). Past analysis has

only used stationary step sizes while our numerical experiments will analyze certain algorithmic implementations in which the step size changes during the algorithm's execution or differ between threads.

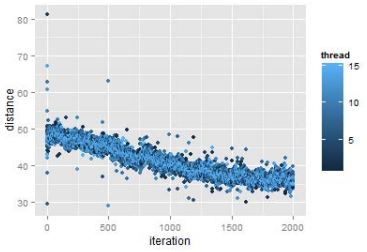


Figure 2. By plotting the distance of each thread to the optimal solution for the flocking algorithm when the step size was set to .1, we can see that the mass of threads continuously improves its position with little deviation.

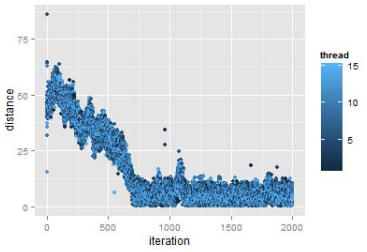


Figure 3. In this plot, where the step size of the algorithm was set to .7, we can see that the mass of threads converged to an optimal solution much quicker but was also subject to greater deviations.

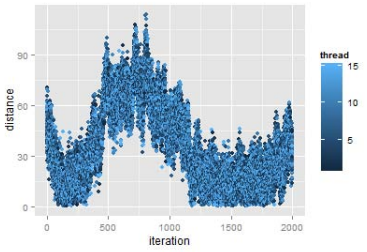


Figure 4. When the step size was set to 2, though the mass of threads does find the optimal solution at times, it does not stay there due to the noise

The second component is the number of distinct threads used in the algorithm. Since changing this parameter would be associated with a significant cost increase in a real-world setting, it can only be optimized within that setting. As a result we will leave this parameter constant throughout our experiments.

The final parameter is the algorithm's flocking potential which is where the majority of our analysis will take place. The flocking potential has significant room for variation in that it can be any function of the distance between the threads. Past analysis has only investigated the use of a stationary flocking potential for all threads, while this paper will look at a variety of potentials that change during execution.

3.2 Performance Evaluation

Because our numerical experiments have various objectives, we will need to characterize the algorithm's performance in several ways.

For algorithms which lack a stopping criterion, we will primarily be interested in 2 metrics, the level of convergence (mean distance between threads) of the various threads, and the best solution found among all threads. Since these implementations lack a stopping criterion we will simply evaluate their performance by artificially stopping the algorithm after some common number of iterations. By simulating the optimization process several times, we will be able to obtain a representative measure of each of the performance metrics.

For algorithms which have a stopping criterion, we will characterize its performance by the amount of time it took to reach the stopping point as well as the objective value of the final solution. As before we repeat the numerical experiments several times in order to minimize the error in our performance analysis.

3.3 Initial Analysis

The first step will be to identify what type of flocking potential engenders different behaviors of the algorithm. It will be important to understand under what conditions will the threads converge to a single solution or diverge and escape local minima. In addition, it will be important to gain insight into how the step size changes the algorithm's behavior perhaps relative to the topography of the objective function.

3.4 Non-stationary Algorithm Design

The next step will be to use the understanding gained from the initial analysis in order to design algorithms which more intelligently search the solution space by changing the algorithm parameters over time. The goal is to find an implementation of the algorithm which at first uses the divergence of the threads to find good global solutions before changing behavior and converging the various threads towards the best one. Once designed we will test this implementation to verify our intuition about the optimal sequencing of algorithm behaviors.

3.5 Inter-Thread Flocking Variability

The next step will look at more exotic options where the flocking potential or step size is no longer the same among all the threads. This will, for example, allow implementations where the attractive flocking potential is increased among threads that have realized better solutions, allowing the algorithm to quickly converge when significantly better than average solutions are found. Another example could be to make threads insensitive to the attractive potential of any thread at a worse objective solution. Alternatively we could design a version where the threads implement one or more of the above behaviors and analyze how a mixture of different thread types could work.

3.6 Abstraction to Real World Problems

The final step will be to see what implementations work well when applied to problems where nothing is known about the noise or the objective function. As mentioned above, previous analysis has shown that, for example, a bad step-size can prevent your algorithm from converging to an optimal solution even among convex problems, as a result, we will need to make sure that the algorithm's parameters vary in such a way that a bad initial set of parameters does not indefinitely prevent the algorithm from working as expected. These implementations will have the greatest potential for actual real world application when very little is known about the shape of the actual problem.

4. Summary

In summary, we are looking to take distributed stochastic optimization algorithms and gain the insight necessary to understand how they could be most effectively used to solve non-convex real world problems.

References

[1] M. Fu. Stochastic Gradient Estimation, *Handbook of Simulation Optimization*, Springer, 2015.

[2] V. Gazi, and K. Passino. *Swarm Stability and Optimization*, Springer, 2012.

[3] P. Hennig. Fast probabilistic optimization from noisy gradients. *Proceedings of The 30th International Conference on Machine Learning*, pp. 62-70, 2013.

[4] I. Necoara, Y. Nesterov, and F. Glineur. A random coordinate descent method on large optimization problems with linear constraints. *Technical Report, University Politechnica Bucharest*, pp. 1-25, 2011.

[5] N. L. Roux, M. Schmidt,and F. R. Bach. A Stochastic Gradient Method with An Exponential Convergence Rate for Finite Training Sets. *Advances in Neural Information Processing Systems*, 2663–2671, 2012.

[6] A. H. Sayed. Adaptation, learning, and optimization over networks. *Foundations and Trends in Machine Learning*, 7(4-5):311–801, 2014.

[7] J. Spall. *Introduction to Stochastic Search and Optimization*, Wiley, 2003.

[8] P. Shi, A. Garcia, and Z. Lin. Noise reduction by swarming in social foraging. *IEEE Transactions on Automatic Control*, 99, pp. 1, 2016.

[9] Z. J. Towfic, and A. H. Sayed. Adaptive penalty-based distributed stochastic convex optimization. *IEEE Transactions on Signal Processing*, 62(15):3924–3938, 2014.

Proceedings of GOW'16, pp. 171 – 174.

A New Global Optimization Algorithm for Diameter Minimization Clustering

Daniel Aloise[1] and Claudio Contardo[2]

[1]*Universidade Federal do Rio Grande do Norte, Natal, Brazil,* daniel.aloise@gerad.ca

[2]*Université du Québec à Montréal, Montreal, Canada,* contardo.claudio@uqam.ca

Abstract We introduce an iterative algorithm for the solution of the diameter minimization clustering problem. Our algorithm is based upon two observations: (i) subsets induce lower bounds on the value of the optimal solution of the original problem; and (ii) there exists a subset whose optimal clustering has the same value as that of the original problem. Computational experiments show that our algorithm takes two seconds or less to solve all tested problems, including one with 5000 points which was previously solved in almost a minute by the current state-of-the-art constraint programming method found in the literature.

Keywords: Clustering, Optimization

1. Introduction

Clustering is a powerful tool for automated analysis of data. It addresses the following problem: given a set of entities find subsets, called clusters, which are homogeneous and/or well separated. Clustering is ubiquitous, with applications in the natural sciences, psychology, medicine, engineering, economics, marketing and other fields [7, 8].

Usually in clustering, we are given a set $O = \{o_1, \ldots, o_n\}$ of n objects that must be partitioned into a set of $P = \{C_1, \ldots, C_k\}$ clusters such that: (i) $C_j \neq \emptyset, \quad \forall j \in \{1, \ldots, k\}$; (ii) $C_i \cap C_j = \emptyset, \quad \forall i, j \in \{1, \ldots, k\}$ with $i \neq j$; and (iii) $\bigcup_{j=1}^{k} C_j = O$, while maximizing or minimizing a function f which defines how homogeneity and separation are expressed in the clusters to be found.

Among the many available criterion used in cluster analysis, the *diameter minimization* (DM) is the most natural to express homogeneity. Actually, the classic book of Garey and Johnson [5] refers to the DM clustering problem as "the clustering problem". The diameter (D_j) of a cluster $j \in \{1, \ldots, k\}$ is defined as:

$$D_j = \max\{d_{ii'} : 1 \leq i < i' \leq n, o_i, o_{i'} \in C_j\}, \tag{1}$$

where $d_{ii'}$ represents the dissimilarity between objects i and i'. The diameter minimization problem is then expressed as $\min \max_{1 \leq j \leq k} D_j$.

A popular heuristic for DM is the complete-linkage hierarchical clustering algorithm [9]. Complete-linkage merges at each step of its hierarchical construction the two subsets for which the maximum dissimilarity between their objects is minimum. The desired solution is obtained by cutting the resulting tree at k clusters. However, the method seldom finds optimal solutions as reported by [2, 6]. Regarding exact methods, Hansen and Delattre [6] explored the relationship between DM and graph-coloring to devise a branch-and-bound method to solve problems of non-trivial size. Brusco and Stahl [3] proposed a backtracking algorithm, denoted Repetitive Branch-and-Bound Algorithm (RBBA), that branches by assigning each object to one of the possible k clusters. More recently, Dao et al. [4] proposed a constraint programming approach for DM whose computational results outperformed the previous exact

approaches found in the literature. In particular, the method obtained the optimal partition regarding the DM criterion for a real-world dataset with 5000 objects grouped into three clusters.

2. Preliminaries and notation

Let V be the complete set of nodes (objects), of size n. We let $E = \{\{u, v\} : u, v \in V, u < v\}$ be the set of edges linking all nodes in V, of size $m = n(n-1)/2$. For each edge $e = \{u, v\} \in E$, we denote by d_e the dissimilarity between nodes u and v. The dissimilarity matrix of V is denoted by $D(V) = [d_e]_{e \in E}$. A feasible solution of the DM is a partition $\mathcal{C} = (C_j)_{j=1}^{k}$ of V. The diameter of a cluster C_j is denoted by $d(C_j)$ and is equal to $\max\{d_e : e = \{u, v\} \in E, u, v \in C_j\}$. The cost associated with $\mathcal{C}$ is denoted by ω and is equal to $\max\{d(C_j) : j = 1, \ldots, k\}$. Our algorithm to solve the DM relies on the following results and hypothesis.

Lemma 1. *Let $U \subset V$ be a subset of V, and let $D(U)$ be its dissimilarity matrix. The optimal solution of the DM on $(U, D(U))$ provides a lower bound on the optimal solution of the DM on $(V, D(V))$.*

Proof. The proof is by induction on the size of $V \setminus U$. Let $|V \setminus U| = 1$. Let ω^V, ω^U be the values of two optimal solutions of the DM on $(V, D(V))$ and on $(U, D(U))$, respectively. Let $v^* = V \setminus U$. If $\omega^U > \omega^V$, the one can remove v^* from the cluster containing it in $(V, D(V))$ and get a solution for $(U, D(U))$ of cost smaller than or equal to ω^V, which contradicts the optimality of ω^U. If $|V \setminus U| = k > 1$ then one can apply the same argument k times. □

Proposition 2. *Let $U \subset V$, and let $\mathcal{C}^U$ be an optimal solution for $(U, D(U))$ of cost ω^U. If one can find a clustering $\mathcal{C}^V$ of V of cost ω^U, then that clustering is optimal for V.*

Proof. From the lemma, ω^U is a lower bound on the optimal solution value of DM for the set V. If ω^U is also the value of an upper bound (a feasible solution), then ω^U is optimal. □

Hypothesis 3. There exists a subset $U \subset V$ of size $n' \ll n$ such that Proposition 2 holds for U.

Our algorithm aims to provide an efficient way of finding such set U.

