# PROCEEDINGS OF THE TOULOUSE GLOBAL OPTIMIZATION WORKSHOP

Edited by

S. Cafieri
Ecole Nationale de l’Aviation Civile, Toulouse, France

B. G.-Tóth
Budapest University of Technology and Economics, Budapest, Hungary

E.M.T. Hendrix
University of Málaga, Málaga, Spain

L. Liberti
Ecole Polytechnique, Paris, France

F. Messine
ENSEEIHT, Toulouse, France

# Preface

Global Optimization Workshops are organized periodically and rather spontaneously by members of the Global Optimization scientific community. Although there is no proper steering committee to speak of, the tradition continues unabated ever since Szeged (Hungary 1995) and Florence (Italy, 1999) and going on to Hanmer Springs (New Zealand, 2001), Santorini (Greece, 2003), San José (Spain, 2005), Mykonos (Greece, 2007), Skukuza (South Africa, 2008). The TOulouse Global Optimization workshop (TOGO 2010) adds a new opportunity to meet on this interesting subject.

In 2009 a new, larger scale format was introduced with the first World Congress of Global Optimization. Although Global Optimization is certainly mature enough to draw enough people for a congress, some of us still relish the small-scale, single-track meetings where everyone gets to meet everyone else. As is customary, a special issue of the Journal of Global Optimization will be dedicated to this workshop.

This year's workshop is special in that it celebrates Pierre Hansen's 70th birthday. Pierre has been an exceptionally important contributor to the development of Global Optimization. His works on symbolic methods for global optimization, nonlinear clustering problems, and extremal polygons all became classics in the field. He also contributed to found the related field of reformulations in mathematical programming, with a seminal paper co-signed with another one of our plenary speakers, Charles Audet. People have lost count of the number of papers written by Pierre; we are sure it must be several hundreds. His H-index, according to GoogleScholar, is over 30: after reading his publications list we once heard a colleague from another field say, "not only does this guy publish like crazy: but people actually read his papers, too!". The breadth of his interests could easily be termed infinite, at least on the human scale. We invite our attendees to seize this opportunity to play the "ask Pierre" game: mention a topic and bet on how many papers he has on the subject. Among other awards, Pierre has been the recipient of the prestigious EURO gold medal. Pierre is fond of saying that he owes his success to the fantastic quality of his co-authors. We – at least those of us who have had the chance of being Pierre's co-authors – think this sentence should be turned around: much of our highest quality work is due to Pierre's encouragement and seemingly inexhaustible reserve of excellent ideas. Happy birthday Pierre, and keep up the good work!

Eligius Hendrix (Málaga, Spain)
Leo Liberti (Paris, France)
Frédéric Messine (Toulouse, France)
*TOGO 2010 Co-chairs*

Scientific committee:

- Robert Baker Kearfott, Lafayette, Lousiana
- Sonia Cafieri, Toulouse, France
- Tibor Csendes, Szeged, Hungary
- Chris Floudas, Princeton, New Jersey
- Inmaculada García, Almería, Spain
- Eligius Hendrix, Malaga, Spain
- Jean-Baptiste Hiriary-Urruty, Toulouse, France
- Jean-Bernard Lasserre, Toulouse, France
- Sven Leyffer, Chicago, Illinois
- Leo Liberti, Paris, France
- Jeff Linderoth, Madison, Wisconsin
- Marco Locatelli, Turin, Italy
- Nelson Maculan, Rio de Janeiro, Brazil
- Frederic Messine, Toulouse, France
- Panos Pardalos, Gainesville, Florida
- Fabio Schoen, Florence, Italy
- Hanif Sherali, Blacksburg, Virginia
- Fabio Tardella, Rome, Italy
- Tapio Westerlund, Abo, Finland

Local organisers:

- Sonia Cafieri, Toulouse, France
- Nicolas Durand, Toulouse, France
- Boglárka G.-Tóth, Budapest, Hungary
- David Gianazza, Toulouse, France
- Eligius Hendrix, Málaga, Spain
- Didier Henrion, Toulouse, France
- Leo Liberti, Paris, France
- Kader Merakeb, Toulouse, France
- Frédéric Messine, Toulouse, France
- Marcel Mongeau, Toulouse, France
- Antonio Mucherino, Lille, France
- Giacomo Nannicini, Pittsburgh, Pennsylvania
- Jordan Ninin, Toulouse, France
- Evelyne Rayssac, Paris, France

Sponsors:

- Ecole Nationale de l'Aviation Civile
- Ecole Nationale Supérieure d'Electrotechnique, d'Electronique, d'Informatique, d'Hydraulique et des Télécommunications
- Groupe d'Etudes et de Recherche en Analyse des Décisions
- Institut de Mathématiques de Toulouse
- Institut de Recherche en Informatique de Toulouse
- Laboratoire d'Informatique de l'Ecole Polytechnique
- Microsoft-CNRS Chair "Optimization and Sustainable Development"
- Université Paul Sabatier
- Institut National Polytechnique de Toulouse
- LAAS-CNRS

# Contents

## Plenary Talks

## Extended Abstracts

# PLENARY TALKS

Proceedings of TOGO 2010, pp. 3 – 6.

# Optimization problems in planar geometry*

Charles Audet

*GERAD and Département de mathématiques et de génie industriel, École Polytechnique de Montréal, C.P. 6079, Succ. Centre-ville, Montréal (Québec), H3C 3A7 Canada,* Charles Audet@gerad.ca

**Abstract** Attributes such as perimeter, area, diameter, sum of distances between vertices and width can be evaluated for every planar convex polygon. Fixing one of these attributes while minimizing of maximizing another defines families of optimization problems. Some of these problems have a trivial solution, and several others have been solved, some since the Greeks, by geometrical reasoning. During the last four decades, this geometrical approach has been complemented by global optimization methods. This combination allowed solution of instances than could be solved by any one of these two approaches alone. This talk surveys research on that topic, and proposes directions for further work.

**Keywords:** Extremal problems, global optimization, convex polygon, perimeter, diameter, area, sum of distances, width

## 1. Introduction

Consider a $n$-sided convex polygon $V_n$ in the Euclidean plane. Let $A_n$ denote its area, $P_n$ its perimeter, $D_n$ its diameter, $S_n$ the sum of distances between all pairs of its vertices and $W_n$ its width. Maximizing or minimizing any of these quantities while setting another to a fixed value defines ten pairs of extremal problems.

These problems were first surveyed in [5], and then some solutions were later updated in [6]. Usually, one problem from each pair has a trivial solution or no solution at all. For example, Zenodorus (200-140 b.c.) showed that the regular polygons have maximal area for a given value of the perimeter, but the question of minimizing the area given a fixed perimeter is trivial, as the area can be made arbitrarily close to zero.

Simple formulations of extremal problems for convex polygons are easily obtained by denoting the consecutive vertices of the $n$-sided polygon $V_n$ by $v_i = (x_i, y_i)$. Then

. $\displaystyle A_n = \frac{1}{2}\left|\sum_{i=1}^{n}(y_{i+1} - y_i)(x_{i+1} + x_i)\right|,$

. $\displaystyle P_n = \sum_{i=1}^{n} \|v_{i+1} - v_i\|,$

. $\displaystyle D_n = \max_{i<j} \|v_i - v_j\|,$

. $\displaystyle S_n = \sum_{i<j} \|v_i - v_j\|,$

*This work is supported by NSERC grant 239436-05,

$$. \; W_n = \min_i \max_{j \neq i, i+1} \frac{|(y_{j+1} - y_j)x_i + (x_j - x_{j+1})y_i + x_{j+1}y_j - x_j y_{j+1}|}{\|v_{j+1} - v_j\|},$$

where the indices $i+1$ and $j+1$ are taken modulo $n$, $|.|$ denotes absolute value and $\|.\|$ the Euclidean norm. Expressions of the objective function and constraints easily follow.

## 2. Currently known solutions

Minimizing a first attribute while fixing a second one is equivalent to maximizing the second one while fixing the first one. Therefore, the present work only considers the maximization questions. Table 1 summarizes the current known solutions to these 20 extremal problems for convex polygons, and provides references to where the solutions may be found. Each line of the table corresponds to a fixed attribute, while each column indicates which attribute is maximized.

| | max $P_n$ | max $A_n$ | max $D_n$ | max $S_n$ | max $W_n$ |
|---|---|---|---|---|---|
| $P_n$=1 | — | Regular<br>$\sim$ 180 BC<br>Zenodorus | Trivial:<br>Segment | Segment<br>2008 Larcher &<br>Pillichshammer<br>2008 [16] | Reuleaux for $n$<br>with odd factor<br>and n=4. 2009 [4] |
| $A_n$=1 | Trivial:<br>flat | — | Trivial:<br>flat | Trivial:<br>flat | open |
| $D_n$=1 | Reuleaux for $n$<br>with odd factor<br>and $n = 4, 8$<br>Tamvakis 1987 [21]<br>Datta 1997 [11]<br>2007 [3] | Regular for odd $n$<br>Reinhardt 1922 [19]<br>and $n \leq 10$.<br>Graham 1975 [13]<br>2002 [8], Foster &<br>Szabo 2007 [12]<br>Mossinghoff 2006<br>[18], Henrion &<br>Messine 2010 [15] | — | $n = 3, 4, 5, 6, 7$<br>2008 [1] | Reuleaux for $n$<br>with odd factor<br>and $n = 4$.<br>Bezdek &<br>Fodor [10] |
| $S_n$=1 | Segment<br>Larcher &<br>Pillichshammer<br>2008 [16] | open | Trivial:<br>flat | — | open |
| $W_n$=1 | Trivial:<br>slice | Trivial:<br>slice | Trivial:<br>slice | Trivial:<br>slice | — |

*Table 1.* Convex polygons with maximal attribute

Several of these problems have a trivial solution, and only a few of the non-trivial ones are solved for every value of $n$. Most of the non-trivial ones have known solutions in the cases where $n$ is very small, or when $n$ is an odd number, or when it has an odd factor. Some of these solutions were obtained numerically, using recent global optimization algorithms. In particular QP [2], a branch and cut algorithm for nonconvex quadratically constrained optimization, IBBA [17], an interval analysis branch and bound algorithm for nonlinear programming, and very recently, GloptiPoly [14], a semidefinite programming approach for polynomial optimization.

Adding the additional constraint that the polygons are equilateral leads to different optimization problems. The cases where the solutions are the regular or clipped-Reuleaux [20] polygons have the equilateral property, and therefore remain optimal. Table 2 details the currently know solutions to these problems. The most recent results are for the maximization of the perimeter, the area and the diameter of unit width equilateral polygon. In the non trivial case (when the number of sides is odd), it is shown that the optimal polygon are arbitrarily close to symmetrical trapezoids.

| | $\max P_n$ | $\max A_n$ | $\max D_n$ | $\max S_n$ | $\max W_n$ |
|---|---|---|---|---|---|
| $P_n$=1 | — | Regular<br>Zenodorus<br>$\sim$ 180 BC | Trivial:<br>Segment | open | Reuleaux for $n$<br>with odd factor.<br>2009 [4] |
| $A_n$=1 | Trivial:<br>flat | — | Trivial:<br>flat | open | open |
| $D_n$=1 | Reuleaux for $n$<br>with odd factor<br>Vincze 1950 [22]<br>and $n = 4, 8$<br>2004 [7] | Regular for odd $n$<br>Reinhardt 1922 [19]<br>and $n = 4$. | — | $n = 3, 5$<br>2008 [1] | Reuleaux for $n$<br>with odd factor<br>and $n = 4$.<br>2000 Bezdek &<br>Fodor [10] |
| $S_n$=1 | open | open | Trivial:<br>flat | — | open |
| $W_n$=1 | Trivial for even $n$<br>Trapezoid for<br>odd $n$ 2010 [9] | Trivial for even $n$<br>Trapezoid for<br>odd $n$ 2010 [9] | Trivial for even $n$<br>Trapezoid for<br>odd $n$ 2010 [9] | open<br>for odd $n$ | — |

*Table 2.* Equilateral convex polygons with maximal attribute

The presentation will discuss recent progress on some of these problems, and will propose potential research directions.