3. The general framework

We begin by selecting (using some criterion) an initial set U^0 containing κ points from V, where κ is a positive integer parameter, usually small. Let us set $i \leftarrow 0$. At iteration i, one solves problem $(U^i, D(U^i))$ to optimality. Let $\mathcal{C}^i, \omega^i$ denote the optimal solution and its value, respectively, for problem $(U^i, D(U^i))$. We then run a heuristic and try to build a feasible solution for problem $(V, D(V))$ of cost ω^i. Our heuristic proceeds as follows: The nodes in $V \setminus U^i$ are sorted using some ordering cheap to compute. Every node $v \in V \setminus U^i$ is then iteratively inserted into $\mathcal{C}^i$ on some cluster that does not see its diameter increased. If for some node v^* this is not possible, the heuristic stops. We set $U^{i+1} \leftarrow U^i \cup \{v^*\}$, $i \leftarrow i + 1$ and the algorithm is restarted. If, on the contrary, all nodes in $V \setminus U^i$ could be inserted into $\mathcal{C}^i$, then an optimal clustering of $(V, D(V))$ has been found. Algorithm 1 illustrates our method.

In this algorithm, procedure `OptimalClustering`$(U, D(U))$ solves, in our case via a branch-and-price algorithm, the clustering problem restricted to the nodes in U to optimality. It returns the optimal clustering $\mathcal{C}^U$ and its objective value ω^U. The function `HeuristicCompletion` $(\mathcal{C}^U, V \setminus U)$ completes the solution found by `OptimalClustering` $(U, D(U))$ and returns the resulting clustering $\mathcal{C}^V$, its value ω^V and a set $W = \{w\}$ containing a node that could not be inserted into $\mathcal{C}^U$ without augmenting its cost ($\emptyset$ if no such node exists).

The exactness and finiteness of our approach is based on the fact that, at every iteration, the heuristic either proves the optimality of the current subproblem (supported by Proposition 2), or the set V is augmented. In the worst case, U will grow up to become equal to V, in which

Input: Problem $P = (V, D(V))$
Output: Optimal clustering $\mathcal{C}^V = \{C_j : j = 1, \ldots, k\}$ of P
Build U by selecting κ points from V
$\omega^U, \omega^V \leftarrow \infty$
$\mathcal{C}^U, \mathcal{C}^V \leftarrow \emptyset$
$W \leftarrow \emptyset$
repeat
  $U \leftarrow U \cup W$
  $(\omega^U, \mathcal{C}^U) \leftarrow$ `OptimalClustering`$(U, D(U))$
  $(\omega^V, \mathcal{C}^V, W) \leftarrow$ `HeuristicCompletion`$(\mathcal{C}^U, V \setminus U)$
until $W = \emptyset$;
return $\mathcal{C}^V$

Algorithm 1: Iterative clustering

case procedure `OptimalClustering`$(V, D(V))$ is guaranteed to return an optimal solution for problem $(V, D(V))$. A simple worst-case time analysis of our method based upon this observation would suggest that it is indeed theoretically slower than solving the complete problem at once. However, in practice, our algorithm solves problems several orders of magnitude smaller than the original one.

4. Computational experiments

The proposed algorithm was implemented in C++ using the GNU g++ compiler v5.2. The linear programming solver used within our branch-and-price procedure was CPLEX 12.6. The algorithm was compiled and executed on a machine powered by an Intel Core i7-4710HQ CPU @ 2.50GHz$\times$8 with 16GB of RAM, running the Ubuntu 15.10 Operating System.

In our experimental analysis, we consider some classical problems from clustering, classification and regression tasks. In Table 1 we provide the detail of the datasets considered in our computational study. In this table, n, k and d stand for the number of points, the number of clusters and the dimension of the problem, respectively. Column labeled *Ref* contains a reference to the source of the problem.

Table 1. Problems details

Problem	n	k	d	Ref
Iris	150	3	4	[11]
Wine	178	3	13	[11]
Glass	214	7	9	[11]
Ionosphere	351	2	34	[11]
User knowledge	403	4	5	[10]
Breast cancer	569	2	30	[11]
Synthetic control	600	6	60	[1]
Vehicle	846	4	18	[12]
Yeast	1,484	10	8	[11]
Mfeat (morph)	2,000	10	6	[4]
Segmentation	2,000	7	19	[4]
Waveform (v2)	5,000	3	40	[11]

Table 2 compares our algorithm against Dao et al.'s CP algorithm [4], the RBBA of Brusco and Stahl [3] and the BB of Delattre and Hansen [6]. As one can see from this table, our algorithm takes two seconds or less to solve all the problems, including problem *Waveform (v2)* that passed from being solved in almost a minute, to only two seconds.

Table 2. Running times (in seconds) on datasets

Problem	Opt	RBBA	BB	CP	IC
Iris	2.58	1.4	1.8	< 0.1	< 0.1
Wine	458.13	2.0	2.3	< 0.1	< 0.1
Glass	4.97	8.1	42.0	0.2	0.2
Ionosphere	8.6		0.6	0.3	0.2
User knowledge	1.17		3.7	0.2	1.2
Breast cancer	2,377.96		1.8	0.5	0.2
Synthetic control	109.36			1.6	0.4
Vehicle	264.83			0.9	0.2
Yeast	0.67			5.2	1.7
Mfeat (morph)	1,594.96			8.59	0.6
Segmentation	436.4			5.7	0.6
Waveform (v2)	15.58			50.1	2.0

Acknowledgments

D. Aloise has been partially financed by CNPq-Brazil grant 308887/2014-0. C. Contardo has been partially financed by the FRQNT Grant no 181909, and NSERC Grant no 435824-2013.

References

[1] R.J. Alcock and Y. Manolopoulos. Time-series similarity queries employing a feature-based approach. In *7th Hellenic Conference on Informatics*, 1999.

[2] C.J. Alpert and A.B. Kahng. Splitting an ordering into a partition to minimize diameter. *Journal of Classification*, 14(1):51–74, 1997.

[3] M.J. Brusco and S. Stahl. *Branch-and-bound applications in combinatorial data analysis*. Springer Science & Business Media, 2006.

[4] T.-B.-H. Dao, K.-C. Duong, and C. Vrain. Constrained clustering by constraint programming. *Artificial Intelligence*, in press.

[5] M.R. Garey and D.S. Johnson. *Computers and intractability: a guide to NP-completeness*. WH Freeman New York, 1979.

[6] P. Hansen and M. Delattre. Complete-link cluster analysis by graph coloring. *Journal of the American Statistical Association*, 73(362):397–403, 1978.

[7] P. Hansen and B. Jaumard. Cluster analysis and mathematical programming. *Mathematical programming*, 79(1-3):191–215, 1997.

[8] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: A review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[9] S.C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[10] H. T. Kahraman, S. Sagiroglu, and I. Colak. Developing intuitive knowledge classifier and modeling of users' domain dependent data in web. *Knowledge Based Systems*, 37:283–295, 2013.

[11] M. Lichman. UCI machine learning repository, 2013.

[12] J. P. Siebert. Vehicle recognition using rule based methods. Research Memorandum TIRM-87-018, Turing Institute, 1987.

Proceedings of GOW'16, pp. 175 – 178.

Global Optimization of Continuous Minimax Problem

Dominique Monnet, Jordan Ninin and Benoit Clement

Lab-STICC, ENSTA-Bretagne, 2 rue Francois Verny, 29806 Brest, France
dominique.monnet@ensta-bretagne.org
jordan.ninin@ensta-bretagne.fr
benoit.clement@ensta-bretagne.fr

Abstract A Branch&Bound algorithm based on interval arithmetic is described solving non-convex minimax problem with continuous variable. Numerical results will follow on control synthesis with H_∞ robust constraint.

Keywords: Global optimization, Interval arithmetic, Contractor programming

1. Introduction

In this talk, we show that problem (1) can be solved in a guaranteed way using a global optimization based on interval arithmetic. Moreover, global optimization gives an enclosure of the minimum value of the objective function:

$$\begin{array}{ll} \min\limits_{x\in\mathbb{X}\subset\mathbb{R}^n} & \sup\limits_{y\in Y\subseteq R} f(x,y) \\ s.t. & c_i(x)\leq 0,\ \forall i\in\{1,\dots,m\}. \end{array} \tag{1}$$

We first introduce the interval arithmetic and the notion of inclusion function. After that we show how to tackle the maximization part of this minimax problem with interval arithmetic. Finally, we propose a Branch&Bound algorithm to solve the whole problem.

2. Interval Arithmetic

An Interval is a closed connected subset of $\mathbb{R}$ [2]. Intervals are denoted using boldface letters $\boldsymbol{x}$. A non-empty interval $\boldsymbol{x}$ can be represented by its endpoints:

$$\boldsymbol{x}=[\underline{x},\overline{x}]=\{x:\underline{x}\leq x\leq\overline{x}\}$$

with $\underline{x}\in\mathbb{R}\cup\{-\infty\}$, $\overline{x}\in\mathbb{R}\cup\{+\infty\}$ and $\underline{x}\leq\overline{x}$. The set of intervals will be denoted by $\mathbb{IR}$ and the set of n-dimensional interval vectors, also called boxes, will be denoted by $\mathbb{IR}^n$.
The main advantage of interval arithmetic is to be reliable, in the sense that it provides a guaranteed enclosure of the result. This inclusion property is the fundamental theorem of interval arithmetic. Let $g:\mathbb{R}^n\mapsto\mathbb{R}^m$ be a function. An inclusion function $\boldsymbol{g}:\mathbb{IR}^n\mapsto\mathbb{IR}^m$ of g is defined as follows,

$$\forall\boldsymbol{x}\in\mathbb{IR}^n, g(\boldsymbol{x})=\{g(x),x\in\boldsymbol{x}\}\subseteq\boldsymbol{g}(\boldsymbol{x}). \tag{2}$$

An inclusion function provides an upper and a lower bound of a function over an interval. Several techniques can be used to construct an inclusion function of every factorable function with $+,-,*,sin,exp,\ max,min,...$ [6].

3. Enclosure of the objective function

The objective function of problem (1) involves the computation of $f_{sup}(x) = \underset{y\in Y}{sup} f(x,y)$. To minimize this objective function using a Branch&Bound algorithm, we need to compute a lower and an upper bound of f_{sup} over a box $\boldsymbol{x}$. This bound must be as close as possible to the real range of f_{sup} to ensure the convergence of the Branch&Bound algorithm.

Using interval arithmetic, an inclusion function $\boldsymbol{f}$ of $f(x,y)$ can be constructed. Indeed, the operators $+$, $*$, abs, $\sqrt{}$, max are well-defined [3]. To use the interval arithmetic, we make an usual assumption and limits the study of $\underset{y}{sup} f(x,y)$ to a bounded set $\mathbb{Y} \subset \mathbb{R}$. Indeed, we have the following equation:

$$\forall \boldsymbol{x} \in \mathbb{IR}^n, f_{sup}(\boldsymbol{x}) = \left\{ \underset{y\in Y}{sup} f(x,y) \ : \ x \in \boldsymbol{x} \right\} \subseteq \boldsymbol{f}(x, \mathbb{Y}).$$

Unfortunately, the bounds obtained using the inclusion function directly over $\mathbb{Y}$ are not close enough to $f_{sup}(\boldsymbol{x})$. That is why we subdivide $\mathbb{Y}$ into several boxes $\boldsymbol{y}_i$ such as $\mathbb{Y} = \bigcup_i \boldsymbol{y}_i$. Thus, we have the following result:

$$\forall \boldsymbol{x} \in \mathbb{IR}^n, f_{sup}(\boldsymbol{x}) \subseteq \left[\max_i \underline{\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}_i)}, \max_i \overline{\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}_i)} \right].$$

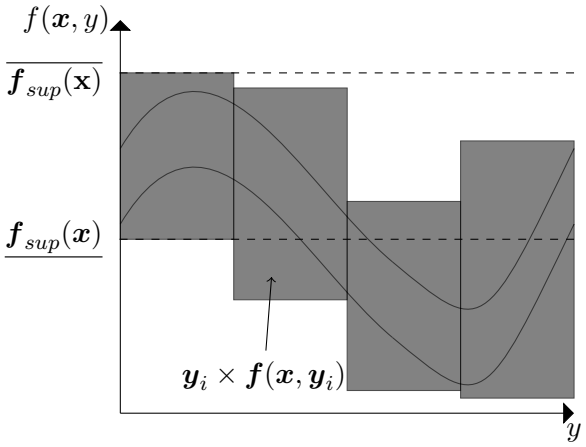


Figure 1. Computation of $\boldsymbol{f}_{sup}(\boldsymbol{x})$.