# References

[1] C. Audet, A. Guillou, P. Hansen, F. Messine, S. Perron *The Small Hexagon and Heptagon with Maximum Sum of Distances between Vertices*, Les Cahiers du GERAD G-2008-20, 2008.

[2] C. Audet, P. Hansen, B. Jaumard, and G. Savard. *A branch and cut algorithm for nonconvex quadratically constrained quadratic programming*. Mathematical Programming, 87(1, Ser. A):131–152, 2000.

[3] C. Audet, P. Hansen, F. Messine, *The small octagon with longest perimeter*, Journal of Combinatorial Theory, Series A, Vol. 114, pp. 135–150, 2007.

[4] C. Audet, P. Hansen, F. Messine, *Isoperimetric Polygons of Maximal Width*, Discrete & Computational Geometry, 41(1): 45-60, 2009.

[5] C. Audet, P. Hansen, F. Messine, *Extremal problems for convex polygons*, Journal of Global Optimization, Vol. 38, pp. 163–179, 2007.

[6] C. Audet, P. Hansen, and F. Messine. Extremal problems for convex polygons - an update. In P.M. Pardalos and T.F. Coleman, editors, *Lectures on Global Optimization*, volume 55 of *Fields Institute Communications*. American Mathematical Society, 2009.

[7] C. Audet, P. Hansen, F. Messine, S. Perron, *The minimum diameter octagon with unit-length sides: Vincze's wife's octagon is suboptimal*, Journal of Combinatorial Theory, Series A, Vol. 108, pp. 63–75, 2004.

[8] C. Audet, P. Hansen, F. Messine, and J. Xiong, *The largest small octagon*, Journal of Combinatorial Theory, Series A, Vol. 98, No. 1, pp. 46–59, 2002.

[9] C. Audet, J. Ninin, *The Maximal perimeter, diameter and area of equilateral unit-width convex polygons*, Les Cahiers du GERAD G-2010, in preparation, 2010.

[10] A. Bezdek and F. Fodor, *On convex polygons of maximal width*, Archiv der Mathematik, Vol. 74, No. 1, pp. 75–80, 2000.

[11] B. Datta, *A discrete isoperimetric problem*, Geometriae Dedicata, Vol. 64, pp. 55–68, 1997.

[12] J. Foster, T. Szabo, *Diameter graphs of polygons and the proof of a conjecture of Graham*, Journal of Combinatorial Theory, Series A, Vol. 114, No. 8, pp. 1515–1525, 2007.

[13] R. L. Graham, *The largest small hexagon*, Journal of Combinatorial Theory, Series A, Vol. 18, pp. 165–170, 1975.

[14] D. Henrion, J. B. Lasserre, J. Löfberg. *GloptiPoly 3: moments, optimization and semideÞnite programming*, Optimization Methods and Software, Vol. 24(4-5), pp. 761–779, 2009.

[15] D. Henrion, F. Messine. *Finding largest small polygons with GloptiPoly*, Proceedings of TOGO 2010.

[16] G. Larcher, F. Pillichshammer, *The sum of distances between vertices of a convex polygon with unit perimeter*, American Mathematical Monthly, Vol. 115, pp. 350–355, 2008.

[17] F. Messine *Deterministic Global Optimization using Interval Contraint Propagation Techniques*, RAIRO Oper. Res., Vol. 38, No. 4, pp. 277–294, 2004.

[18] M.J. Mossinghoff, *Isodiametric Problems for Polygons*, Discrete and Computational Geometry, Vol. 36, pp. 363–379, 2006.

[19] K. Reinhardt, *Extremale polygone gegebenen durchmessers*, Jahresber. Deutsch. Math. Verein, Vol. 31, pp. 251–270, 1922.

[20] F. Reuleaux, *The Kinematics of Machinery*, translation of german original, New York: Dover, 1963.

[21] N.K. Tamvakis, *On the perimeter and the area of the convex polygon of a given diameter*, Bull. Greek Math. Soc., Vol. 28, pp. 115–132, 1987.

[22] S. Vincze, *On a geometrical extremum problem*, Acta Sci. Math. Szeged, Vol. 12, pp. 136–142, 1950.

# Valid inequalities for sets defined by multilinear functions

Pietro Belotti[1], Andrew J. Miller[2], and Mahdi Namazifar[3]

[1] *Lehigh University, 200 W. Packer Ave, Bethlehem PA 18018.* belotti@lehigh.edu

[2] *Institut de Mathématiques de Bordeaux (IMB), 351 cours de la Libération, 33405 Talence, France.* Andrew.Miller@math.u-bordeaux1.fr

[3] *University of Wisconsin, 3239 Mechanical Engineering Building, 1513 University Avenue, Madison WI 53706, USA.* namazifar@wisc.edu

We are interested in finding the convex hull of the *multilinear* set

$$M_n = \{x \in R^{n+1} : x_{n+1} = \prod_{i=1}^{n} x_i,\ \ell_i \leq x_i \leq u_i\ \forall i = 1, 2 \ldots, n+1\},$$

with $\ell_i$ and $u_i$ constants. Computing the convex hull of $M_n$ is of great theoretical and practical importance in Global Optimization. An important special case is the one in which $x_{n+1}$ is not *explicitly* bounded, i,e., $\ell_{n+1} = \prod_{i=1}^{n} \ell_i$ and $u_{n+1} = \prod_{i=1}^{n} u_i$, which we denote as $M_n^\star$.

We review some of the literature about this important problem, and highlight some recent results that help understand the structure of $M_n$ and of $M_n^\star$ and their impact on state-of-the-art global optimization solvers.

We describe a family of inequalities for $M_2$, i.e., when $\ell_3 > \ell_1 \ell_2$ and $u_3 < u_1 u_2$. Together with the well known inequalities introduced by McCormick, these inequalities are valid for the convex hull of $M_2$. There are infinitely many such inequalities, given that the convex hull of $M_2$ is not, in general, a polyhedron. The generalization to $M_n$ for $n > 2$ is straightforward, and allows us to define strengthened relaxations for these higher dimensional sets as well.

# (Globally) minimizing the distance to a geometrical property of points

Emilio Carrizosa

*Universidad de Sevilla, Sevilla, Spain,* ecarrizosa@us.es

**Keywords:** D.C. optimization, distances minimization

Let $\mathcal{F}_{k,n}$ be a non-empty subset of $\Re^n \times \overset{(k)}{\ldots} \times \Re^n$. We are interested in sets $\mathcal{F}_{k,n}$ defining a geometrical property. Examples of such geometrical property are $r$-coincidence (the set $\{x_1, \ldots, x_k\}$ has cardinality $r$), collinearity (the set $\{x_1, \ldots, x_k\}$ is contained in a line in $\Re^n$), cocircularity, cohyperplanarity (the set is contained in a spherical surface or a hyperplane), linear separability (the sets $\{x_1, \ldots, x_r\}$ and $\{x_{r+1}, \ldots, x_k\}$ are linearly separable), etc.

Given $\mathbf{x} = (x_1, \ldots, x_k)$ in $\Re^n \times \overset{(k)}{\ldots} \times \Re^n$, we seek a perturbation vector $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_k) \in \Re^n \times \overset{(k)}{\ldots} \times \Re^n$ of $\mathbf{x}$ such that $\mathbf{x} + \varepsilon \in \mathcal{F}_{k,n}$ and the intensity of the perturbation $\varepsilon$ is minimized. The intensity of the perturbation is measured by a nondecreasing function $\phi$ of the vector of norms $\|\varepsilon_j\|$.

Different sets $\mathcal{F}_{k,n}$ as well as different choices of $\phi$ and $\| \cdot \|$ yield optimization problems with rather different properties. In particular, in many cases we face multimodal optimization problems which can be written as optimizing a d.c. function on a simple region.

Since the way a d.c. function is expressed as a difference of two convex functions is never unique, there is room for choosing a d.c. function which also quickly yields sharp bounds, allowing one to solve problems by a branch-and-bound in reasonable time. A particularly promising concept is the concept of d.c.m. decomposition, which assumes the function is written as a sum of d.c. monotonic functions of norms. Monotonicity can be exploited to yield sharper bounds.

Applications of these ideas to particular instances will be presented.

## References

[1] Blanquero, R., and E. Carrizosa. (2009). “Continuous Location Problems and Big Triangle Small Triangle: Constructing Better Bounds," Journal of Global Optimization 45 389-402.

[2] Blanquero, R., E. Carrizosa and P. Hansen. (2009). “Locating Objects in the Plane Using Global Optimization Techniques," Mathematics of Operations Research 34 837-858.

[3] Carrizosa, E., and F. Plastria. (2008). “Optimal Expected-Distance Separating Halfspace," Mathematics of Operations Research 33 662-677.

[4] Plastria, F., and E. Carrizosa. “Minmax-distance approximation and separation problems: geometrical properties," Mathematical Programming.

# Celebration of the Octagon

Pierre Hansen

*GERAD and HEC Montreal,*
*3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada, H3T 2A7,*
pierre.hansen@gerad.ca

*LIX, École Polytechnique, F-91128 Palaiseau, France*

The octagon already appears in the oldest mathematical manuscripts of the Egyptians in relation to the squaring of the circle. Since then, it appears in numerous inspiring and delightful guises in various disciplines. We follow the octagon through almost forty centuries of transformations in mathematics and less transcendental fields such as architecture, engineering, philosophy, poetry and numismatics.

# EXTENDED ABSTRACTS

# Strong formulations for the pooling problem*

Mohammed Alfaki and Dag Haugland

*Department of Informatics, University of Bergen, P.O. Box 7803, N-5020 Bergen, Norway,* {mohammeda,dag}@ii.uib.no

**Abstract** The pooling problem is a well-studied global optimization problem with applications in oil refining and petrochemical industry. Despite the NP-hardness of the problem, which is proved formally in this paper, most instances from the literature have recently been solved efficiently by use of strong formulations. The main contribution from this paper is a new formulation that proves to be stronger than the most successful formulations known to date. We also provide computational experiments confirming the strength of the new formulation.

**Keywords:** global optimization, pooling problem, convex and concave envelopes, linear relaxation

## 1. Introduction

The pooling problem can be considered as an extension of the minimum cost flow problem on networks with three layers of nodes. Raw materials of unequal quality are supplied at the sources, and are first mixed in intermediate nodes (pools). At the terminals, sellable end products are formed by blending the output from the pools. The resulting qualities of the end products thus depend on what sources they originate from, and in what proportions. Restrictions, which may vary between the terminals, apply to these qualities. This problem is typically modeled as a bilinear, nonconvex optimization problem.

There are two main categories of formulations for the pooling problem: The P-formulation [1] consists of flow and quality variables, whereas the Q-formulation [2] uses flow proportions instead of quality variables. This formulation has later been shown to perform better when fed into generic branch-and-cut algorithms. Applying the reformulation linearization technique (RLT) [3] to the Q-formulation gives a stronger formulation called the PQ-formulation [4].

In this paper, we prove formally that the pooling problem is NP-hard. Hence, no compact linear program formulation exists unless $P = NP$. Moreover, we extend the idea in [2] and [4], and give an even stronger formulation based on proportion variables.

### 1.1 The pooling problem

Consider a directed acyclic graph $G = (N, A)$, where the node set $N$ consists of the sources $S$, the pools $Q$, and the terminals $T$, and where the arc set is $A \subseteq (S \times Q) \cup (S \times T) \cup (Q \times T)$.

We define a finite set of *quality attributes* $K$. With each $i \in S \cup T$, we associate a real constant $q_i^k$ for each $k \in K$. If $i \in S$, $q_i^k$ is referred to as the *quality parameter* of attribute $k$ at source $i$, and if $i \in T$, $q_i^k$ is referred to as the *quality bound* of attribute $k$ at terminal $i$. For each $i \in N$, we define the constant *flow capacity* $u_i$, and for each arc $(i, j) \in A$, we define the constant unit cost $c_{ij}$. Typically, $c_{ij} > 0$ if $(i, j) \in A \cap (S \times Q)$ and $c_{ij} < 0$ if $(i, j) \in A \cap (Q \times T)$.

*This research was sponsored by the Norwegian Research Council, Gassco, and Statoil under contract 175967/S30

Define the *flow polytope* $\mathcal{F}(G, S, T, u)$ as the set of $v \in \mathbb{R}_+^A$ satisfying $\sum_{j:(s,j)\in A} v_{sj} \leq u_s$ $(s \in S)$, $\sum_{j:(j,i)\in A} v_{ji} \leq u_i$ $(i \in N \setminus S)$, $\sum_{j:(i,j)\in A} v_{ij} - \sum_{j:(j,i)\in A} v_{ji} = 0$ $(i \in Q)$. For any $v \in \mathcal{F}(G, S, T, u)$, we associate a unique *quality matrix* $w \in \mathbb{R}^{N\times K}$. The elements of $w$ are for all $k \in K$ defined as $w_i^k = q_i^k$ if $i \in S$ and $w_i^k = \sum_{j:(j,i)\in A} w_j^k v_{ji} / \sum_{j:(j,i)\in A} v_{ji}$ if $i \in N \setminus S$. We then have the formal definition:

**Problem 1** (The pooling problem). *Find $v \in \mathcal{F}(G, S, T, u)$ with associated quality matrix $w \in \mathbb{R}^{N\times K}$ satisfying $w_t^k \leq q_t^k$ $\forall t \in T, k \in K$ such that $c^T v$ is minimized.*

**Proposition 1.** *The pooling problem is NP-hard*

*Proof.* The full version of this paper gives a detailed proof based on a polynomial reduction from the maximum three-dimensional matching problem. □

## 2. The PQ-formulation

Define $y_i^s = v_{si} / \sum_{t:(i,t)\in A} v_{it}$ as the proportion of the flow through pool $i \in Q$ that comes from the source $s \in S$ (let $y_i^s = 0$ if $(s, i) \notin A$). Also, define $x_{sit}$ as the flow along the path $(s, i, t)$ in $G$ if $(s, i), (i, t) \in A$, and let $x_{sit} = 0$ otherwise. We also let $v_{ij} = 0$ if $i, j \in N$ and $(i, j) \notin A$. Combining these variables with the flow variables introduced in the previous section, we arrive at the PQ-formulation written as:

$$
\begin{aligned}
&[\mathrm{PQ}] & \min \sum_{(i,j)\in A} c_{ij} v_{ij} & & (1)\\
& & \sum_{j\in Q\cup T} v_{sj} \leq u_s, \quad \sum_{j\in S\cup Q} v_{ji} \leq u_i & \quad s \in S, i \in N \setminus S & (2)\\
& & \sum_{i\in Q}\sum_{s\in S}\left(q_s^k - q_t^k\right) x_{sit} + \sum_{s\in S}\left(q_s^k - q_t^k\right) v_{st} \leq 0 & \quad t \in T, k \in K & (3)\\
& & \sum_{s\in S} y_i^s = 1 & \quad i \in Q & (4)\\
& & \sum_{t\in T} x_{sit} = v_{si} & \quad s \in S, i \in Q & (5)\\
& & x_{sit} - y_i^s v_{it} = 0 & \quad s \in S, i \in Q, t \in T & (6)\\
& & \sum_{s\in S} x_{sit} = v_{it} & \quad i \in Q, t \in T & (7)\\
& & \sum_{t\in T} x_{sit} \leq u_i y_i^s & \quad s \in S, i \in Q & (8)\\
& & 0 \leq v_{ij} \leq \min\{u_i, u_j\} & \quad (i, j) \in A & (9)\\
& & 0 \leq y_i^s \leq 1 & \quad i \in Q, s \in S & (10)
\end{aligned}
$$

Without altering the model, we have rewritten the formulation from [4] using our notation. Note that the variable $v_{si}$ and the constraint (5) are not present in the PQ-formulation written in [4]. However, this is only a notational difference, and will not alter the strength of the formulation. Constraints (7) and (8) are redundant, but as shown in [4], these cuts strengthen the formulation significantly.

A linear relaxation of (1)-(10) is constructed by bounding $x_{sit}$ between the convex and concave envelopes of $y_i^s v_{it}$ for all $(s, i, t)$ defining a path in $G$. For all $(i, j) \in A$, let $\underline{v}_{ij}$ and $\overline{v}_{ij}$ be some lower an upper bound on the variable $v_{ij}$. A corresponding notation will be applied for all other variables. It can be shown [5] that the convex and concave envelopes of $yv$ (for convenience, we drop sub- and superscripts here) on the rectangle $C = [\underline{y}, \overline{y}] \times [\underline{v}, \overline{v}]$, are $\mathrm{vex}_C(yv) =$

$\max\left\{\underline{y}v + \underline{v}y - \underline{v}\underline{y}, \overline{y}v + \overline{v}y - \overline{v}\overline{y}\right\}$ and $\mathrm{cav}_C(yv) = \min\left\{\underline{y}v + \overline{v}y - \overline{v}\underline{y}, \overline{y}v + \underline{v}y - \underline{v}\overline{y}\right\}$, respectively. For any rectangle $C = [\underline{y}, \overline{y}] \times [\underline{v}, \overline{v}] \subset \mathbb{R}^2$, define the polyhedron
$\mathcal{H}[C] = \left\{(y, v, x) \in \mathbb{R}^3 : \mathrm{vex}_C(yv) \leq x \leq \mathrm{cav}_C(yv), (y, v) \in C\right\}$.

Note that it is straightforward to find a rectangle $C_{it}^s$ enclosing all feasible $(y_i^s, v_{it})$. The linear relaxation of the PQ-formulation is thus obtained by replacing the bilinear constraint (6) by the linear inequalities that impose $(y_i^s, v_{it}, x_{sit}) \in \mathcal{H}\left[C_{it}^s\right]$ for all $i \in Q, s \in S, t \in T$.

## 3. Strong formulations with terminal proportions

Analogous to the proportion variables $y_i^s$ ($s \in S$), define for all pools $i \in Q$, $y_i^t$ as the proportion of the flow at $i$ destined for terminal $t \in T$. That is, we let $y_i^t = v_{it} / \sum_{s:(s,i) \in A} v_{si}$. A new model resembling the PQ-formulation is established by replacing constraints (4), (6), (8) and (10) by

$$\sum_{t \in T} y_i^t = 1 \qquad i \in Q, \tag{11}$$

$$x_{sit} - y_i^t v_{si} = 0 \quad s \in S,\ i \in Q,\ t \in T, \tag{12}$$

$$\sum_{s \in S} x_{sit} \leq u_i y_i^t \quad i \in Q,\ t \in T, \tag{13}$$

$$0 \leq y_i^t \leq 1 \qquad i \in Q,\ t \in T, \tag{14}$$

respectively. In other words, this formulation uses terminal instead of source proportions, and is henceforth referred to as the TP-formulation. A comparison to the PQ-formulation shows that the formulations do not in general have equal strength, but none dominates the other (see Table 1 for a comparison on instances from the literature).

The full benefit of the new proportion variables is achieved when they are combined with source proportions in the same model. It follows from the definition of $y_i^s$ and $y_i^t$ that $y_i^s v_{it}$ and $y_i^t v_{si}$ both can be interpreted as the flow along the path $(s, i, t)$. Given this observation, a formulation based on source and terminal proportions (denoted the STP-formulation) can be derived by adding the constraints (11)-(14) to the PQ-formulation.

Clearly, the STP-formulation is at least as strong as the PQ- and TP-formulations. Table 1 shows that there are standard instances from the literature in which the STP-formulation is stronger. It follows that relying solely on either source (see instances Bental4 and Rt2) or terminal (see instances Adhya1, Adhya4 and Haverly3) proportions, may give weaker relaxations than including both.

*Table 1.* Strength of all three relaxations. The best relaxations are given in bold unless all are equal.

| problem | Objective function value | | | Global Solution |
|---|---|---|---|---|
| | PQ | TP | STP | |
| Adhya1 | **−840.27** | −856.25 | **−840.27** | −549.80 |
| Adhya2 | −574.78 | −574.78 | −574.78 | −549.80 |
| Adhya3 | −574.78 | −574.78 | −574.78 | −561.05 |
| Adhya4 | **−961.93** | −967.43 | **−961.93** | −877.65 |
| Bental4 | −550.00 | **−541.67** | **−541.67** | −450.00 |
| Bental5 | −3500.00 | −3500.00 | −3500.00 | −3500.27 |
| Foulds3 | −8.00 | −8.00 | −8.00 | −8.00 |
| Foulds4 | −8.00 | −8.00 | −8.00 | −8.00 |
| Foulds5 | −8.00 | −8.00 | −8.00 | −8.00 |
| Haverly1 | −500.00 | −500.00 | −500.00 | −400.00 |
| Haverly2 | −1000.00 | −1000.00 | −1000.00 | −600.00 |
| Haverly3 | **−800.00** | −875.00 | **−800.00** | −750.00 |
| Rt2 | −6034.87 | **−5528.00** | **−5528.25** | −4391.83 |

## 4. Computational results

To investigate the value of the improved strength, we submitted the STP-formulation and its two competing formulations to the global optimization solver BARON v.1.8.5 [4]. Standard test problems reported in [4] are used for the computational experiments.

Table 2 shows the results using a Dell OPTIPLEX 755 computer with 2GB memory and a 2.4Gz processor. Following [4], we apply the `RELAXATION_ONLY_EQUATIONS` option to the redundant constraints ((7)-(8) for PQ, (5) and (13) for TP, (7)-(8) and (11)-(13) for STP). This forces BARON to neglect these constraints in the local search procedure used in order to produce feasible solutions.

For each formulation, the table contains the total number of nodes in the search tree ($N_{tot}$), the node where the optimal solution was found ($N_{sol}$), and the total CPU-time in seconds ($T_{cpu}(sec)$). We observe that the STP-formulation performs significantly better than the PQ- and TP-formulations, as it solves 12 out of 14 problems in the root node.