Figure 1 illustrates the previous equation. $\mathbb{Y}$ is discretized with non-overlapping intervals and $\boldsymbol{f}$ is evaluated for each of them. The highest lower bound and upper bound of the evaluations represented by dotted lines in Figure 1 give a guaranteed enclosure of $f_{sup}(\boldsymbol{x})$. Indeed, the function $\boldsymbol{f}_{sup}$, defined as follows, is an inclusion function of f_{sup}:

$$\forall \boldsymbol{x} \in \mathbb{IR}^n, \boldsymbol{f}_{sup}(\boldsymbol{x}) = \left[\max_i \underline{\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}_i)}, \max_i \overline{\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}_i)} \right].$$

To save computation time, some boxes $\boldsymbol{y}_i$ are eliminated if it is certified that it cannot contain the maximum, i.e. if $\max_j \underline{\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}_j)} \geq \overline{\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}_i)}$. Moreover, for a box $\boldsymbol{x}$, this computation will be performed inside a Branch&Bound algorithm (described in the next section), so the discretization of $\mathbb{Y}$ can be transmitted to all its sub-boxes. Indeed, for a box $\boldsymbol{x}$, if a box $\boldsymbol{y}_i$ is

eliminated, we do not need to compute $\boldsymbol{f}(\boldsymbol{x}_p, \boldsymbol{y}_i)$ for every sub-boxes $\boldsymbol{x}_p \subset \boldsymbol{x}$. This process accelerates the computation time of the Branch&Bound algorithm.

4. Global Optimization Algorithm

We propose to solve problem (1) with a Branch&Bound algorithm [3, 4]. We denote μ the global minimum of problem (1), $\mathbf{X} \subseteq \mathbb{R}^n$ the initial domain of the variable, and lb_μ and ub_μ a lower and an upper bounds of μ respectively.

The principle of a Branch&Bound algorithm is to split $\mathbf{X}$ into smaller subsets and eliminate them if they do not satisfy the constraint or if it is certified that they do not contain the global minimum. At the end of the algorithm, we obtained:

- x^* the best feasible solution found,
- $[lb_\mu, ub_\mu]$ a reliable enclosure of the minimum μ,
- A certificate of infeasibility if no solution has been found.

The complexity of our algorithm is exponential, but since the past fifteen years, the *global optimization* community has divided the required CPU time by a factor 10^9 for solving such problems. This gain is due (in roughly equal parts) to (i) hardware improvement, (ii) progress in the resolution of linear programs, and (iii) implementation efficiency of advanced mathematical techniques [1]. Most of these progress are included in our implementation.

Several ideas are used to improve the convergence:

- Based on the constraints $c_i(x) \leq 0$, advanced techniques of Constraint Satisfaction Problem are used to reduce a box $\boldsymbol{x}$ to the feasible domain [5]. These techniques contract the domain under study, to focus on the feasible domain.
- We test some feasible points x to improve the upper bound up_μ. Indeed, the value of each feasible point is an upper of the global minimum.
- Using the method of Section 3, lower bounds of the objective function can be obtained for each box $\boldsymbol{x}$. These bounds focus the algorithm on the part of the domain which has more chance to contain the global minimum.

Branch&Bound algorithm based on interval arithmetic.

```
While |ub_μ − lb_μ| ≥ ε and L ≠ ∅
    Extract a box x from L
    Contract x using [5].
    Bissect x into x_1 and x_2.
    for (i = 1, 2) {
        Compute f_sup(x_i).
        if (f_sup(x_i) ≤ ub_μ) {
        Add x_i in L.
        Choose x ∈ x_i that respects ∀i, c_i(x) ≤ 0.
        if (f_sup(x) < ub_μ)
            Update ub_μ and x* := x.
    }
    Update lb_μ := min_{x∈L} f_sup(x).
```

This algorithm works as follows. A list of boxes $\mathcal{L}$ is initialized with $\mathbf{X}$. At Line 2, the box $\boldsymbol{x}$ with the lowest lower bound $\underline{\boldsymbol{f}_{sup}(\boldsymbol{x})}$ is chosen from $\mathcal{L}$. At Line 3, $\boldsymbol{x}$ is contracted using [5]. If the result is not empty, $\boldsymbol{x}$ is bisected in two non-overlapping boxes $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. For $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, $\boldsymbol{f}_{sup}(\boldsymbol{x}_i)$ is computed at Line 6 using Section 3. If $\underline{\boldsymbol{f}_{sup}(\boldsymbol{x})} > ub_\mu$, it is proved that the global minimum cannot be in $\boldsymbol{x}$, else the box is added to $\mathcal{L}$. At Line 9, if possible, a point x that stabilizes $P(s)$ is chosen in $\boldsymbol{x}$. If $\overline{\boldsymbol{f}_{sup}(x)}$ is lower than the current value ub_μ, ub_μ is updated. When this algorithm stops, $\mathcal{L}$ contains a set of boxes that can contain the global minimizer; $[lb_\mu, ub_\mu]$ provides a guaranteed enclosure of the global minimum μ; x^* contains the best known solution. If $(lb_\mu > 1)$ or if $(\mathcal{L} = \emptyset) \wedge (x^* = \emptyset)$, it is certified that problem (1) has no solution.

References

[1] R. E. Bixby. A brief history of linear and mixed-integer programming computation. *Documenta Mathematica*, Extra Volume: Optimization Stories:107–121, 2012.

[2] Luc Jaulin, Michel Kieffer, Olivier Didrit, and ÃL'ric Walter. *Applied Interval Analysis*. Springer London, 2001.

[3] R. B. Kearfott. An interval Branch&Bound algorithm for bound constrained optimization problems. *Journal of Global Optimization*, 2(3):259–280, 1992.

[4] J. Ninin. *Global Optimization based on Interval Analysis: Affine Relaxation and Limited Memory*. Theses, Institut National Polytechnique de Toulouse - INPT, December 2010.

[5] J. Ninin, F. Messine, and P. Hansen. A reliable affine relaxation method for global optimization. *4OR: A Quarterly Journal of Operations Research*, 13(3):247–277, September 2015.

[6] B. Tóth. *Interval Method for Competitive Location Problems*. phdthesis, University of Almeria, Spain, 2007.

Proceedings of GOW'16, pp. 179 – 181.

Core-Based Upper Bounds for the Maximum Clique Problem

Chitra Balasubramaniam[1], Balabhaskar Balasundaram[2] and Sergiy Butenko[1]

[1]*Texas A&M University, Department of Industrial and Systems Engineering, College Station, TX 77843-3131,* {bcvaidyanath,butenko}@tamu.edu

[2]*School of Industrial Engineering and Management, Oklahoma State University, Stillwater, OK 74078,* baski@okstate.edu

Abstract In this paper, we develop strong and easily computable upper bounds for the maximum clique problem that take advantage of continuous relaxations of integer programming formulations for the minimum k-core problem and its diameter-restricted version. We show that the proposed upper bounds are better than those obtained from the standard relaxations of the clique polytope. We also report the results of computational experiments with several classes of graph instances and provide a comparison with some existing bounds from the literature.

Keywords: Clique, k-Core, Clique relaxations

1. Introduction

A clique is a subset of pairwise adjacent vertices, and the maximum clique problem aims to find a clique of maximum cardinality, also called the clique number. This problem has diverse applications in various fields [5], that include social, biological and communication networks [3, 10, 21], and hence, is one of the most popular and well studied problems in combinatorial optimization and theoretical computer science. However, the problem is NP-hard [12] and is hard to approximate [17]. Motivated by these factors, many exact branch and bound algorithms [22, 23, 25, 29, 32], enumerative algorithms [7, 30], and a variety of heuristics [1, 4, 8, 13, 14, 16] have been developed.

Pertinently, many convex relaxations [24] that determine a good upper bound on the clique number have also been proposed. Perhaps the strongest results in this direction are related to the notion of Lovász theta [20], denoted by $\theta(G)$, which satisfies the inequalities $\omega(G) \leq \theta(\bar{G}) \leq \chi(G)$, known as the sandwich theorem [18]. Here, $\bar{G}$ is the complement graph of G, $\omega(G)$ is the clique number, and $\chi(G)$ is the chromatic number, which is the minimum number of colors required in a proper coloring of G. Both $\omega(G)$ and $\chi(G)$ are hard to compute, whereas $\theta(\bar{G})$ can be computed in polynomial time. In fact, there is no provably better than $\theta(\bar{G})$ upper bound on $\omega(G)$ unless P=NP [9]. The Lovász theta and its stronger variants have been used to obtain tight upper bounds on the clique number in several works [6, 15, 19, 27, 28]. However, computing such bounds involves lifting to higher dimensional spaces, limiting their applicability to only smaller graphs in practice.

In this paper, we take an alternate approach in which we seek a *minimum* cardinality subset of vertices satisfying one or more elementary graph-theoretic properties that must hold for any clique of a fixed size, say $k+1$, also referred to as *elementary clique-defining properties* [26]. If this minimization problem is infeasible for G, or the minimum is greater than $k+1$, we can deduce that $\omega(G) \leq k$. Even if we obtain a valid lower bound on the minimum that is greater than $k+1$, we can still conclude that $\omega(G) \leq k$. Hence, our objective in this work is to develop nontrivial, easily computable lower bounds for the minimization problem in order to get a tight upper bound on $\omega(G)$. More specifically, linear programming (LP) based bounds for two such minimization problems based on minimum degree and pairwise distance

requirements are investigated and compared to the standard LP relaxations for the maximum clique problem.

Given a positive integer k, a subset S of vertices is a k-core if $\delta(G[S]) \geq k$, and the degeneracy of G is the largest k for which G has a nonempty k-core. Obviously, any clique of size $k+1$ is a k-core. Hence, given a lower bound k for $\omega(G)$, the k-core obtained by recursively removing all vertices of degree less than k from the graph ("peeling") can be used for scale-reduction in very large and sparse graphs for employing exact algorithms to solve the maximum clique problem [1, 11, 31]. However, the peeling procedure yields the maximum-size k-core in the graph that may be much larger than the clique number. In this work, we use *minimum* k-cores to get an upper bound on the clique number of a graph. Consider the following problem:

Definition 1 (Minimum k-core)**.** *Given a graph $G = (V, E)$, and a positive integer k such that G has degeneracy at least k, find a smallest non-empty k-core in G. The size of a minimum k-core is denoted by $m_k(G)$.*

It is easy to see that if G contains a clique of size $k+1$, then $m_k(G) = k+1$, and for a given k, if there is no k-core in G or $m_k(G) > k+1$, then $\omega(G) \leq k$. Although this property helps us obtain a valid upper bound for the maximum clique problem, using exact algorithms for finding the smallest such k satisfying this property is not a viable approach from a computational standpoint for the following reason. Unlike the maximum k-core problem, which is polynomially solvable, the minimum k-core problem does not admit a polynomial-time constant factor approximation algorithm for $k \geq 3$, unless P = NP [2]. Hence, as a starting point, we aim to show that LP relaxations of the minimum k-core problem, and strengthened variations of the same can be used to easily compute good upper bounds for the maximum clique problem.

References

[1] J. Abello, P.M. Pardalos, and M.G.C. Resende. On maximum clique problems in very large graphs. In J. Abello and J. Vitter, editors, *External Memory Algorithms and Visualization*, pages 119–130. American Mathematical Society, Boston, 1999.

[2] O. Amini, D. Peleg, S. Pérennes, I. Sau, and S. Saurabh. On the approximability of some degree-constrained subgraph problems. *Discrete Applied Mathematics*, 160:1661–1679, 2012.

[3] B. Balasundaram, S. Butenko, and S. Trukhanov. Novel approaches for analyzing biological networks. *Journal of Combinatorial Optimization*, 10:23–39, 2005.

[4] R. Battiti and M. Protasi. Reactive local search for the maximum clique problem. *Algorithmica*, 29:610–637, 2001.

[5] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo. The maximum clique problem. In D.-Z. Du and P. M. Pardalos, editors, *Handbook of Combinatorial Optimization*, pages 1–74, Dordrecht, The Netherlands, 1999. Kluwer Academic Publishers.

[6] I. M. Bomze, F. Frommlet, and M. Locatelli. Copositivity cuts for improving SDP bounds on the clique number. *Mathematical Programming*, 124:13–32, 2010.

[7] C. Bron and J. Kerbosch. Algorithm 457: Finding all cliques of an undirected graph. *Commun. ACM*, 16(9):575–577, September 1973.

[8] S. Busygin, S. Butenko, and P. M. Pardalos. A heuristic for the maximum independent set problem based on optimization of a quadratic over a sphere. *Journal of Combinatorial Optimization*, 6(3):287–297, 2002.

[9] S. Busygin and D. V. Pasechnik. On np-hardness of the clique partitionâĂŤindependence number gap recognition and related problems. *Discrete Mathematics*, 306(4):460 – 463, 2006.

[10] Y. P. Chen, A. L. Liestman, and J. Liu. Clustering algorithms for ad hoc wireless networks. In Y. Pan and Y. Xiao, editors, *Ad Hoc and Sensor Networks, Wireless Networks and Mobile Computing*, pages 145–164, New York, 2005. Nova Science Publishers.

[11] D. Eppstein and D. Strash. Listing all maximal cliques in large sparse real-world graphs. In P. M. Pardalos and S. Rebennack, editors, *Experimental Algorithms*, volume 6630 of *Lecture Notes in Computer Science*, pages 364–375. Springer, Heidelberg, 2011.

[12] 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.

[13] M. Gendreau, P. Soriano, and L. Salvail. Solving the maximum clique problem using a tabu search approach. *Annals of Operations Research*, 41:385–403, 1993.

[14] L. E. Gibbons, D. W. Hearn, and P. M. Pardalos. A continuous based heuristic for the maximum clique problem. In D. S. Johnson and M. A. Trick, editors, *Cliques, Coloring, and Satisfiability: Second* DIMACS *Implementation Challenge*, pages 103–124, Providence, RI, 1996. American Mathematical Society.

[15] G. Gruber and F. Rendl. Computational experience with stable set relaxations. *SIAM Journal on Optimization*, 13(4):1014–1028, 2003.

[16] P. Hansen, N. Mladenović, and D. Uroševic. Variable neighborhood search for the maximum clique. *Discrete Applied Mathematics*, 145:117–125, 2004.

[17] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182:105–142, 1999.

[18] D. E. Knuth. The sandwich theorem. *Electronic Journal of Combinatorics*, 1:A1, 1994.

[19] M. Locatelli. Improving upper bounds for the clique number by non-valid inequalities. *Mathematical Programming*, 150:511–525, 2015.

[20] L. Lovász. On the Shannon capacity of a graph. *IEEE Trans. Inform. Theory*, 25:1–7, 1979.

[21] R. D. Luce and A. D. Perry. A method of matrix analysis of group structure. *Psychometrika*, 14:95–116, 1949.

[22] P. R. J. Östergård. A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics*, 120:197–207, 2002.

[23] P. M. Pardalos and G. P. Rodgers. A branch and bound algorithm for the maximum clique problem. *Computers and Operations Research*, 19:363–375, 1992.

[24] P. M. Pardalos and J. Xue. The maximum clique problem. *Journal of Global Optimization*, 4:301–328, 1994.

[25] B. Pattabiraman, M. Patwary, A. Gebremedhin, W. Liao, and A. Choudhary. Fast algorithms for the maximum clique problem on massive sparse graphs. In *Algorithms and Models for the Web Graph*, volume 8305 of *Lecture Notes in Computer Science*, pages 156–169. Springer, Heidelberg, 2013.

[26] J. Pattillo, N. Youssef, and S. Butenko. On clique relaxation models in network analysis. *European Journal of Operational Research*, 226:9–18, 2013.

[27] J. Peña, J. Vera, and L. F. Zuluaga. Computing the stability number of a graph via linear and semidefinite programming. *SIAM Journal on Optimization*, 18:87–105, 2007.

[28] A. Schrijver. A comparison of the Delsarte and Lovász bounds. *IEEE Trans. Inform. Theory*, 25:425–429, 1979.

[29] E. Tomita and T. Seki. An efficient branch-and-bound algorithm for finding a maximum clique. In *Proceedings of the 4th International Conference on Discrete Mathematics and Theoretical Computer Science*, DMTCS'03, pages 278–289, 2003.

[30] E. Tomita, A. Tanaka, and H. Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theor. Comput. Sci.*, 363(1):28–42, October 2006.

[31] A. Verma, A. Buchanan, and S. Butenko. Solving the maximum clique and vertex coloring problems on very large sparse networks. *INFORMS Journal on Computing*, 27(1):164–177, 2015.

[32] D. R. Wood. An algorithm for finding a maximum clique in a graph. *Operations Research Letters*, 21(5):211–217, 1997.

SESSION: COMPLEX SOLUTION PATTERNS FOR STANDARD QUADRATIC OPTIMIZATION

Proceedings of GOW'16, pp. 185 – 188.

Finding and Analyzing Hard Instances of the Standard Quadratic Optimization Problem*

Immanuel Bomze[1], Werner Schachinger[1] and Reinhard Ullrich[1]

[1]*Dept. of Statistics and OR, University of Vienna* immanuel.bomze@univie.ac.at

Abstract In a Standard Quadratic Optimization Problem (StQP), a possibly indefinite quadratic form (the simplest nonlinear function) is extremized over the standard simplex, the simplest polytope. Despite this simplicity, the nonconvex instances of this problem class allow for remarkably rich patterns of coexisting local solutions, which are closely related to practical difficulties in solving StQPs globally. In this study, we improve on existing lower bounds for the number of strict local solutions by a new technique to construct instances with a rich solution structure. To this end, we profit from the close connections of StQP with Evolutionary Game Theory and other, seemingly unrelated areas of Mathematical Modeling.

Keywords: Quadratic optimization, Evolutionary stability

1. Introduction

We consider the *Standard Quadratic Optimization Problem (StQP)* given by

$$\max\left\{\mathsf{x}^\top \mathsf{A}\mathsf{x};\, \mathsf{x}\in\Delta^n\right\} \quad \text{with} \quad \Delta^n=\left\{\mathsf{x}\in\mathbb{R}^n_+ : \sum_{i\in N} x_i = 1\right\}, \tag{1}$$

where A is a symmetric $n\times n$-matrix and $N=\{1,\ldots,n\}$. Despite of its simplicity, this model is quite versatile. Applications are numerous, ranging from the famous Markowitz portfolio problem in Finance, Economics (evolutionary game theory) through Machine Learning (background-foreground clustering in image analysis) to life science applications, e.g. in Population Genetics (selection models) and Ecology (replicator dynamics). So it is not too surprising that the following questions are closely related (for detailed explanations and background see e.g.[1] and references therein):

How many strict local solutions are there at most in a given StQP ?
How many evolutionarily stable states (ESSs) can coexist in a given (partnership) game ?
How many asymptotically stable fixed points can coexist for the replicator dynamics ?
How many stable equilibria can coexist in a one-locus, multi-allelic system ?
How many maximal cliques can coexist in an undirected graph ?

The last question sheds light on an important aspect of StQPs, namely the discrete combinatorial structure of this problem class of continuous optimization models. Indeed, for the subclass of StQPs based upon an adjacency matrix A of an undirected graph, the answer to the last question is well known by the famous Moon/Moser bound [10]: asymptotically $\sqrt[3]{3}^n \approx (1.4422)^n$, a number exponential in the order n of the graph, and this bound is sharp, attained at the complete multipartite Turán graph $T(n,\lceil\frac{n}{3}\rceil)$.

However, the Moon/Moser bound is not valid for general symmetric matrices A (with also non-binary entries). In this paper, we will push this exponential bound up, improving, and building, upon earlier investigations [5] where the basis ≈ 1.4592 was established, to a basis

*See also "Constructing StQPs with many local solutions" and "Complexity of an StQP: a theory-guided experimental study"

of ≈ 1.48106, establishing, to the best of our knowledge, a new world record (as of March 2016). Note that a worst-case *upper* bound on the maximal number of strict local solutions asymptotically equals $\approx \frac{2^n}{1.25\sqrt{n}}$ (see (6) below), so that the basis necessarily is smaller than two.

2. Looking over the fence; local solution structure

The above mentioned close relations between the different fields Optimization, game theory and population/selection dynamics; as explained in [1], the optimization problem (1) is closely related to an *evolutionary game* with strategy set N in pairwise contests, with payoff matrix A. If $\mathsf{A} = \mathsf{A}^\top$, this means that the row and the column player share the payoff equally, a partnership game. Likewise, the symmetric matrix $\mathsf{A} = [a_{ij}]$ could also collect the (incremental) fitness values for the allelic combination $\{i, j\} \in N \times N$ where N is the allele set for a single autosomal locus. Finally we may look a certain population dynamics called *replicator dynamics* defined as a system of coupled autonomous differential equations, a continuous-time dynamical system (a dot $\dot{y}$ signifies derivative w.r.t. time t) and perform the usual qualitative equilibrium analysis:

$$\dot{x}_i(t) = x_i(t)\left([\mathsf{A}\mathsf{x}(t)]_i - \mathsf{x}(t)^\top \mathsf{A}\mathsf{x}(t)\right), \quad i \in N, \quad \mathsf{x}(0) = \mathsf{x}_0 \in \Delta^n; \quad t \geq 0. \tag{2}$$

We have the following equivalences for a point $\mathsf{x} \in \Delta^n$:

$$\left.\begin{array}{rll} & \mathsf{x} & \text{is a strict local maximizer of (1), i.e. strictly maximizes population overall fitness} \\ \Leftrightarrow & \mathsf{x} & \text{is an evolutionary stable strategy (ESS) for payoff matrix } \mathsf{A} \\ \Leftrightarrow & \mathsf{x} & \text{is an asymptotically stable fixed point for the dynamics (2).} \end{array}\right\} \tag{3}$$

For succinct proofs, we refer to [1, Theorem 10] where also equivalences for the weaker versions of solutions are stated and proved:

$$\left.\begin{array}{rll} & \mathsf{x} & \text{is a local maximizer of (1), i.e. maximizes population overall fitness} \\ \Leftrightarrow & \mathsf{x} & \text{is a neutrally stable strategy (NSS) for payoff matrix } \mathsf{A} \\ \Leftrightarrow & \mathsf{x} & \text{is a Lyapunov stable fixed point for the dynamics (2);} \end{array}\right\} \tag{4}$$

and

$$\left.\begin{array}{rll} & \mathsf{x} & \text{is a KKT point for (1), i.e. satisfies first-order conditions for local maximality} \\ \Leftrightarrow & \mathsf{x} & \text{is a Nash equilibrium strategy (NES) for payoff matrix } \mathsf{A} \\ \Leftrightarrow & \mathsf{x} & \text{is a saturated fixed point for the dynamics (2).} \end{array}\right\} \tag{5}$$

For us relevant are the first equivalences in (3), (4) and (5), and in the following will use both terms (strict local maximizer/ESS; local maximizer/NSS; KKT point/NES) interchangeably. For the readers' convenience, we will repeat the definitions of ESS, NSS, NES below; note that these also apply to non-symmetric square matrices $\mathsf{A} \neq \mathsf{A}^\top$, used for modeling non-partnership evolutionary games; to this end, we will introduce a bit of notation first: given any point $\mathsf{x} \in \Delta^n$, let $I(\mathsf{x}) = \{i \in N : x_i > 0\}$ be the *support* of x.