*Table 2.* Computational results from all three formulations solved by BARON.

| Problem | PQ-formulation | | | TP-formulation | | | STP-formulation | | |
|---|---|---|---|---|---|---|---|---|---|
| | $N_{tot}$ | $N_{sol}$ | $T_{cpu}(sec)$ | $N_{tot}$ | $N_{sol}$ | $T_{cpu}(sec)$ | $N_{tot}$ | $N_{sol}$ | $T_{cpu}(sec)$ |
| Adhya1 | 21 | 20 | 0.20 | 39 | 27 | 0.59 | 7 | 7 | 0.33 |
| Adhya2 | 33 | 12 | 0.18 | 15 | 12 | 0.29 | 1 | 1 | 0.27 |
| Adhya3 | 31 | 31 | 0.38 | 37 | 16 | 0.60 | 19 | 1 | 0.82 |
| Adhya4 | 1 | −1 | 0.17 | 17 | 17 | 0.24 | 1 | 1 | 0.29 |
| Bental4 | 1 | −1 | 0.01 | 1 | −1 | 0.02 | 1 | −1 | 0.02 |
| Bental5 | −1 | −1 | 0.03 | −1 | −1 | 0.10 | −1 | −1 | 0.17 |
| Foulds3* | −1 | −1 | 0.85 | −1 | −1 | 0.42 | −1 | −1 | 7.46 |
| Foulds4* | −1 | −1 | 1.15 | −1 | −1 | 0.55 | −1 | −1 | 1.71 |
| Foulds5* | −1 | −1 | 0.52 | −1 | −1 | 0.83 | −1 | −1 | 1.79 |
| Haverly1 | 1 | −1 | 0.01 | 1 | −1 | 0.01 | 1 | −1 | 0.02 |
| Haverly2 | 5 | 5 | 0.02 | 1 | 1 | 0.02 | 1 | 1 | 0.03 |
| Haverly3 | 1 | −1 | 0.02 | 1 | −1 | 0.01 | 1 | −1 | 0.01 |
| Rt2 | 13 | 6 | 0.14 | 1 | 1 | 0.10 | 1 | 1 | 0.18 |

* means that we used MINOS as LP-solver instead of CPLEX in BARON in these problems.

-1 means that the problem was solved during the preprocessing.

## 5. Conclusion

We have shown that the pooling problem is NP-hard, and that strong LP-relaxations are obtained by combining variables representing source proportions and terminal proportions in the same model.

## References

[1] Haverly, C.A. (1978). *Studies of the behaviour of recursion for the pooling problem*, ACM SIGMAP Bulletin, **25**:19–28.

[2] Ben-Tal, A., Eiger, G. and Gershovitz, V. (1994). *Global minimization by reducing the duality gap*, Mathematical programming, **63(2)**:193–212.

[3] Adams, W.P. and Sherali, H.D. (1999). A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems, Kluwer Academic Publishers, Vol. 31, Dordrecht.

[4] Sahinidis, N.V. and Tawarmalani, M. (2005). *Accelerating branch-and-bound through a modeling language construct for relaxation-specific constraints*, Journal of Global Optimization, **32(2)**:259–280.

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

# On the facility location problem with limited distances and side constraints

Daniel Aloise[1], Isaac F. Fernandes[1], Pierre Hansen[2,3], Leo Liberti[3] and Dario J. Aloise[1]

[1] *Department of Production Engineering, Universidade Federal do Rio Grande do Norte,*
*Campus Universitário s/n, Natal-RN, Brazil, 59072-970,*
daniel.aloise@gerad.ca, isaac@syncsistemas.com, dario@pep.ufrn.br

[2] *GERAD and HEC Montreal,*
*3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada, H3T 2A7,*
pierre.hansen@gerad.ca

[3] *LIX, École Polytechnique, F-91128 Palaiseau, France,*
liberti@lix.polytechnique.fr

**Abstract** The objective in the facility location problem with limited distances is to minimize the sum of distance functions from the facility to the customers, but with a limit on each of the distances, after which the corresponding function becomes constant. The problem has applications in situations where the service provided by the facility is insensitive after a given threshold distance (e.g. fire station location). In this work, we propose a global optimization algorithm for the case in which there are lower and upper limits on the numbers of customers that can be served.

**Keywords:** facility location, mixed integer nonlinear programming, global optimization

## 1. Introduction

The minisum single facility problem is one of the most fondamental problems in location theory. The objective is to locate a single facility on the plane so that the sum of distances from the facility to a set of demand points is minimized. The problem is often referred to in the literature as the 1-center problem [6].

Drezner, Mehrez and Wesolowsky investigated in [3] the 1-center problem for the case in which the distance functions are constant after given threshold values, which they call *the facility location problem with limited distances*. This problem has applications in situations where the service provided by the facility is insensitive after a given threshold distance. For instance, consider the problem of locating a firestation. In this context, each property has a distance limit after which the service provided by the firemen is useless, and the property is completely destroyed.

In this work, we study the situation on which there are lower and upper bounds in the number of demand points that must be served within the distance limits. Lower bounds in the number of served points may be used to justify the installation of a facility, while upper bounds may express the capacity limitations of the service provided.

## 2. Problem definition

Let us denote $\|p_1 - p_2\|_p$ as the $l_q$-distance between points $p_1$ and $p_2$ in the plane. Given $n$ service points in the plane $p_1, p_2, \ldots, p_n$ with threshold distances $\lambda_i$ for $i = 1, \ldots, n$, the

Limited Distance Minisum Problem with Side Constraints (LDMPSC) can be expressed by:

$$\begin{aligned}
&\min_{y\in\mathbb{R}^2, v\in\{0,1\}^n} \sum_{i}^{n} [\lambda_i(1-v_i) + \|p_i - y\|_q v_i] \\
&\text{subject to} \\
&\|p_i - y\|_q v_i \le \lambda_i \qquad \text{for } i = 1,\ldots,n \\
&L \le \sum_{i=1}^{n} v_i \le U
\end{aligned} \tag{1}$$

where $L$ and $U$ define side constraints in the number of variables $v_i$ which can be equal to 1. The objective function determines if the effective distance from the facility $y \in \mathbb{R}^2$ to service point $p_i$ is equal to $\|p_i - y\|_q$ or $\lambda_i$ depending on the value attributed to variable $v_i$. The first set of constraints assures that $v_i$ can be equal to 1 only if the distance between $p_i$ and the facility $y$ is smaller than the distance limit $\lambda_i$. This avoids the attribution $v_i = 1$ only to satisfy $\sum_{i=1}^{n} v_i \ge L$. This problem cannot be approched directly by MINLP solvers since the feasible set is nonconvex.

## 3. Optimization algorithm

From the formulation above, we have that for a given location $y$, $v_i$ is equal to 1 only if $\|p_i - y\|_q \le \lambda_i$, and to 0 otherwise. If $q = 2$, this is geometrically equivalent in the plane to the condition that $v_i = 1$ if $y$ belongs to a disc with radius $\lambda_i$ centered at $p_i$, and 0 otherwise. Analogously, if $q = 1$, this is equivalent to the condition that $v_i = 1$ if $y$ belongs to a $45^{\circ}$ rotated square with diagonal $2\lambda_i$ centered at $p_i$, and 0 otherwise.

A branch-and-bound algorithm based on the vector $v$ would consider implicitly all $2^n$ subproblems generated by branching on binary variables $v_i$ for $i = 1,\ldots,n$, while adding constraints $\|p_i - y\|_q \le \lambda_i$ and $\|p_i - y\|_q \ge \lambda_i$ to the resulting subproblems. However, the resulting subproblems are difficult to solve. Another possibility is to focus on components $v_i$ of $v$ which might be equal to 1 at the same time. When $q = 2$ for instance, these components are directly associated to convex regions generated by intersections of discs (see Figure 1). For instance, for the region indicated by the bullet in Figure 1, only the components $v_1$, $v_2$ and $v_3$ can be equal to 1.

Hence, we can solve (1) by solving subproblems of the following type:

$$\begin{aligned}
&\min_{y\in\mathbb{R}^2, v\in\{0,1\}^{|S|}} \quad \sum_{i\in S}(\|p_i - y\|_q - \lambda_i)v_i \\
&\text{subject to} \\
&\|p_i - y\|_q \le \lambda_i \;\; \forall i \in S, \\
&L \le \sum_{i\in S} v_i \le U
\end{aligned} \tag{2}$$

where $S \subseteq \{1,2,\ldots,n\}$ is a non-empty set. Each one of the subproblems of type (2) is associated to a distinct region in the plane. For instance, we have a subproblem with $S = \{1,2,3\}$ for the region indicated by the bullet in Figure 1. The number of these regions was proved to be polynomially bounded in [1, 3].

In order to better tackle subproblems (2), they are reformulated exactly (in the sense of [5]) by introducing parameters:

$$M_i \ge \lambda_i \quad \forall i \in S,$$

decision variables:

$$\omega_i \in [-M_i, 0] \quad \forall i \in S,$$

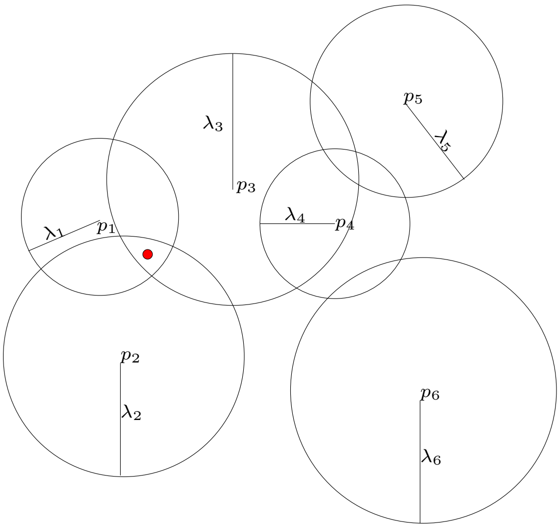


*Figure 1.* Intersection of discs.

and constraints:

$$\omega_i + (1 - v_i)M_i \geq \|p_i - y\|_q - \lambda_i \quad \forall i \in S$$

$$\omega_i + v_i M_i \geq 0 \quad \forall i \in S$$

to (2). We then replace the terms $\|p_i - y\|_p - \lambda_i$ for $i \in S$ in the objective function by $\omega_i$. We thus obtain the reformulated problem:

$$\begin{array}{lr}
\min \sum_{i \in S} \omega_i & \\
\text{subject to} & \\
\omega_i + (1 - v_i)M_i \geq \|p_i - y\|_q - \lambda_i & \forall i \in S \\
\omega_i + v_i M_i \geq 0 & \forall i \in S \\
\|p_i - y\|_q \leq \lambda_i & \forall i \in S \\
L \leq \sum_{i \in S} v_i \leq U & \\
v_i \in \{0, 1\} & \forall i \in S \\
\omega_i \in [-M_i, 0] & \forall i \in S \\
y \in \mathbb{R}^2 &
\end{array} \tag{3}$$

For $q = 1$, (3) is a MIP program, while for $q = 2$ it is still not differentiable due to the $l_2$-distance. Particularly, if squared Euclidean distances are used (i.e., $\| \cdot \|_2^2$), then (3) is a convex MINLP for which there exist practically efficient algorithms (e.g. [2, 4]).

Algorithm 1 below enumerates the sets $S$ corresponding to regions delimited by convex figures (i.e., rotated squares when $q = 1$, discs when $q = 2$). This algorithm executes in $O(n^2\tau)$ time where $\tau$ is the time required for solving each subproblem in steps 4 and 7. We have MIP subproblems for $q = 1$ and 1-center subproblems with side constraints for $q = 2$.

Algorithm 1

1. Enumerate all intersection points of convex figures in the plane as well as all convex figures whose boundary does not intersect any other one. Let $L_1$ and $L_2$ be the corresponding lists.
2. For each intersection point $p \in L_1$ defined by convex figures centered at points $p_i$ and $p_j$, find the set $S$ of all $k$ such that $k \neq i, j$ and $\|p_k - p\|_q \leq \lambda_k$.
3. Consider the four sets: $S, S \cup \{i\}, S \cup \{j\}$, and $S \cup \{i, j\}$.
4. For each one of these sets, solve the associated subproblem of type (3) if the size of that set is larger than $L$.
5. Update the best solution if an improving one is found.
6. For each convex figure in $L_2$ find the set $S'$ composed of its own index and the indices of all convex figures containing it.
7. Solve subproblems of type (3) defined by each $S'$.
8. Update the best solution if an improving one is found.