A vector $\mathsf{x} \in \Delta^n$ is called a (symmetric) *Nash Equilibrium State/Strategy* (NES), in symbols $\mathsf{x} \in NES(\mathsf{A})$, if $\mathsf{x}^\top \mathsf{A}\mathsf{x} \geq \mathsf{y}^\top \mathsf{A}\mathsf{x}$ for all $\mathsf{y} \in \Delta^n$, i.e., if x is a best reply to itself. The set of all best replies y to x is denoted by $BR_{\mathsf{A}}(\mathsf{x}) = \left\{\mathsf{y} \in \Delta^n : \mathsf{y}^\top \mathsf{A}\mathsf{x} = \mathsf{x}^\top \mathsf{A}\mathsf{x}\right\}$.

A *Neutrally Stable State/Strategy* (NSS) is a NES which in addition satisfies $\mathsf{x}^\top \mathsf{A}\mathsf{y} \geq \mathsf{y}^\top \mathsf{A}\mathsf{y}$ for all $\mathsf{y} \in BR_{\mathsf{A}}(\mathsf{x})$; this is denoted by $\mathsf{x} \in NSS(\mathsf{A})$. In words, a state/strategy x is *neutrally stable* if it is at least as good a reply to any y than that y to itself, for any alternative best reply y to x. Finally, an NSS is called *Evolutionarily Stable State/Strategy* (ESS), in symbols $\mathsf{x} \in ESS(\mathsf{A})$, if

the last inequality is strict: $\mathsf{x}^\top \mathsf{A}\mathsf{y} > \mathsf{y}^\top \mathsf{A}\mathsf{y}$ for all $\mathsf{y} \in BR_{\mathsf{A}}(\mathsf{x}) \setminus \{\mathsf{x}\}$: any alternative best reply y to x fares strictly worse against itself than the incumbent x performs against y.

Obviously, $ESS(\mathsf{A}) \subseteq NSS(\mathsf{A}) \subseteq NES(\mathsf{A})$. The last two sets are never empty, but they can be infinite, while the first one has to be always finite (but may be empty, e.g. for $\mathsf{A} = \mathsf{O}$). However, generically, the first two sets coincide [1, Corollary 14], in which case they are both nonempty and finite. More precisely, since the *(ESS) pattern* of A,

$$\text{pattern}(\mathsf{A}) := \{I(\mathsf{x}) : \mathsf{x} \in ESS(\mathsf{A})\}$$

forms an antichain in the system of all subsets of N, which has n elements, by Sperner's theorem about maximal antichains and Stirling's approximation we get

$$|ESS(\mathsf{A})| \leq \binom{n}{\lfloor \frac{n}{2} \rfloor} \sim 2^n/\sqrt{\pi n}\,. \tag{6}$$

Instances with a rich such pattern (and thus many coexisting strict local solutions) will be the main focus of our investigations, extending and improving upon previous studies [5]. Despite its discrete combinatorial nature, the pattern comprises all essential information on the solution set of the instance, at least generically; note that generically we have $ESS(\mathsf{A}) = NSS(\mathsf{A})$. Indeed, given $I \in \text{pattern}(\mathsf{A})$, it is a mere matter of solving a linear equation system in $|I|$ variables to obtain the unique ESS x such that $I(\mathsf{x}) = I$.

3. Global solutions; worst- and average case complexity

Our present study draws upon our earlier work [3, 4] focused on the coexistence of *global* solutions in StQPs. Generically, there is only one global solution, but there are instances with exponentially many (strict) global solutions. We know that StQPs are NP-hard (e.g., by reduction to the maximum clique problem), and that they form a PTAS [2]. For general QPs, determining one local solution is already NP-hard [11]; note that for the StQP, a very efficient local maximization algorithm for the general case was proposed in [12]. Obviously, determining *all* local solutions cannot be easier than solving the StQP globally. For a thorough discussion of these and related issues, we refer to [7].

So what is a typical hard instance ? In a natural model for random $n \times n$ instances $\tilde{\mathsf{A}}$, Kontogiannis and Spirakis [8, 9] showed that $\mathbb{E}\left[|\text{pattern}(\tilde{\mathsf{A}})|\right] = \mathbb{E}\left[|ESS(\tilde{\mathsf{A}})|\right] = \gamma_n \mathbb{E}\left[|NES(\tilde{\mathsf{A}})|\right]$ with $\gamma_n \to 0$ as $n \to \infty$, and that

$$\mathbb{P}\left[\max\left\{|I|\colon I \in \text{pattern}(\tilde{\mathsf{A}})\right\} \geq n^{2/3}\right] \to 0 \quad \text{as } n \to \infty\,.$$

So relatively large support sizes exists in some patterns, but instances with those patterns become increasingly rare as n gets large. Interesting complex (and therefore large) patterns would likely be missed or occur too rarely in a naive random sampling approach. On a related question, Chen and Peng obtained very recently in [6] an even stronger probability bound, under a similar model for random instances, denoting by $\mathsf{x}^*(\mathsf{A})$ a (generically unique) global solution of (1):

$$\mathbb{P}\left[|I(\mathsf{x}^*(\tilde{\mathsf{A}}))| \geq 3\right] \to 0 \quad \text{as } n \to \infty\,.$$

Again, this shows that naive random sampling is not very promising; if we know that the support of the global solution is at most a doubleton (and this happens with a high probability), we simply have to search among all such $\mathcal{O}(n^2)$ doubletons and singletons; such instances occur with a high probability, and are no challenge algorithmically. Note that also the possible pattern structures are constrained as they cannot contain any supports containing a doubleton already detected as a member of the pattern. Therefore, the worst cases are hidden in regions of small probability, and we will explore that region more systematically than just random searching, developing structures which guarantee highly complex patterns.

4. Summary

This is part one of a series of three presentations, given by I. Bomze, W. Schachinger and R. Ullrich. The present part serves as an introduction and to give some background on the used framework and methods to empirically explore the complexity of the simplest of the hard optimization models – the Standard Quadratic Optimization Problem. We address the question of coexistence of many local solutions and the system of their support, the so-called patterns of an instance.

References

[1] Immanuel M. Bomze. Regularity versus degeneracy in dynamics, games, and optimization: a unified approach to different aspects. *SIAM Review*, 44(3):394–414, 2002.

[2] Immanuel M. Bomze and Etienne de Klerk. Solving standard quadratic optimization problems via linear, semidefinite and copositive programming. *Journal of Global Optimization*, 24(2):163–185, 2002.

[3] Immanuel M. Bomze, Werner Schachinger, and Reinhard Ullrich. From seven to eleven: completely positive matrices with high cp-rank. *Linear Algebra and its Applications*, 459:208–221, 2014.

[4] Immanuel M. Bomze, Werner Schachinger, and Reinhard Ullrich. New lower bounds and asymptotics for the cp-rank. *SIAM Journal on Matrix Analysis and Applications*, 36(1):20–37, 2015.

[5] Mark Broom, Chris Cannings, and Glenn T. Vickers. On the number of local maxima of a constrained quadratic form. *Proc. R. Soc. Lond. A*, 443:573–584, 1993.

[6] Xin Chen and Jiming Peng. New analysis on sparse solutions to random Standard Quadratic Optimization Problems and extensions. *Mathematics of Operations Research*, 40(3):725–738, 2015.

[7] Etienne de Klerk. The complexity of optimizing over a simplex, hypercube or sphere: a short survey. *Central European Journal of Operations Research*, 16(2):111–125, 2008.

[8] Spyros Kontogiannis and Paul Spirakis. Counting stable strategies in random evolutionary games. In X. Deng and D. Du, editors, *Proceedings of the 16th Annual International Symposium on Algorithms and Computation (ISAAC 2005)*, volume 3827 of *LNCS*, pages 839–848. Springer, Heidelberg, 2006.

[9] Spyros C. Kontogiannis and Paul G. Spirakis. On the support size of stable strategies in random games. *Theoretical Computer Science*, 410(8–10):933–942, 2009.

[10] John W. Moon and Leo Moser. On cliques in graphs. *Israel Journal of Mathematics*, 3(2):23–28, 1965.

[11] Katta G. Murty and Santosh N. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39(2):117–129, 1987.

[12] Samuel Rota-Bulò and Immanuel M. Bomze. Infection and immunization: a new class of evolutionary game dynamics. *Games Econ. Behav.*, 71(special issue in honor of J.F.Nash, jr.):193–211, 2011.

Proceedings of GOW'16, pp. 189 – 192.

Constructing Standard Quadratic Optimization Problems with Many Local Solutions *

Werner Schachinger,[1] Immanuel Bomze[1] and Reinhard Ullrich[1]

[1] *Dept. of Statistics and OR, University of Vienna* werner.schachinger@univie.ac.at

Abstract In this study, we improve on existing lower bounds for the number of strict local solutions of Standard Quadratic Optimization Problems by employing a new technique to construct instances with a rich solution structure. To prove some vital properties of that construction it is necessary to study the influence of perturbations of a matrix on its pattern of coexisting local solutions. The results obtained along this line may be of independent interest.

Keywords: Quadratic optimization, Evolutionary stability

1. Introduction

While it is easy to find an instance of the *Standard Quadratic Optimization Problem (StQP)*

$$\max\left\{x^\top Ax : x \in \Delta^n\right\}$$

with no strict local solution at all ($A = O$, the zero matrix, will do) or just one such (take $A = E_n - I_n$, with E_n the all-ones matrix and I_n the identity matrix of order n), the other end of the spectrum of difficulty is less well known. One of our approaches to get closer to that far end is based upon a collection of given symmetric matrices $A, B_1, \ldots, B_n$ of orders $n, k_1, \ldots, k_n$. These will be used to construct a matrix of order $N := \sum_{i=1}^n k_i$. Define

$$Q = \begin{pmatrix} 1_{k_1} & o & \ldots & o \\ o & 1_{k_2} & \ldots & o \\ \vdots & \vdots & \ddots & \vdots \\ o & o & \ldots & 1_{k_n} \end{pmatrix}^\top$$

where 1_k is the all ones vector in $\mathbb{R}^k$, and o stands for the zero vector of appropriate dimension. Then both $A_Q := Q^\top AQ$ and $B := \mathrm{Diag}(B_1, \ldots, B_n)$ are symmetric matrices of order N, and, thinking of t as a large real parameter, we let

$$G_t = G_t(A, B_1, \ldots, B_n) = tA_Q + B\,.$$

Special cases of this construction have been considered before: The case $A = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, B_1, B_2 arbitrary has been used in the proof of [6, Theorem 1], and the case A arbitrary of order n, $B_1 = I_2$, $B_i = I_1$ for $2 \le i \le n$, has been used in the proof of [7, Theorem 1].

2. Some prerequisites

During the investigation of pattern(G_t) we come across matrices which are perturbations of A, those perturbations becoming slighter as t increases. For our first aim - to understand how the pattern changes as the matrix is perturbed - we need some definitions.

*See also "Finding and analyzing hard instances of the StQP" and "Complexity of an StQP: a theory-guided experimental study"

Denote by $\mathcal{F}_n$ those symmetric $n\times n$-matrices A such that (1) has only strict local maximizers (and therefore finitely many):

$$\mathcal{F}_n := \{\mathsf{A} \in \mathcal{S}^n : NSS(\mathsf{A}) = ESS(\mathsf{A})\} \ ;$$

and by $\mathcal{F}_n^r$ those symmetric $n \times n$-matrices A such that all NESs are quasistrict in the sense that e_i is an alternative best answer to x if and only if strategy i is used in x with positive frequency/probability. With $J_{\mathsf{A}}(\mathsf{x}) = \left\{i \in \{1,\ldots,n\} : [\mathsf{Ax}]_i = \mathsf{x}^\top\mathsf{Ax}\right\}$, that is

$$\mathcal{F}_n^r := \{\mathsf{A} \in \mathcal{S}^n : I(\mathsf{x}) = J_{\mathsf{A}}(\mathsf{x}) \text{ for all } \mathsf{x} \in NES(\mathsf{A})\} \ .$$

We have $\mathcal{F}_n^r \subseteq \mathcal{F}_n$, and that $\mathcal{F}_n^r$ is open and dense in $\mathcal{S}^n$ (see [2, Theorem 13, Corollary 14]). However, the set $\mathcal{F}_n$ is not open for $n \geq 3$, as can be seen from an example in [1].

Let $\mathcal{F} := \bigcup_{n\geq 1} \mathcal{F}_n$ and $\mathcal{F}^r := \bigcup_{n\geq 1} \mathcal{F}_n^r$. We can show that

a) if $\mathsf{A} \in \mathcal{F}_n$, then for all $\mathsf{B} \in \mathcal{F}_n$ sufficiently close to A we have

$$|\{K' \in \text{pattern}(\mathsf{B}) : \ K \subseteq K'\}| \geq 1, \quad \text{for all } K \in \text{pattern}(\mathsf{A}), \tag{1}$$
$$|\{K \in \text{pattern}(\mathsf{A}) : \ K \subseteq K'\}| \leq 1, \quad \text{for all } K' \in \text{pattern}(\mathsf{B}), \tag{2}$$

which implies

$$|\text{pattern}(\mathsf{B})| \geq |\text{pattern}(\mathsf{A})|.$$

b) if $\mathsf{A} \in \mathcal{F}_n^r$, then for all $\mathsf{B} \in \mathcal{F}_n$ sufficiently close to A we have

$$\text{pattern}(\mathsf{B}) = \text{pattern}(\mathsf{A}) \, .$$

We can have strict inequalities in any of (1) and (2), resulting in $|\text{pattern}(\mathsf{B})| > |\text{pattern}(\mathsf{A})|$, as well as strict inclusion $K \subset K'$ in (1) and (2).

Another ingredient to our analysis of the set $ESS(\mathsf{G}_t)$ is the following representation in terms of ESSs of the constituent matrices $\mathsf{B}_1, \ldots, \mathsf{B}_n$ and (a perturbation of) A.

Lemma 1. *Let* $\bar{b}_i := \max\limits_{\mathsf{x}\in\Delta^{k_i}} \mathsf{x}^\top\mathsf{B}_i\mathsf{x}$ *for* $1 \leq i \leq n$, *so that* $\bar{b}_i = \bar{\mathsf{p}}_i^\top\mathsf{B}_i\bar{\mathsf{p}}_i$ *for some* $\bar{\mathsf{p}}_i \in \Delta^{k_i}$. *Then, with*

$$\mathcal{E}_K := \left\{ \begin{pmatrix} \alpha_1\mathsf{p}_1 \\ \vdots \\ \alpha_n\mathsf{p}_n \end{pmatrix} : I(\boldsymbol{\alpha}) = K, \ \mathsf{p}_i \in ESS(\mathsf{B}_i) \text{ for } i \in K, \ \mathsf{p}_i = \bar{\mathsf{p}}_i \text{ for } i \notin K, \right. \\ \left. \boldsymbol{\alpha} \in ESS(t\mathsf{A} + \text{Diag}(\mathsf{b})), \text{ where } b_i := \begin{cases} \mathsf{p}_i^\top\mathsf{B}_i\mathsf{p}_i, & i \in K, \\ \bar{b}_i, & \text{else.} \end{cases} \right\} \tag{3}$$

we have $ESS(\mathsf{G}_t) = \bigcup\limits_{K\subseteq\{1,\ldots,n\}} \mathcal{E}_K$.

With the same assumptions as in the lemma, there are analogous representations

$$NSS(\mathsf{G}_t) = \bigcup_{K\subseteq\{1,\ldots,n\}} \mathcal{N}_K \qquad \text{and} \qquad NES(\mathsf{G}_t) = \bigcup_{K\subseteq\{1,\ldots,n\}} \mathcal{Q}_K,$$

where for the definition of the sets $\mathcal{N}_K$ and $\mathcal{Q}_K$ we just have to replace the two occurrences of ESS in the right hand side of (3) by twice NSS respectively twice NES.

We further introduce the following useful tools. For $\mathsf{A} \in \mathcal{F}_n$ define the *pattern generating* polynomial p_{A} and the *support size generating* polynomial q_{A} by

$$p_{\mathsf{A}}(x_1,\ldots,x_n) := \sum_{I\in\text{pattern}(\mathsf{A})} \prod_{i\in I} x_i, \qquad q_{\mathsf{A}}(x) := \sum_{I\in\text{pattern}(\mathsf{A})} x^{|I|} = p_{\mathsf{A}}(x,\ldots,x).$$

Then we have $|ESS(\mathsf{A})|= q_{\mathsf{A}}(1)$, and the mean support size of $ESS(\mathsf{A})$ is given by $\frac{q'_{\mathsf{A}}(1)}{q_{\mathsf{A}}(1)}$. Also, if P is a permutation matrix, then for $\mathsf{A}' := \mathsf{P}^\top \mathsf{A}\mathsf{P}$ we have $q_{\mathsf{A}'}(x) = q_{\mathsf{A}}(x)$.

As a consequence of our perturbation results and Lemma 1 we obtain the following.

Theorem 2.
(a) If $\{\mathsf{A}, \mathsf{B}_1, \ldots, \mathsf{B}_n\} \subset \mathcal{F}^r$, then for t big enough, we have $\mathsf{G}_t \in \mathcal{F}^r$.
(b) If $\mathsf{A} \in \mathcal{F}^r$ and $\{\mathsf{B}_1, \ldots, \mathsf{B}_n\} \subset \mathcal{F}$, then for t big enough and $\mathsf{x}_i = (x_{i,1}, \ldots x_{i,k_i})$ for $1 \leq i \leq n$, we have $p_{\mathsf{G}_t}(\mathsf{x}_1, \ldots, \mathsf{x}_n) = p_{\mathsf{A}}(p_{\mathsf{B}_1}(\mathsf{x}_1), \ldots, p_{\mathsf{B}_n}(\mathsf{x}_n))$, and thus $|ESS(\mathsf{G}_t)|= p_{\mathsf{A}}(q_{\mathsf{B}_1}(1), \ldots, q_{\mathsf{B}_n}(1))$.

One merit of this theorem is that it allows us to enrich an initial subset of $\mathcal{F}^r$ consisting of matrices with many ESSs with further matrices G_t of even more ESSs, that can themselves be used as building blocks for even larger matrices and so on and so on, and another that it makes it possible to keep track of those large numbers of ESSs just by evaluating polynomials. Writing down G_t explicitly requires to find the claimed large enough t experimentally, as we did not state universal lower bounds for t in the theorem. The following corollary can be of help in that regard, it deals with two cases where explicit lower bounds for t ensuring validity of the assertion of Theorem 2(b) can easily be constructed.

Corollary 3.
(a) Let $\mathsf{A} = \mathsf{E}_n - \mathsf{I}_n$ and $\mathsf{B}_i \in \mathcal{F}_{k_i}$ for $1 \leq i \leq n$, with $p_i := |ESS(\mathsf{B}_i)|$. Let $b := \max\limits_{1\leq i\leq n} \max\limits_{\mathsf{x}\in\Delta^{k_i}} \mathsf{x}^\top \mathsf{B}_i \mathsf{x}$. Then for $t > b$ the matrix G_t satisfies $|ESS(\mathsf{G}_t)|= \prod\limits_{1\leq i\leq n} p_i$.
(b) Let $\mathsf{A} \in \mathcal{F}^r_n$ and $\mathsf{B}_i \in \mathcal{F}_{k_i}$ for $1 \leq i \leq n$, with $\mathsf{p}_i^\top \mathsf{B}_i \mathsf{p}_i = 0$ for every i and every $\mathsf{p}_i \in ESS(\mathsf{B}_i)$. Then for $t = 1$ the matrix G_t satisfies $|ESS(\mathsf{G}_t)|= p_{\mathsf{A}}(q_{\mathsf{B}_1}(1), \ldots, q_{\mathsf{B}_n}(1))$.

There are further nice closure properties of G_t, like
(i) if $\{\mathsf{A}, \mathsf{B}_1, \ldots, \mathsf{B}_n\} \subset \mathcal{F}^r$ and all the local solutions of all the corresponding StQPs are global maximizers, with the global maxima being the same for all B_i, then for t large enough this will also be the case for G_t, i.e. all the local solutions of the StQP corresponding to G_t are global maximizers, which is interesting in the light of the results in [3, 4],
(ii) if $\{\mathsf{A}, \mathsf{B}_1, \ldots, \mathsf{B}_n\} \subset \mathcal{F}^r$ and the set $S(\mathsf{M}) := \{|K|\colon K \in \text{pattern}(\mathsf{M})\}$ of support sizes is a singleton for every $\mathsf{M} \in \{\mathsf{A}, \mathsf{B}_1, \ldots, \mathsf{B}_n\}$, with $S(\mathsf{B}_i)$ the same for all i, then for t large enough also $S(\mathsf{G}_t)$ will be a singleton, which can be used to improve results in [5],
(iii) if $\{\mathsf{A}, \mathsf{B}_1, \ldots, \mathsf{B}_n\} \subset \mathcal{F}^r$ are all cyclically symmetric, with B_i the same for all i, then G_t is congruent to a cyclically symmetric matrix via a permutation matrix.

We end this section with a short demonstration of our methods.

Example 1. A computer search restricted to cyclically symmetric matrices reveals a matrix $\mathsf{A} \in \mathcal{F}^r_9$ with first row $[0, 35, 12, 20, 20, 20, 20, 12, 35]$, satisfying $|ESS(\mathsf{A})|= 27$ (which is less than 30, the largest number of ESS known for 9×9 matrices). All the supports are of size 5, therefore $q_{\mathsf{A}}(x) = 27x^5$, and the pattern generating polynomial of A turns out to be

$$\begin{aligned} p_{\mathsf{A}}(\mathsf{x}) = {} & x_1x_2x_3x_4x_5 + x_2x_3x_4x_5x_6 + x_3x_4x_5x_6x_7 + x_4x_5x_6x_7x_8 + x_5x_6x_7x_8x_9 + x_6x_7x_8x_9x_1 \\ & + x_7x_8x_9x_1x_2 + x_8x_9x_1x_2x_3 + x_9x_1x_2x_3x_4 + x_1x_2x_3x_4x_7 + x_2x_3x_4x_5x_8 + x_3x_4x_5x_6x_9 \\ & + x_4x_5x_6x_7x_1 + x_5x_6x_7x_8x_2 + x_6x_7x_8x_9x_3 + x_7x_8x_9x_1x_4 + x_8x_9x_1x_2x_5 + x_9x_1x_2x_3x_6 \\ & + x_1x_2x_3x_6x_7 + x_2x_3x_4x_7x_8 + x_3x_4x_5x_8x_9 + x_4x_5x_6x_9x_1 + x_5x_6x_7x_1x_2 + x_6x_7x_8x_2x_3 \\ & + x_7x_8x_9x_3x_4 + x_8x_9x_1x_4x_5 + x_9x_1x_2x_5x_6 \end{aligned}$$

We want to construct a matrix $G_t \in \mathcal{F}^r_{12}$ having many ESSs with support size 5, where we allow for $\mathsf{B}_1, \ldots, \mathsf{B}_9$ the matrices $\mathsf{I}_2, \mathsf{I}_2, \mathsf{I}_2, \mathsf{I}_1, \mathsf{I}_1, \mathsf{I}_1, \mathsf{I}_1, \mathsf{I}_1, \mathsf{I}_1$ in some order. Note that $q_{\mathsf{I}_k}(x) = kx$. As it turns out, for $t = 1000000$ and the order specified above we have

$$104 = |ESS(\mathsf{G}_t)|= p_{\mathsf{A}}(2, 2, 2, 1, 1, 1, 1, 1, 1) = \max\{p_{\mathsf{A}}(\mathsf{x}) : \mathsf{x} \in \{1, 2\}^9, x_1 + \cdots + x_9 = 12\},$$

and accordingly $q_{\mathsf{G}_t}(x) = 104x^5$. That number, 104, is the largest number of ESS that we have been able to find for any symmetric matrix of order 12.