## 4. Conclusions

The introduction of side constraints while locating a facility in the plane with limited distances may serve to justify its installation or to describe service limitations. Our work extends that of Drezner, Mehrez and Wesolowsky [3], adapting it to the presence of side constraints. This approach leads to subproblems having products of the continuous location variable with assignment binary variables. The subproblem model is then reformulated in order to ease its resolution. To the best of our knowledge, this paper presents the first exact algorithm for locating a facility in the plane with limited distances and side constraints. The presented algorithm takes more time to execute as the complexity of the subproblems increases and as more intersections of convex figures exist. Finally, it is important to remark that the algorithms and formulations presented in this work are converted without loss of generality to the case of weighted distances.

## References

[1] D. Aloise, P. Hansen, and L. Liberti. (2010). "An improved column generation algorithm for minimum sum-of-squares clustering". To appear in *Mathematical Programming*, 2010.

[2] P. Bonami and J. Lee. BONMIN user's manual. Technical report, IBM Corporation, June 2007.

[3] Z. Drezner, A. Mehrez, and G. O. Wesolowsky. (1991). "The facility location problem with limited distances". *Transportation Science*, v. 25, pp. 183–187.

[4] S. Leyffer. "User manual for MINLP_BB". Technical report, University of Dundee, UK, March 1999

[5] L. Liberti. (2009). "Reformulations in Mathematical Programming: Definitions and systematics". *RAIRO-RO*, v. 43, pp. 55–86.

[6] N. Megiddo. (1983). "The weighted Euclidean 1-center problem". *Mathematics of Operations Research*, v.8, pp. 498–504.

# On interval Branch-and-Bound algorithm for additively separable functions with one common variable[*]

José L. Berenguel[1], L.G. Casado[1], E.M.T. Hendrix[2], Frédéric Messine[3] and I. García[2]

[1]*Department of Computer Architecture and Electronics, University of Almería, 04120, Spain.* jlberenguel@gmail.com, leo@ual.es

[2]*Department of Computer Architecture, University of Málaga, 29017, Spain.* Eligius.Hendrix@wur.nl, igarcia@uma.es

[3]*ENSEEIHT-IRIT, UMR-CNRS 5505, 2 rue Camichel BP 7122, 31071 Toulouse Cedex 7, France* Frederic.Messine@n7.fr

**Abstract** Interval Branch-and-Bound algorithms are powerful methods which aim for guaranteed solutions of Global Optimization problems. The computational effort to reach this aim increases exponentially with the problem dimension in the worst case. For separable functions this effort can be less as lower dimensional subproblems can be solved individually. We investigate possibilities to design specific methods for cases where the objective function can be considered separable, but common variables occur in the subproblems. As initial research we tackle the case where the problem can be decomposed in two additively separable functions with just one variable in common.

**Keywords:** Branch-and-Bound, Interval Arithmetic, Separable functions.

## 1. Introduction

Interval Branch-and-Bound methods are powerful methods which aim for guaranteed solutions of Global Optimization problems. Although these methods have the ability to handle constraints, we focus here on the generic box constrained global optimization problem, which is to find

$$f^* = \min_{x \in S} f(x) \tag{1}$$

where $S \in \mathbb{R}^n$. With increasing dimension $n$ the computational effort of B&B interval methods increases drastically. In design problems, it is not unusual that the objective function $f$ is composed of several functions. If this is the case in an additive way and the variables can be split into subgroups that do not overlap, we call the function completely additively separable (CASF). CASF can be solved by finding solutions of subfunctions independently and adding the obtained results. On the other hand, if subfunctions share variables we call the function additively shared separable (ASF). Here we study ASF with two subfunctions that share one variable. In general, solving (1) for ASF seems to be easier when separable character is taken into account. Here we study the adaption of an interval B&B algorithm to ASF. Section 2 describes the standard B&B algorithm we compare with. Section 3 describes the modifications in previous B&B algorithm to solve ASF problems. Results of experiments are presented in Section 4 and conclusion are shown in Section 5.

[*] This work has been funded by grants from the Spanish Ministry of Science and Innovation (TIN2008-01117), Junta de Andalucía (P08-TIC-3518), in part financed by the European Regional Development Fund (ERDF). Eligius Hendrix is a fellow of the Spanish "Ramon y Cajal" contract program, co-financed by the European Social Fund.

## 2. Interval B&B Algorithm

B&B algorithms can be determined by several rules, as it is shown in the following algorithm:

```
B&B ($f$,$S$):
Set the working list $L = \{S\}$ and the final list $Q = \emptyset$
While ( $L \neq \emptyset$ )
  Select an interval $X$ from $L$                              Selection rule
  if $X$ cannot be eliminated                                  Elimination rule
    Divide $X$ into $X^k$,   $k = 1, \ldots, d$, subintervals  Division rule
    foreach $X^k$
      Compute a lower bound of $f(X^k)$                        Bounding rule
      if $X^k$ satisfies the termination criterion             Termination rule
        Store $X^k$ in $Q$
      else
        Store $X^k$ in $L$
```

To solve (1) we use the following rules:

**Bounding.** An interval extension $F$ of $f$ is an inclusion function, i.e, $f(X) \subseteq F(X)$.
**Selection.** Interval $X \in L$ with smallest lower bound ($\underline{F}(X)$) is selected.
**Division.** The widest component of $X$ is bisected. Two subintervals are generated.
**Termination.** If the length of the widest component of $X$ is smaller or equal than $\epsilon$, i.e. $w(X) \leq \epsilon$, interval $X$ is stored in the final list $Q$.
**Elimination.** Two elimination tests are used:

- RangeUp test. Given $\overline{f^*}$ an upper bound of the global minimum $f^*$, an interval $X$ does not contain a minimizer point if the lower bound of $f(X)$, $\underline{F}(X) > \overline{f^*}$. $\overline{f^*}$ is updated with the smallest value of $f$ evaluated at the middle point of the selected intervals ($m(X)$).
- Monotonicity test. If $0 \notin F'_i(X)$ and $X$ does not intersect the boundary of $S$, $X$ can be rejected.

More elaborated rules can be found in literature but we use this simple algorithm as a basis to compare with [1–3].

## 3. Interval B&B Algorithm for ASF (ASF-B&B)

The instances of ASF problem that we want to solve here can be formulated from (1) as:

$$f^* = \min_{x \in S} f(x) = \min_{x^{[1]} \in S^{[1]}, x^{[2]} \in S^{[2]}} f^{[1]}(x^{[1]}) + f^{[2]}(x^{[2]}), \tag{2}$$

with $S^{[1]} \cup S^{[2]} = S$, $x^{[1]} \in \mathbb{R}^{n^{[1]}}$, $x^{[2]} \in \mathbb{R}^{n^{[2]}}$, $n^{[1]} + n^{[2]} = n + 1$ and without loss of generality we take as common variable $x^{[1]}_{n^{[1]}} = x^{[2]}_{n^{[2]}}$. To solve (2) using a B&B algorithm, we define two working lists $L^{[1]}$, $L^{[2]}$ and two final lists $Q^{[1]}$, $Q^{[2]}$ for the corresponding subfunctions. Following, the rules that define ASF-B&B algorithm are described:

**Bounding.** Given an interval $X^{[i]}$, we store two lower bounds, one for $f^{[i]}$ and another for $f$. They are calculated as follows:

- $\underline{F}^{[i]}(X^{[i]})$ is a lower bound of $f^{[i]}(X^{[i]})$ due to interval arithmetic.
- We define $EX^{[i]} = \{E \subseteq S, E^{[i]} = X^{[i]}\}$. Without loss of generality, let us focus on $X^{[1]}$. Then,

$$\underline{F}(EX^{[1]}) = \underline{F}^{[1]}(X^{[1]}) + \underline{F}^{[2]}(Z^{[2]}), \text{ where} \tag{3}$$

$$Z^{[2]} = \arg \min_{X^{[2]} \in L^{[2]} \cup Q^{[2]}} \underline{F}^{[2]}(X^{[2]}), \text{ with } X^{[1]}_{n^{[1]}} \cap X^{[2]}_{n^{[2]}} \neq \emptyset. \tag{4}$$

The proof of $\underline{F}(EX^{[1]})$ as a correct lower bound of $f(EX^{[1]})$ will be given in the final article.

**Selection.** List $L^{[1]}$ and $L^{[2]}$ are visited using round robin. Interval $X^{[i]}$ with the smallest lower bound $\underline{F}(EX^{[i]})$ is selected (see eq. (3) ).

**Division.** The widest component of $X^{[i]}$ is bisected. Two subintervals are generated.

**Termination.** If the length of the widest component of $X^{[i]}$ is smaller or equal than $\epsilon$, i.e. $w(X^{[i]}) \leq \epsilon$, interval $X^{[i]}$ is stored in final list $Q^{[i]}$.

**Elimination.** Wlog, let us focus on $X^{[1]}$. Elimination tests are the following:

- Unshared value of common variable. $X^{[1]}$ can be removed if $\forall X^{[2]} \in L^{[2]} \cup Q^{[2]}$, $X^{[1]}_{n^{[1]}} \cap X^{[2]}_{n^{[2]}} = \emptyset$.
- Non common variables monotonicity. $X^{[1]}$ can be removed if $0 \notin F'_i(X^{[1]})$, $i \neq n^{[1]}$ and $X^{[1]}$ does not intersect the boundary of $S^{[1]}$.
- RangeUp. Given $\overline{f^*}$ an upper bound of the global minimum $f^*$, an interval $X^{[1]}$ does not contain a minimizer point if $\underline{F}(EX^{[1]}) > \overline{f^*}$ (see eq. (3) ). $\overline{f^*}$ is updated with the smallest value of $f^{[1]} + f^{[2]}$ evaluated at the middle point of selected intervals $X^{[1]}$, $Z^{[2]}$ (see eq. (4) ), but using as common variable $X^{[1]}_{n^{[1]}} = Z^{[2]}_{n^{[2]}} = X^{[1]}_{n^{[1]}} \cap Z^{[2]}_{n^{[2]}}$.
- Subfunction RangeUp. An interval $X^{[1]}$ does not contain a minimizer point if $\underline{F}^{[1]}(X^{[1]}) > \overline{g}^{[1]}(X^{[1]}_{n^{[1]}})$, where $\overline{g}^{[1]}(X^{[1]}_{n^{[1]}})$ is the lowest value for $f^{[1]}(m(Y^{[1]}))$ found so far with $m(Y^{[1]})_{n^{[1]}} \in X^{[1]}_{n^{[1]}}$.

## 4. Results

We designed several instances to measure the performance of the ASF-B&B algorithm compared with B&B algorithm. The design of these test functions has been done using well-known functions in Global Optimization. The process is the following: we select two functions, for example Levy-5 (2 dimensions) and Price (2 dimensions), and create the new separable function L5P sharing the last variable of both functions. Table 1 shows the list of separable functions with their search domain and the subfunctions used to create them.

*Table 1.* Separable Test Functions.

| Name | $S$ | $f^{[1]}$ | $f^{[2]}$ |
|---|---|---|---|
| Eligius | $[-10, 10]^3$ | $x_1^2 + x_1x_3 + \frac{1}{2}x_3^2 + x_3$ | $x_2^2 - 2x_2x_3$ |
| L5P | $[-10, 10]^3$ | Levy5 | Price |
| GP3 | $[-2, 2]^3$ | Goldstein-Price | Goldstein-Price |
| SHCBL3 | $[-10, 10]^3$ | Six-Hump-Camel-Back | Levy3 |

Table 2 shows the execution results of the functions in Table 1 using B&B and ASF-B&B algorithms. From left to right, the columns of the table are the name of the function; the precision of the final boxes; the interval containing the minimum value; if $FE$ is the number of functions evaluations and $GE$ is the number of gradient evaluations, the effort is measured as: $Effort = FE + n \cdot GE$; the execution time; the number of final boxes for B&B; and the number of final boxes for ASF-B&B.

Values in column $Q$ for ASF-B&B algorithm shows the number of boxes after post-processing boxes in $Q^{[1]}$ and $Q^{[2]}$. This post-processing is based on applying the B&B elimination tests on combining separable final boxes in non-separable ones. The execution time for ASF-B&B is the algorithm running time plus the post-processing time. These results show that monotonicity test in common variable, which is not used in ASF-B&B, has importance when the precision increases.

The current ASF-B&B algorithm running time is in general worse than in B&B due to coding details of the data structures that needs more than one sorting index. The version in the final paper will be improved and then the execution times shoould be reduced. Focusing on Effort, ASF-B&B outperform B&B for low precisions. For higher precisions, ASF-B&B outperform B&B in two, out of four cases, and it is similar in one. The difference in dimension for separable and non separable functions in the experimentation is just one. This shows that it is interesting to investigate the improvements of the ASF-B&B algorithm to solve larger dimensional problems.

*Table 2.* Execution Results.

| Name | $\epsilon$ | Minimum | Effort | Time | $Q$ | $Q^{[1]}$ | $Q^{[2]}$ |
|---|---|---|---|---|---|---|---|
| Eligius-B&B | $10^{-2}$ | $[-85.234328, -84.843821]$ | 2,227 | 0.01 | 4 | - | - |
| Eligius-ASF-B&B | | $[-85.244046, -84.921886]$ | 1,921 | $0.04 + 0$ | 4 | 6 | 8 |
| Eligius-B&B | $10^{-4}$ | $[-85.001832, -84.998779]$ | 3,424 | 0.02 | 4 | - | - |
| Eligius-ASF-B&B | | $[-85.001908, -84.999389]$ | 2,929 | $0.05 + 0.01$ | 4 | 6 | 8 |
| L5P-B&B | $10^{-2}$ | $[-174.6145, -172.2646]$ | 3,742 | 0.07 | 3 | - | - |
| L5P-ASF-B&B | | $[-174.6145, -172.2646]$ | 3,568 | $0.09 + 0$ | 3 | 7 | 22 |
| L5P-B&B | $10^{-4}$ | $[-172.2961, -172.2769]$ | 4,363 | 0.08 | 3 | - | - |
| L5P-ASF-B&B | | $[-172.2961, -172.2769]$ | 14,894 | $0.52 + 0.06$ | 3 | 77 | 239 |
| GP3-B&B | $10^{-1}$ | $[-66, 498.05, 65.118653]$ | 239,683 | 3.21 | 8,893 | - | - |
| GP3-ASF-B&B | | $[-66, 169.43, 65.118653]$ | 26,978 | $2.09 + 1.18$ | 6,103 | 665 | 514 |
| GP3-B&B | $10^{-2}$ | $[-4, 280.754, 65.000372]$ | 2,681,671 | 35.45 | 68,017 | - | - |
| GP3-ASF-B&B | | $[-4, 280.412, 65.000372]$ | 287,367 | $161.44 + 51.92$ | 52,838 | 5,013 | 4,073 |
| SHCBL3-B&B | $10^{-2}$ | $[-171.4726, -168.6490]$ | 40,846 | 0.73 | 48 | - | - |
| SHCBL3-ASF-B&B | | $[-171.4702, -168.6490]$ | 9,723 | $0.41 + 0.02$ | 30 | 62 | 21 |
| SHCBL3-B&B | $10^{-4}$ | $[-168.6792, -168.6566]$ | 46,372 | 0.82 | 24 | - | - |
| SHCBL3-ASF-B&B | | $[-168.6792, -168.6566]$ | 46,767 | $3.78 + 0.5$ | 24 | 635 | 246 |

## 5. Conclusions

A new B&B algorithm to solve additively separable functions has been presented. Numerical results show that the current version of the algorithm outperforms the classic one for low precision results. The causes of poor results for higher precisions are known and deserve additional research. Results of an improved ASF-B&B algorithm will be shown in the final article.

## References

[1] E. R. Hansen and G. W. Walster. *Global optimization using interval analysis*. Marcel Dekker, 2nd edition, 2004.

[2] R. B. Kearfott. *Rigorous Global Search: Continuous Problems*. Kluwer Academic Publishers, Dordrecht, Holland, 1996.

[3] H. Ratschek and J. Rokne. *New Computer Methods for Global Optimization*. Ellis Horwood Ltd., Chichester, England, 1988.

# A mixed-integer optimization model for Air Traffic Deconfliction

Sonia Cafieri,[1] Pascal Brisset,[1] and Nicolas Durand[2]

[1] *École Nationale de l'Aviation Civile, 7 avenue E. Belin 31055 Toulouse, France* sonia.cafieri@enac.fr

[2]*Direction des Services de la Navigation Aérienne, Direction de la Technique et de l'Innovation, Domaine R&D, 7 avenue E. Belin 31055 Toulouse, France* nicolas.durand@aviation-civile.gouv.fr

We dedicate this paper to Pascal Brisset, our friend and colleague who died on May 22th 2010 in a tragic hiking accident. Pascal, we miss you so much!

**Abstract** A mixed-integer nonlinear optimization model is presented for the resolution of aircraft conflict. Aircraft conflicts occur when aircraft sharing the same airspace are "too close" to each other and represent a crucial problem in Air Traffic Management. We describe the model and show some numerical experiments.

**Keywords:** aircraft conflict aivoidance, modeling, global optimization, MINLP

## 1. Introduction

The problem of detecting and solving aircraft conflicts, that occur when the distance between two aircraft sharing the same airspace is less than a given safety distance, is crucial in Air Traffic Management to guarantee air traffic safety. Currently, the resolution of conflicts is still largely performed manually by air traffic controllers watching the movement of traffic on a radar screen. Therefore, a great interest is devoted to the development of automatic tools.

One aims at solving a conflict while deviating as little as possible from the original flight plan. An optimization problem can thus be naturally defined. Notwithstanding the importance of the problem and the urgent need of automatic tools able to integrate human work to face the growing air traffic security requirements, there is still a need for suitable models. Different models have been proposed based on allowing both heading angle deviation and speed change maneuvers, either in a centralized [5][6][7] or in an autonomous [4][3] approach. The advantages of subliminal control using only small speed adjustments were shown in the ERASMUS [2] project. In this paper, we propose a new model for air conflict avoidance based on velocity changes. It is mixed-integer because it requires the use of continuous and discrete variables, in particular 0-1 variables to represent logic choices, and involves nonlinear terms. The model is then in the area of Mixed-Integer Nonlinear Programming. In the following sections we describe the model and we show some computational results obtained using a general-purpose global optimization solver.

## 2. Modelization

Aircraft are assumed to be flying on a horizontal plane and are identified by points in the plane. We propose a model based on instantaneous velocity changes, while the trajectory is kept unchanged. The main idea is to deal with the different time windows where aircraft fly with their original (known) speed $v$ or with a changed speed $v + q$, $q$ representing a possible positive or negative speed change. Time windows are defined by instant times such that each aircraft changes its original velocity, i.e., it starts or ends flying with speed $v + q$. Because of the assumption of instantaneous velocity changes, we can consider uniform motion laws in each time window, where the velocity to be considered for each aircraft $k$ is $v_k$ or $v_k + q_k$ depending on the time configuration. There are 6 possible time configurations, obtained considering permutations of instant times when aircraft change their speed, and, for each time configuration, 5 time intervals have to be taken into account. Given a pair of aircraft $i$ and $j$, let $t_{1i}, t_{1j}$ and $t_{2i}, t_{2j}$ be the instant times when $i$ and $j$ start and respectively end flying with changed speed. An order for $t_{1i}, t_{2i}, t_{1j}, t_{2j}$ is not a priori known. By permutations of these instant times, excluding some cases giving rise to inconsistency (i.e., taking into account that $\forall k \ t_{1k} \leq t_{2k}$ and so a time sequence always starts with a $t_1$ instant and ends with a $t_2$ one), we obtain the following time configurations, where $T$ represents the upper bound on time instants:

$$0 \leq t_{1i} \leq t_{1j} \leq t_{2i} \leq t_{2j} \leq T \tag{1}$$
$$0 \leq t_{1j} \leq t_{1i} \leq t_{2i} \leq t_{2j} \leq T \tag{2}$$
$$0 \leq t_{1i} \leq t_{2i} \leq t_{1j} \leq t_{2j} \leq T \tag{3}$$
$$0 \leq t_{1j} \leq t_{2j} \leq t_{1i} \leq t_{2i} \leq T \tag{4}$$
$$0 \leq t_{1i} \leq t_{1j} \leq t_{2j} \leq t_{2i} \leq T \tag{5}$$
$$0 \leq t_{1j} \leq t_{1i} \leq t_{2j} \leq t_{2i} \leq T. \tag{6}$$

Each of these configurations defines 5 time intervals.

We use as decision variables:

- $\forall k \in A \quad q_k$, where $A$ is the set of aircraft, expressing the speed change of aircraft. Note that $q_k$ can be positive (acceleration), negative (deceleration) and null (if there is no speed change). We impose, as it is done in practice, that the speed change for aircraft $k$ cannot be greater than $+3\%$ and smaller than $-6\%$ of its original speed.
- $\forall k \in A \quad t_{1k}$, $t_{2k}$, representing the instant times such that aircraft $k$ starts and respectively ends flying with changed speed. Instant time are always $\geq 0$ and have an upper bound $T$. They are such that $t_{1k} \leq t_{2k}$.

We also employ auxiliary variables to model the problem, both continuous and integer (and in particular binary). Suitable integer variables are in particular used to describe all possible time configurations.

We aim at obtaining conflict avoidance with the minimum speed change for aircraft that should fly with changed speed during a time interval which also has to be minimized. We then use as objective function:

$$\min \sum_{k \in A} q_k^2 (t_{2k} - t_{1k})^2. \tag{7}$$

We impose a number of constraints that are used to handle time configurations and to express aircraft separation conditions in each time interval.

Firstly, the interval time for speed change must be at least equal a certain amount $t_{min}$:

$$\forall k \in A \quad t_{2k} - t_{1k} \geq t_{min}. \tag{8}$$

Modeling all possible time configurations needs the introduction of binary variables $z_\ell$, $\ell \in \{1, \ldots, 6\}$ stating, for each time configuration, what is the order of instant times for that configuration. So, for example, the binary variable $z_1$ is such that:

$$t_{1i} \leq t_{1j} \quad \text{and} \quad t_{1j} \leq t_{2i} \quad \text{and} \quad t_{2i} \leq t_{2j}. \tag{9}$$

The following constraint imposes that only one configuration must hold:

$$\sum_{\ell \in \{1,\ldots,6\}} z_\ell = 1. \tag{10}$$

Aircraft separation is expressed by the following condition:

$$||\overrightarrow{x}^r(t)|| \geq d, \tag{11}$$

where $d$ is the minimum required separation distance and $\overrightarrow{x}^r(t)$ is given by

$$\overrightarrow{x}^r(t) = \overrightarrow{x}^{rd}_{ij} + \overrightarrow{v}^{r}_{ij} t, \tag{12}$$

where $\overrightarrow{x}^{rd}_{ij}$ is the relative initial position of aircraft $i$ and $j$ and $\overrightarrow{v}^{r}_{ij}$ their relative speed. Squaring (11) and deriving with respect to $t$, one can see that the minimum is attained for $t_m = -\frac{\overrightarrow{v}^{r}_{ij} \overrightarrow{x}^{rd}_{ij}}{(v^r_{ij})^2}$. We are only interested in the minimum in each interval $[t_s, t_{s'}]$ Substituting, the following separation condition is obtained:

$$(x^{rd}_{ij})^2 - \frac{(\overrightarrow{v}^{r}_{ij} \overrightarrow{x}^{rd}_{ij})^2}{(v^r_{ij})^2} - d^2 \geq 0. \tag{13}$$

Initial position in each time interval, relative distances and speeds between aircraft are then exploited, and new variables introduced accordingly. Distances covered by aircraft during each time interval are computed exploiting laws of uniform motion because of the aircraft's constant speed in each of such intervals. In the $h$-th time interval $[t_s, t_{s'}]$, $h \in \{1, \ldots, 5\}$, for all aircraft $k \in A$ the initial position $x_{kh}$ is given by

$$x_{kh} = x_{k(h-1)} + (t_{s'} - t_s)\bar{v}_k, \tag{14}$$

where $\bar{v}_k$ is the original speed $v_k$ or the changed speed $v_k + q_k$, depending on the time configuration holding. So, (continuous) variables $x_{kh}$ $\forall k \in A$ $\forall h \in \{1, \ldots, 5\}$, are introduced and corresponding constraints added to the formulation, expressing for each aircraft the 5 initial positions in the 5 time intervals. Each aircraft $k$ has speed $\bar{v}_k$ equal to its original speed or to the changed speed depending on the time configuration, so that variables $z_\ell$ are used to identify the configuration holding. Relative distances $x^{rd}_{ij}$ and relative speeds $v^r_{ij}$ between aircraft are also defined, for each time configuration and each time interval, and constraints adjoined accordingly using variables $x_{kh}$ and again $z_\ell$.