3. Improving lower bounds for the largest number of ESSs that a symmetric $n \times n$ matrix can have

Denote by U_n the largest number of ESSs that a symmetric $n \times n$ matrix can have. We know from [6, Theorem 2] that $\gamma := \lim_{n\to\infty} U_n^{1/n}$ exists and satisfies $U_n \leq \gamma^n$ for all n. To the best of our knowledge, the best lower bound for γ that has been found so far is $30^{1/9} \approx 1.4592$, originating from a 9×9 matrix with 30 ESSs, each of support size 3, which has been published in [6]. Building upon a certain small set of promising matrices found by computer search, our results enable us to show that for any order $n \geq 10$ there are matrices that lead to improved lower bounds for γ, in particular, $U_n^{1/n} \geq 70^{1/11} \approx 1.47142$ holds for $n \geq 10$, our best lower bound for γ being $536^{1/16} \approx 1.48106$.

4. Summary

This is part two of a series of three presentations, given by I. Bomze, W. Schachinger and R. Ullrich. The present part introduces a method for constructing Standard Quadratic Optimization Problems with many local solutions.

References

[1] Immanuel M. Bomze. Regularity versus degeneracy in dynamics, games, and optimization: a unified approach to different aspects. *SIAM Review*, 44(3):394–414, 2002.

[2] Immanuel M. Bomze and Etienne de Klerk. Solving standard quadratic optimization problems via linear, semidefinite and copositive programming. *Journal of Global Optimization*, 24(2):163–185, 2002.

[3] Immanuel M. Bomze, Werner Schachinger, and Reinhard Ullrich. From seven to eleven: completely positive matrices with high cp-rank. *Linear Algebra and its Applications*, 459:208–221, 2014.

[4] Immanuel M. Bomze, Werner Schachinger, and Reinhard Ullrich. New lower bounds and asymptotics for the cp-rank. *SIAM Journal on Matrix Analysis and Applications*, 36(1):20–37, 2015.

[5] Mark Broom. Bounds on the number of ESSs of a matrix game. *Mathematical Biosciences*, 167(2):163–175, 2000.

[6] Mark Broom, Chris Cannings, and Glenn T. Vickers. On the number of local maxima of a constrained quadratic form. *Proceedings of the Royal Society of London A*, 443:573–584, 1993.

[7] Mark Broom, Chris Cannings, and Glenn T. Vickers. Sequential methods for generating patterns of ESS's. *Journal of Mathematical Biology*, 32:597–615, 1994.

Proceedings of GOW'16, pp. 193 – 196.

Complexity of a Standard Quadratic Problem: a Theory-Guided Experimental Study *

Reinhard Ullrich[1], Immanuel Bomze[1] and Werner Schachinger[1]

[1] *Dept. of Statistics and OR, University of Vienna* reinhard.ullrich@univie.ac.at

Abstract This is the third part of a series circling around questions regarding StQPs with many solutions. We present an algorithm to find all maximizers for a given StQP, and focus on a class of matrices (the cyclically symmetric matrices), which has a natural potential for many coexisting local solutions. Furthermore we present extensive mathematical experiments using cyclically symmetric matrices of order 6,7 and 19.

Keywords: Quadratic optimization, Evolutionary stability

1. Algorithmic challenges: finding all ESSs of a given game

An algorithm to detect all ESSs of a given game (i.e., matrix) was presented in [1], but to our knowledge it has never been implemented and utilized. We implemented it with minor modifications and want to summarize it very briefly, focusing on the modifications and implementation issues.

Since for every support $I \subseteq N$, $I \neq \emptyset$, in a game with strategy set N, there exists at most one ESS x with $I(\mathsf{x}) = I$, the idea of the algorithm is to search for ESSs on (potentially) all the $2^n - 1$ supports of the game. This is done exploiting [1, Proposition 2.1], so a full support search only happens when the game does not exhibit any ESSs at all.

On every searched support the following two steps are carried out:

1. FINDEQ - Find a serious candidate for a strict local maximizer.
2. CHECKSTAB - Check if the candidate is really a strict local maximizer.

The algorithm we implemented differs from the original approach in two points. The original algorithm suggests to search the entire power set of N by selecting sets $\mathcal{J}_{\min}$ and $\mathcal{J}_{\max}$, being minimal (maximal) with respect to the set inclusion, and this is done iteratively (see p.317 in [1]). From an computational point of view - especially when one expects to find many ESSs - it is faster to search only "from the smallest supports upwards" (i.e. neglecting the $\mathcal{J}_{\max}$-sets), since it is cheaper to run FINDEQ on small supports. The second difference is the way the sets are chosen: instead of minimality with respect to the set inclusion we focus on the minimal number of elements. This is clearly more efficient to implement.

We coded this algorithm as multi-threaded for rational matrices using exact arithmetics, thus avoiding any roundoff errors. Our system setup is an Intel i7-4930K CPU with 6 kernels, 16GB RAM and a SSD harddisk, where the output of the algorithm is stored in a database. Note that this implementation has been utilized in the literature before, see [2] and [3], although in a slightly different context.

To give an estimate, this setup can search and record up to 5000 matrices of order $n = 9$ per second, but this depends largely on the complexity of the rational numbers and on the pattern (support structure numbers of ESSs) in the game.

*See also "Finding and analyzing hard instances of the StQP" and "Constructing StQPs with many local solutions"

FINDEQ - Finding a serious candidate for an ESS. In the original paper [1] the characterization of extremal equilibria with the help of polyhedra is used. To this end the vertices of a polyhedron have to be found, which can be accomplished by linear programming techniques. This original method is far too slow for practical applications, so we adapted it by presenting a new method, admitting every NES as candidate for an ESS, if it is the only NES with the currently considered support.

CHECKSTAB - Checking which candidates are really ESS. Verification of the ESS property of the candidates is a quite cumbersome task, since it potentially involves checks for copositivity [1, Section 3]. It is interesting to note that empirically only a small percentage of ESSs is detected by copositivity, usually a check for positive definiteness is sufficient in most cases. This is in line with genericity, but as we are working on discrete data, it is not straightforward.

2. Cyclically symmetric matrices and restrictions on them

We will employ symmetry transformations of the coordinates of vectors given by cyclic permutation (see also [2], where this notation has been introduced), denoting by $a \oplus b$, $a \ominus b$ and $a \odot b$ the result of addition, subtraction and multiplication modulo n. To keep in line with standard notation, we consider the remainders $[1\!:\!n]$ instead of $[0\!:\!n-1]$, e.g. $1 \oplus (n-1) = n$. To be more precise, let P_i be the square $n \times n$ permutation matrix which effects $\mathsf{P}_i\mathsf{x} = [x_{i\oplus j}]_{j\in[1:n]}$ for all $\mathsf{x} \in \mathbb{R}^n$ (for example, if $n = 3$ then $\mathsf{P}_2\mathsf{x} = [x_3, x_1, x_2]^\top$). Obviously $\mathsf{P}_i = (\mathsf{P}_1)^i$ for all integers i (recall P^{-3} is the inverse matrix of PPP), $\mathsf{P}_i^\top = \mathsf{P}_{n-i} = \mathsf{P}_i^{-1}$ and $\mathsf{P}_n = \mathsf{I}_n$. A *circulant matrix* $\mathsf{S} = \mathsf{C}(\mathsf{a})$ based on a vector $\mathsf{a} \in \mathbb{R}^n$ (as its last column rather than the first) is given by $\mathsf{S} = [\mathsf{P}_{n-1}\mathsf{a}, \mathsf{P}_{n-2}\mathsf{a}, \ldots, \mathsf{P}_1\mathsf{a}, \mathsf{a}]$. If $\mathsf{S} = \mathsf{C}(\mathsf{a})$ is symmetric it is called *cyclically symmetric*, and that holds whenever $a_i = a_{n-i} \quad$ for all $i \in [1\!:\!n-1]$.

It is easy to see that any circulant matrix $\mathsf{S} = \mathsf{C}(\mathsf{a})$ satisfies $\mathsf{P}_i^\top \mathsf{S} \mathsf{P}_i = \mathsf{S}$ for all $i \in [1\!:\!n]$, and this is the key to their use in finding matrices with many ESSs.

Let a problem $\max_{\mathsf{x}\in\Delta^n} \mathsf{x}^\top \mathsf{S}\mathsf{x}$ be given and, let M be an arbitrary permutation matrix with $\mathsf{M}^\top \mathsf{S} \mathsf{M} = \mathsf{S}$. If x^* is a solution to the problem , then $\mathsf{M}^\top \mathsf{x}^*$ is also solutions of the problem. The two vectors need not differ from each other, though, if additional symmetry prevails.

Due to the structure of the matrices $\mathsf{C}(\mathsf{a})$, any found ESS leads to further $n-1$ ESSs, where the involved permutation matrices are $\mathsf{P}_1, \ldots, \mathsf{P}_{n-1}$, if we are cautious enough to break abovementioned symmetry. In this case, the number of ESSs found in a game with a cyclically symmetric matrix must be the sum of multiples of the factors of n (or 0 or 1). When n is prime, this leads to the nice property that the game has 0,1 or a multiple of n ESSs.

We started experimenting to search cyclically symmetric matrices $\mathsf{C}(\mathsf{a})$ for ESSs on different n, where we set $a_n = 0$ in every case. This reduces the degrees of freedom and does not do any harm, since constants can be added to columns of a game matrix without changing the game, see [4]. So in total we have $\lfloor \frac{n}{2} \rfloor$ variables for every n, and for these variables we allowed integers.

This approach leads to good results for smaller n (say ≤ 12), but for larger n this procedure became prohibitively slow. For these instances, our idea is not only to exploit cyclic permutations matrices inherent to the matrices $\mathsf{C}(\mathsf{a})$, but to enforce $\mathsf{M}^\top \mathsf{C}(\mathsf{a}) \mathsf{M} = \mathsf{C}(\mathsf{a})$ for one or more arbitrary permutation matrices M. This leads to more restrictions on the degrees of freedom for constructing the vector a.

Empirically for $n \in [13 : 22]$ it turned out that using the following construction is a good choice. Define $n \times n$ matrices $\mathsf{E}_{i,j} = \mathsf{e}_i \mathsf{e}_j^\top$ and let for k and n mutually prime

$$\mathsf{M}_k = \sum_{i=1}^{n} \mathsf{E}_{k\odot i, i}. \tag{1}$$

If we now require $\mathsf{M}_k^\top \mathsf{C}(\mathsf{a})\mathsf{M}_k = \mathsf{C}(\mathsf{a})$, then further restrictions on the a_i result. Therefore, if too many restrictions are imposed, then a becomes trivial in that a_i is constant across all $i \in [1 : n-1]$, and if we do not use enough restriction by this construction, all a_i may have different values. A careful choice of these further restrictions was successfully applied to various cyclically symmetric matrices for different n, which will be detailed in the talk.

3. Experimental studies on cyclically symmetric matrices

The theory detailed before enables us "more than just educated guesses" on the solution complexity of StQPs, which were impossible to obtain by naive random sampling or brute force enumeration: indeed, by these methods, it is highly unlikely to encounter even one of the interesting instances. To gain insight how a large and representative number of matrices with a potentially complex solution structure behave, we therefore developed the resulting refined mathematical experiments which provide an illustration and a statistical evaluation for cyclically symmetric matrices for three different n, namely $n = 6$, $n = 7$ and $n = 19$.

All the matrices we generated and tested are of the form $\mathsf{C}(\mathsf{a})$ where

$$\begin{array}{lcll} \mathsf{a} & = & (a,b,c,b,a,0)^\top & \text{for } n = 6\,, \\ \mathsf{a} & = & (a,b,c,c,b,a,0)^\top & \text{for } n = 7\,, \text{ and} \\ \mathsf{a} & = & (a,b,b,c,b,c,a,a,c,c,a,a,c,b,c,b,b,a,0)^\top & \text{for } n = 19\,. \end{array}$$

Note that $\mathsf{M}_7^\top \mathsf{C}(\mathsf{a})\mathsf{M}_7 = \mathsf{C}(\mathsf{a})$ is satisfied in the latter case $n = 19$.

Since multiplying a matrix with a positive factor does not change the amount and structure of an ESS, just the direction and not the length of a is important. Therefore we used points on the unit sphere as input for a, employing the following procedure: take a Lambert projection of the unit sphere, which is an equal-area projection (to prevent graphical biases in the following illustrations) with $\theta \in [-\pi, \pi]$ on the horizontal and $\sin(\phi) \in [-1, 1]$ on the vertical axis, where (ϕ, θ) are polar coordinates representing $x = \cos(\phi)\cos(\theta), y = \cos(\phi)\sin(\theta), z = \sin(\phi)$.

We performed another coordinate transformation, building an orthonormal system such that the north pole of the sphere points into direction $\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$, i.e. $\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \frac{1}{\sqrt{6}} \begin{pmatrix} \sqrt{3} & 1 & \sqrt{2} \\ -\sqrt{3} & 1 & \sqrt{2} \\ 0 & -2 & \sqrt{2} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$.

The reason for this transformation lies in a remarkable symmetry around direction $(1,1,1)^\top$ for $n \in \{7, 19\}$.

To approximate a uniform distribution on the sphere we generated (approximately) equidistant data points on the θ-$\sin(\phi)$-plane, giving 1201×401 data points. Note that points on the boundary are calculated multiple times due to periodicity, but we did not correct for this.

While all three values of n will be discussed in the talk, here we only provide details for $n = 7$, where the left part of Figure 1 shows the support-pattern for the entire search space.

It is interesting to see that all the areas for the different ESSs and supports are clearly separated (for all n), and they are connected for all supports, except for zero ESSs. The areas with one ESS lie in the positive orthant (w.r.t. the standard basis) for all n. The areas with zero ESSs have measure zero on the unit sphere, they form arcs and isolated points on the θ-$\sin(\phi)$-plane. But these points are rarely hit due to the construction of the data (isolated points of zero ESSs can be seen in the plots upon closer look).

For $n \in \{7, 19\}$ there exists a remarkable symmetry. Next to the area with one ESS there exist three areas with the highest number of ESSs. For $n = 7$ these areas with 14 ESSs having different support structure. The right part of Figure 1 "zooms" into the plot on the left side, in the sense that we want to analyze the structure of one of these three areas in more detail. We generated 1015×501 data points in the interval $\theta \in [-0.75\pi, -0.65\pi]$ and $\sin(\phi) \in [0.765, 0.92]$, which again simulates a uniform distribution. The separation of the different support pattern is clearly visible, with a particular point close to $(-0.68\pi, 0.88)$ where all the separating lines meet.

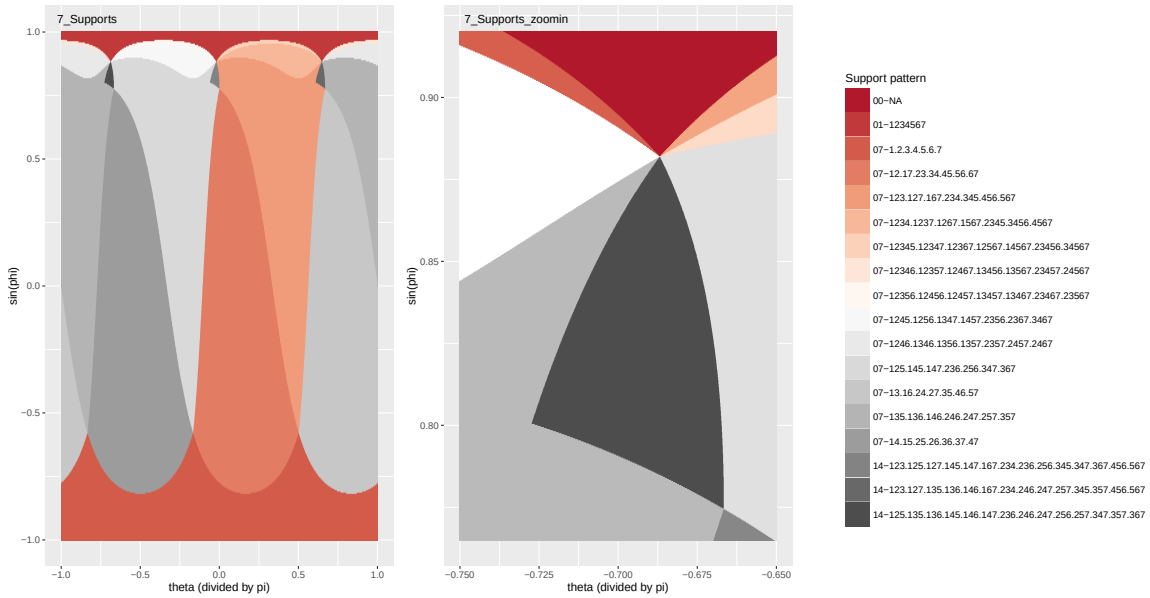


Figure 1. Entire search space and zoom in for $n = 7$

4. Summary

This is part three of a series of three presentations, given by I. Bomze, W. Schachinger and R. Ullrich. The present part describes an algorithm to find all strict local solutions of a given StQP and introduces cyclically symmetric matrices, which have the potential to have many coexisting solutions. An extensive experimental study for cyclically symmetric matrices of order 6,7 and 19 is presented.

References

[1] Immanuel M. Bomze. Detecting all evolutionarily stable strategies. *Journal of Optimization Theory and Applications*, 75(2):313–329, 1992.

[2] Immanuel M. Bomze, Werner Schachinger, and Reinhard Ullrich. From seven to eleven: completely positive matrices with high cp-rank. *Linear Algebra and its Applications*, 459:208–221, 2014.

[3] Immanuel M. Bomze, Werner Schachinger, and Reinhard Ullrich. New lower bounds and asymptotics for the cp-rank. *SIAM Journal on Matrix Analysis and Applications*, 36(1):20–37, 2015.

[4] Jörgen W. Weibull. *Evolutionary game theory*. MIT Press, Cambridge, Mass., 1995.

Topic Index

Author Index

federico.cabassi@studenti.unipr.it, 129
Felipe Müller
Universidade Federal de Santa Maria, Santa Maria, RS, Brazil, felipe@inf.ufsm.br, 141
Fernanda Raupp
National Laboratory for Scientific Computing, Brazil, fernanda@lncc.br, 73
Frédéric Messine
LAPLACE-ENSEEIHT-INPT, University of Toulouse, France, Frederic.Messine@n7.fr, 149
Francisco Pinillos Nieto
Federal University of Rio de Janeiro, Brazil, paquito@cos.ufrj.br, 93
G Ortega
University of Almería , Spain, gloriaortega@ual.es, 25
Gábor Szabó
Department of Optics and Quantum Electronics, University of Szeged, H-6720 Szeged, Dóm tér 9, Hungary, szabog@physx.szte.hu, 159
Gabriele Eichfelder
Technische Universität Ilmenau, Institute for Mathematics, Weimarer Straße 25, 98693 Ilmenau, Germany, gabriele.eichfelder@tu-ilmenau.de, 57
Gilles Trombettoni
LIRMM, Université de Montpellier, France, Gilles.Trombettoni@lirmm.fr, 133
Gražina Gimbutienė
Institute of Mathematics and Informatics, Vilnius University, 4 Akademijos str., Vilnius, Lithuania, grazina.gimbutiene@mii.vu.lt, 53
Gustavo Dias
CNRS LIX Ecole Polytechnique, Palaiseau, France, dias@lix.polytechnique.fr, 13
H. Rocha
INESCC & FEUC, Universidade de Coimbra, Coimbra, Portugal, hrocha@mat.uc.pt, 155
Helena den-Haan
Universidad Católica San Antonio de Murcia, Murcia, Spain, hdenhaan@ucam.edu, 105
Horacio Pérez Sánchez
Universidad Católica San Antonio de Murcia, Murcia, Spain, hperez@ucam.edu, 105
Ignacio Araya
Pontificia Universidad Católica de Valparaíso, Escuela de Ingeniería Informática, Chile, ignacio.araya@pucv.cl, 109
Immanuel M. Bomze
ISOR, Univ.of Vienna, immanuel.bomze@univie.ac.at, 185, 189, 193
Inmaculada García
University of Málaga, Spain, igarciaf@uma.es, 33
Ivo Nowak
Hamburg University of Applied Sciences, Berliner Tor 21 20099 Hamburg, Germany, ivo.nowak@haw-hamburg.de, 65
J. M. Dias
INESCC & FEUC, Universidade de Coimbra, Coimbra, Portugal, joana@fe.uc.pt, 155
J.D. Alvarez
University of Almeria, Almeria, Spain, jhervas@ual.es, 61
J.F.R. Herrera
Dept. Informatics, University of Almería (CeiA3), Spain, juanfrh@ual.es, 45
J.M.G. Salmerón
Dept. Informatics, University of Almería (CeiA3), Spain, josemanuel@ual.es, 41, 45
JA Martínez
University of Almería , Spain, jmartine@ual.es, 25
Jan Kronqvist
Process Design and Systems Engineering, Faculty of Science and Engineering, Åbo Akademi University, 20500 Åbo, Finland, jan.kronqvist@abo.fi, 117, 101
Ji Ma
ENAC-Université de Toulouse, 7 avenue Edouard Belin, 31055 Toulouse cedex 4, France, ji.ma@recherche.enac.fr, 113
JJ Moreno
University of Almería , Spain, juanjonrg@gmail.com, 25
Jon Lee
Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI, jonxlee@umich.edu, 81
Jordan Ninin
Lab-STICC, ENSTA-Bretagne, Brest, France, jordan.ninin@ensta-bretagne.fr, 175
José Fernández
University of Murcia, Murcia, Spain, josefdez@um.es, 29, 21
Juana L. Redondo
Dept. Informatics, University of Almería (CeiA3), Spain, jlredondo@ual.es, 21, 29, 41, 61, 69, 105
Julia Niebling
Technische Universität Ilmenau, Institute for Mathematics, Weimarer Straße 25, 98693 Ilmenau, Germany, julia.niebling@tu-ilmenau.de, 57
Julius Žilinskas
Institute of Mathematics and Informatics, Vilnius University, Akademijos 4, LT-08663 Vilnius, Lithuania, julius.zilinskas@mii.vu.lt, 121, 153, 163
Kristóf Kovács
Budapest University of Technology and Economics, Budapest, Hungary, kkovacs@math.bme.hu, 137
L.G. Casado
Dept. Informatics, University of Almería (CeiA3), Spain, leo@ual.es, 33, 41, 45, 163
Leo Liberti
CNRS LIX, Ecole Polytechnique, Palaiseau, France, liberti@lix.polytechnique.fr, 13
M. Berenguel
University of Almeria, Almeria, Spain, beren@ual.es, 61
M.C. Lopes
IPOC-FG, EPE, Coimbra, Portugal, mclopes@ipocoimbra.min-saude.pt, 155
M. Fernanda P. Costa
Centre of Mathematics, University of Minho, Braga, Portugal, mfc@math.uminho.pt, 97
Mária Csete
Department of Optics and Quantum Electronics, University of Szeged, H-6720 Szeged, Dóm tér 9, Hungary, mcsete@physx.szte.hu, 159
Marat S. Mukhametzhanov
University of Calabria, Italy, marat@dimes.unical.it, 85
Marcel Mongeau
ENAC-Université de Toulouse, 7 avenue Edouard Belin, 31055 Toulouse cedex 4, France, marcel.mongeau]recherche.enac.fr, 113
Marcia Fampa
Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, fampa@cos.ufrj.br, 73, 93, 141
Marco Locatelli
Dipartimento di Ingegneria dell'Informazione, Università di Parma, Parma, Italy, marco.locatelli@unipr.it, 129
Marleen Hermelink
Operations Research and Logistics, Wageningen University, Hollandseweg 1, 6706 KN, Wageningen, The Netherlands, marleen.hermelink@wur.nl, 77
Martin de la Gorce
LIGM, Ecole des Ponts Paristech, Université Paris Est, France, 133
Milan Hladík
Charles University in Prague, Czech Republic, milan.hladik@matfyz.cz, 17

http://apolo.dps.uminho.pt/gow16

University of Minho
School of Engineering

University of Minho
School of Sciences
Centre of Mathematics