Further constraints are then adjoined to the model to impose the condition (13) in each of the 5 time intervals, when $t_m \in [t_s, t_{s'}]$, where $[t_s, t_{s'}]$ is the $h$-th time interval. In order to check if $t_m \in [t_s, t_{s'}]$, binary variables are used. For all $h \in \{1, \ldots, 5\}$ a binary variable $y_{lh}$ is introduced such that $y_{lh} = 1$ if $t_{sh} \leq t_{mh}$ and 0 otherwise, $y_{rh}$ is such that $t_{mh} \leq t_{s'h}$ and 0 otherwise. The following constraints are then imposed:

$$\forall h \in \{1, \ldots, 5\} \qquad t_{sh} \leq t_{mh} + M(1 - y_{lh}), \quad t_{mh} \leq t_{sh} + M y_{lh} \tag{15}$$

$$t_{mh} \leq t_{s'h} + M(1 - y_{rh}), \quad t_{s'h} \leq t_{mh} + M y_{rh} \tag{16}$$

with $M$ sufficiently large. Condition (13) is then imposed for each time configuration $\ell \in \{1, \ldots, 6\}$, $\forall h \in \{1, \ldots, 5\}$ and $i, j \in A$, as follows:

$$\left( y_{lh} y_{rh} \left( (x^{rd}_{ijh})^2 - \frac{(v^r_{ijh} x^{rd}_{ijh})^2}{(v^r_{ijh})^2} - d^2 \right) \right) \geq 0 \tag{17}$$

Finally, for each time interval, the following separation condition is also imposed:

$$\forall h \in \{1, \ldots, 5\}, \forall i, j \in A \qquad (x^{rd}_{ijh})^2 \geq d^2. \tag{18}$$

## 3. Computational experience

We carried out preliminary computational experiments considering a pair of aircraft. The two aircraft are supposed to move from an initial position given, in 2-dimensional space, by $(-100, 0)$ and $(0, -100)$ respectively and with a velocity $v = 400Nm/h$. Separation distance $d$ is equal to $5Nm$. Aircraft $k$ is assumed to change its speed on an instant $t_{1k}$ and keep the new speed. We solved the problem using the `Couenne` [1] software for MINLP, obtaining the following optimal solution:

$$q_1 = -0.05636 \times v,\ q_2 = 0.02492 \times v,\ t_{11} = 0.00611072,\ t_{12} = 0.0115235, \qquad (19)$$

corresponding to the objective function value 0.00086678. This solution required 1.99 seconds of CPU time on a 2.4 GHz CPU.

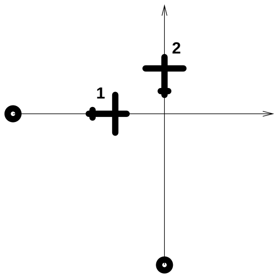


*Figure 1.* Example of conflict resolution, as described in Sect.3. The conflict in (0,0) is solved by decelerating the first aircraft and accelerating the second one in an optimal way.

## 4. Summary

We presented a mixed-integer nonlinear model for the problem of aircraft conflict resolution, a challenging problem in Air Traffic Management. In this model, conflicts are avoided allowing aircraft to only accelerate or decelerate in a time window, and speed changes are minimized together with time windows when they occur. Preliminary computational experiments show that the model is promising in air conflict resolution. We plan to extend the proposed model to the case of $n > 2$ aircraft.

**Acknowledgments.** The authors wish to thank Pietro Belotti for helpful discussions about the use of `Couenne`.

## References

[1] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, 24(4):597–634, 2009.

[2] Deidre Bonini, Carole Dupré, and Géraud Granger. How erasmus can support an increase in capacity in 2020. In *Proceedings of the 7th International Conference on Computing, Communications and Control Technologies: CCCT 2009, Orlando, Florida*, 2009.

[3] Vu. N. Duong and Karim Zeghal. Conflict resolution advisory for autonomous airborne separation in low-density airspacse. In *Proceedings of the 36th IEEE Conf. on Decision and Control, San Diego USA*, 1997.

[4] N. Durand, J.M. Alliot, and F. Medioni. Neural nets trained by genetic algorithms for collision avoidance. *Applied Artificial Intelligence*, 2000.

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

[6] E. Frazzoli, Z.H. Mao, and E. Feron. Aircraft conflict resolution via semidefinite programming. *AIAA Journal of Guidance, Control and Dynamics*, 2001.

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

# On regular simplex subdivision in Branch-and-Bound algorithms for blending problems *

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

[1]*Computer Architecture and Electronics Department, University of Almería, 04120, Spain.* leo@ual.es

[2]*Department of Computer Architecture, University of Málaga, Campus de Teatinos, 29017, Spain.* Eligius.Hendrix@wur.nl, igarciaf@uma.es

[3]*Department of Differential Equations, Budapest University of Technology and Economics, Hungary.* bog@math.bme.hu

**Abstract** One of the aspects of Branch-and-Bound (B&B) algorithms is the use of an effective rejection (also called pruning) tests. Blending problems have the unit simplex as search space. The aim of this article is to study division schemes that generate new B&B sub-problems. The division scheme aims to increase the success of rejection tests and to decrease the number of vertex and simplex evaluations. In this way a division scheme improves the performance of the algorithm. [3] show that a simplex can be rejected if it is covered by infeasibility spheres centered at its vertices. In general, a regular simplex has more chance to be covered than an irregular one due to the equal distance between its vertices. Unfortunately, regular division without overlapping is not known for $d$-simplices, with $d > 2$. This work shows empirically the advantages of a regular partition in blending problems. Therefore, it is important to solve issues associated to overlap in regular division. Some strategies are described.

**Keywords:** Branch-and-Bound, blending, simplex partition, covering.

## 1. Introduction

Consider the following formulation of a mixture design problem which actually consists of identifying mixture products, each represented by a vector $x \in \mathbb{R}^n$, which meet certain requirements [3, 8]. The set of possible mixtures is mathematically defined by the unit simplex

$$S = \left\{ x \in \mathbb{R}^n \mid \sum_{j=1}^{n} x_j = 1.0;\ 0 \leq x_j \leq 1 \right\}, \tag{1}$$

where the variables $x_j$ represent the fraction of the components in a product $x$. In mixture design (blending) problems, the objective is to minimize the cost of the material,

$$f(x) = e^T x, \tag{2}$$

where vector $e$ gives the costs of the raw materials. In the model under study, linear inequality constraints and bounds define the design space $X \subset S$. The requirements are defined as

*This work has been funded by grants from the Spanish Ministry of Science and Innovation (TIN2008-01117), Junta de Andalucía (P08-TIC-3518), in part financed by the European Regional Development Fund (ERDF). Eligius Hendrix is a fellow of the Spanish "Ramon y Cajal" contract program, co-financed by the European Social Fund. B.G.-Tóth was supported by the János Bolyai fellowship of the Hungarian Academy of Sciences.

quadratic inequalities.

$$g_i(x) = x^T A_i x + b_i^T x + c_i \leq 0; \;\; i = 1, \ldots, m, \tag{3}$$

in which $A_i$ is a symmetric $n$ by $n$ matrix, $b_i$ is an $n$-vector and $c_i$ is a scalar. In this way we formulate the problem to be solved as finding elements of the set of "satisfactory" (feasible) products

$$D = \{x \in S \mid g_i(x) \leq 0; \;\; i = 1, \ldots, m\}. \tag{4}$$

Finding a point $x \in X \cap D$ defines the quadratic mixture design problem (QMDP), as studied in [7]. From practical considerations, this problem was extended towards robust solutions. One can define robustness $R(x)$ of a design $x \in D$ with respect to $D$ as

$$R(x) = \max\{R \in \mathbb{R}^+ \mid (x + h) \in D, \; \forall h \in \mathbb{R}^n, \; \|h\| \leq R\} \tag{5}$$

Notice that for mixture problems $x + h$ is projected on the unit simplex. Additionally, variables has semi-continuity property related to a minimum acceptable dose $md$ that the practical problems reveal. Therefore, we are merely interested in methods for finding an $\epsilon$-robust solution with minimum cost, i.e.

$$\begin{array}{lll} \min & f(x) & \text{(Cost} \quad (2)) \\ \text{s.t.} & x \in X \cap D & \text{(Feasibility} \quad (4)) \\ & R(x) \geq \epsilon & \text{(Robustness} \quad (5)) \\ & x_j = 0 \text{ or } x_j \geq md & \text{(Minimal dose)} \end{array} \tag{6}$$

Independently of the application, we are dealing with a B&B algorithm where the search region defined as a simplex is decomposed iteratively [8]. The left hand side graph of Figure 1 shows the initial search space for $n = 2$, which is composed of two 0-simplices (one raw material) and one 1-simplex (two raw materials). The right hand side graph of Figure 1 shows the initial search space for $n = 3$, which consists of three 0-simplices, three 1-simplices and one 2-simplex.

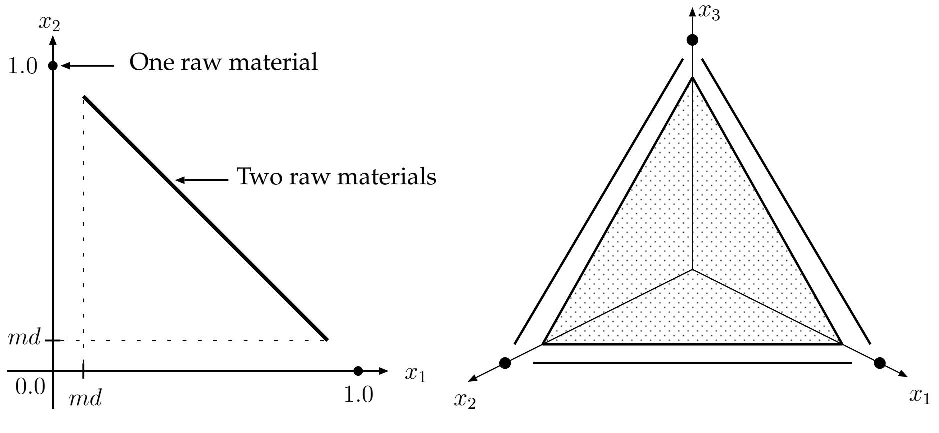


*Figure 1.* n=2 and n=3 initial simplices by removing the minimum dose region

B&B methods can be characterized by four rules: *Branching, Selection, Bounding,* and *Elimination* [10, 11]. For continuous problems, like the mixture design problem, a termination criterion has to be incorporated; i.e, one has to establish a minimum sampling precision $\alpha$. A detailed description of these rules can be found in [8]. Here we focus on *Division rule* because it affects the effectiveness of the elimination tests. The use of simplicial sets in B&B and several ways of splitting them has been studied extensively in [4–6, 9]. Bisection of the longest edge (BLE), as shown in Figure 2a, is most used because it is simple and for all the generated simplices the length of the longest edge is at most twice the size of the shortest edge.

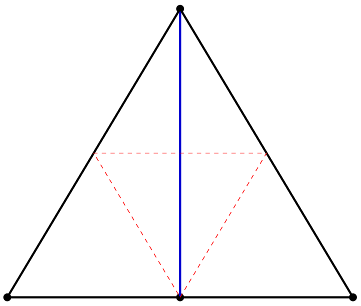


*Figure 2a.* BLE.
Bisect the Longest Edge.

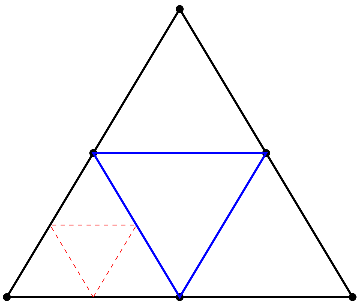


*Figure 3b.* BAE.
Bisect All Edges.

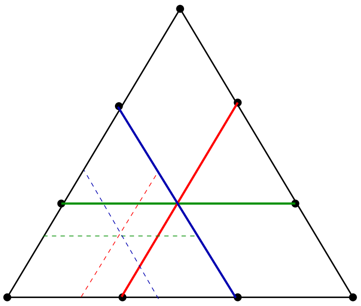


*Figure 4c.* ROD.
Regular Overlapping Division.

In general, regular shaped simplices give better bounding results than nonregular ones as can be found among others in [2]. A regular partition of a simplex similar to bisecting all edges (BAE), as in Figure 3b, has not been found for dimension higher than 3. Therefore we develop and investigate the potential of a simplicial subdivision which is regular, but is not a partition. Figure 4c shows an example of a regular overlapping division (ROD) for $n = 3$. The number of simplices generated is $n$ and the length of its edges is $(n-1)/n$, as shown by [4]. Dashed lines in Figures 2a to 4c are possible future divisions.

## 2. Experimental results for blending problems

Results for B&B algorithm using bisection of the largest edge are shown in [8]. Among several rejection tests, we want to highlight here those based on covering the simplex by infeasibility spheres centered at its vertices. A summary of them is:

- SCTest (Single Cover Test): One sphere covers all the simplex.
- MCTest (Multiple Cover Test): A simplex $S$ can be rejected if a point $p \in S$ is covered by all spheres. The correctness of the test was proved in [3]. The proposed $p$ in [8] is heuristic and can be calculated at low computational cost.
- $\theta$-Test: The point $\theta$ to be covered is determined by a system of equations in [2]. If one sphere covers $\theta$ all spheres cover it, even if $\theta \notin S$. Even if $\theta$ is not covered, there exist cases where the covering of $S$ can be determined from $\theta$ with additional computational cost, but we will not consider them in this experimentation.

Table 1 shows the efficiency of different division schema for problems defined in [8]. The efficiency is measured in terms of number of simplex (NSimplex) and vertex evaluations (NVertex). The rejection tests: SCTest, MCTest and $\theta$-test, with others shown in [8], are checked in order.

BLE, BAE and ROD has been evaluated for 3-dimensional problems Case2 and RumCoke. BAE outperform BLE in efficiency. If the rejection test is not very successful it is better to do multisection, as it is shown for boxes in [1]. Additionally, the SCTest rejects more simplices at earlier stages of the algorithm because they are regular. BAE reduces the need of MCTest and only one expensive $\theta$-Test is needed for Case2 and none for RumCoke. On the other hand, ROD is the worst of all divisions due to the fact that one simplex is overlapped by several simplices. So, unnecessary redundant computation is done. Additionally, it lacks vertex reusability.

Table 1 shows the necessary development of a regular division for larger dimensional problems. The number of congruent classes of simplices generated by BLE is $n!/2$ [6]. This hinders the success of SCTest. We research how to avoid redundant computation in ROD division, increasing the vertex reusability at the same time. Some strategies are designed but they deserve a complete article.

*Table 1.* Experimental results for different division schema. $\alpha = \epsilon = \sqrt(2)/100$, $md = 0.03$.

| *Problem* | $n$ | Division | NSimplex | NVertex | SCTest | NCTest | $\theta$-Test |
|---|---|---|---|---|---|---|---|
| Case2 | 3 | BLE | 393 | 136 | 58 | 18 | 3 |
| | | BAE | 291 | 153 | 66 | 6 | 1 |
| | | ROD | 5,946 | 11,878 | 745 | 37 | 16 |
| RumCoke | 3 | BLE | 569 | 179 | 70 | 31 | 6 |
| | | BAE | 341 | 172 | 86 | 13 | 0 |
| | | ROD | 9,038 | 18,059 | 1,141 | 100 | 23 |
| UniSpec1 | 7 | BLE | 72,419 | 7,561 | 11,146 | 5,442 | 780 |
| UniSpec5b | 7 | BLE | 94,422,861 | 1,962,173 | 15,135,582 | 9,546,656 | 1,108,185 |

## 3. Conclusions

Regular partition seems to outperform bisection and increases the performance of simple rejection tests. Unfortunately, regular division for dimension greater than 3 is only known with overlapping divisions. New methods to avoid redundant computation and to increase the vertex reusability in regular division are investigated. They will be shown in the final paper.

## References

[1] L. G. Casado, I. García, and T. Csendes. A New Multisection Technique in Interval Methods for Global Optimization. *Computing*, 65(3):263–269, 2000. doi:10.1007/s006070070010.

[2] L. G. Casado, I. García, B. G.-Tóth, and E.M.T. Hendrix. On determining the cover of a simplex by spheres centered at its vertices. *Journal of Global Optimization*, 2010. doi:10.1007/s10898-010-9524-x.

[3] L. G. Casado, E.M.T. Hendrix, and I. García. Infeasibility spheres for finding robust solutions of blending problems with quadratic constraints. *Journal of Global Optimization*, 39(2):215–236, 2007. doi:/10.1007/s10898-006-9072-6.

[4] J. Claussen and A. Zilinskas. Subdivision, sampling and initialization strategies for simplical branch and bound in global optimization. *Computers and Mathematics with Applications*, 44:943–955, 2002.

[5] Herbert Edelsbrunner and Daniel R. Grayson. Edgewise subdivision of a simplex. In *SCG '99: Proceedings of the fifteenth annual symposium on Computational geometry*, pages 24–30, New York, NY, USA, 1999. ACM.

[6] Eduardo N. Gonçalves, Reinaldo M. Palhares, Ricardo H. C. Takahashi, and Renato C. Mesquita. Algorithm 860: Simples—an extension of freudenthal's simplex subdivision. *ACM Trans. Math. Softw.*, 32(4):609–621, 2006.

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

[8] E.M.T. Hendrix, L. G. Casado, and I. García. The semi-continuous quadratic mixture design problem: Description and branch-and-bound approach. *European Journal of Operational Research*, 191(3):803–815, 2008. doi:10.1016/j.ejor.2007.01.056.

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

[10] T. Ibaraki. Theoretical comparisons of search strategies in branch and bound algorithms. *Int. J. Comput. Inform. Sci.*, 5:315–344, 1976.

[11] L. G. Mitten. Branch and bound methods: general formulation and properties. *Operations Research*, 18:24–34, 1970.

# Optimization Algorithm for Improving the Efficacy of an Information Retrieval Model

Alberto Costa

*LIX, École Polytechnique, 91128 Palaiseau, France,* costa@lix.polytechnique.fr

**Abstract** The aim of Information Retrieval (IR) is to find the documents that are more relevant for a query, usually given by a user. This problem is very difficult, and in the last four decades a lot of different models were proposed, the most famous being the logical models, the vector space models, and the probabilistic models. In this paper is proposed a greedy algorithm for maximizing the efficacy of an Information Retrieval model based on Discrete Fourier Transform (DFT), which has shown a good efficacy level in the first tests. Even if the mathematical programming model used to increase the efficacy is a Mixed-Integer Nonlinear Program (MINLP), with nonlinear objective function and binary variables, its structure is very simple and a greedy algorithm can find the optimal solution.

**Keywords:** Information Retrieval, Efficacy, Greedy Algorithm, Discrete Fourier Transform

## 1. Introduction

Information Retrieval (IR) tries to solve this problem: given a query and a set of documents (collection), which are the relevant documents for the query?

The efficacy of an IR model depends on the number of relevant and non-relevant documents retrieved: the "perfect" IR model (that is the model with the maximum efficacy) should be able to retrieve all and only the relevant documents. Each time a non-relevant document is retrieved, or a relevant document is not retrieved, the efficacy decreases.

Several models were proposed in the last four decades, such as the logical models, the vector space models and the probabilistic models. The different techniques proposed by these models produced a significant increase of retrieval efficteveness in the last fifteen years, as experimentally observed within the Text REtrieval Conference (TREC) [8]. However, the current technology is far from being optimal, and the quest for new theoretical frameworks has been intense [3–5, 7].

Recently, a new IR model based on Discrete Fourier Transform (DFT) called Least Spectral Power Ranking (LSPR) was proposed, and it has shown good efficacy level in the first tests [2]. In this model the input is a collection of a document and a query, while the output is the ranking list, that is the list of the retrieved documents ordered from the most to the least relevant. The important thing to remark is that each document is associated with a score (called power in the LSPR model), such that if a document has a low power, it is considered highly relevant by the system, hence, the documents are ordered by increasing power. This is why the model is called *Least Spectral Power Ranking*.

In this paper an algorithm is proposed, which tries to increase the efficacy of LSPR: starting from the ranking list, this algorithm removes the documents that are not relevant with high probability. Basically, the problem of choosing the documents that maximize the efficacy can be described as a Mixed-Integer Nonlinear Program (MINLP), with a quadratic objective function and binary variables. However, due to the structure of the problem, a simple greedy algorithm can find the optimal solution.

The remainder of the paper is organized as follows. In section 2 there is a more formally explanation of the concept of efficacy. After that, in section 3 is presented the algoritm. Finally in section 4 there are the conclusions.

## 2. Evaluation of an IR system

The most important parameters for evaluating an IR system are:

- efficiency, that refers to the time complexity and the memory occupation;
- efficacy, that refers to the quality of the results.

In order to evaluate the efficacy, the so called "experimental collections" were introduced. An experimental collection is composed of a collection of documents, a set of queries and the relevance judgements; the latter is the list of the relevant documents for each query. In this way, comparing the documents retrieved by the system with the relevance judgements, it is possible to have an indication about the efficacy.

Among the most used parameters there are:

- precision: ratio between the number of relevant documents retrieved and the number of retrieved documents; is a measure of accuracy of search,
- recall: ratio between the number of relevant documents retrieved and the number of relevant documents; is a measure of completeness of search.

It is easy to see that if the number of document retrieved increases, the precison decreases and the recall increases.

In recent years, other measures have become more common, such as the *Mean Average Precision* (MAP), that is the average of the precision value obtained for the top $k$ documents, each time a relevant document is retrieved, or graphically it is roughly the average area under the precision-recall curve for a set of queries. The MAP varies from 0 to 1; in the tests performed in [2], using the CACM experimental collection,[1] the MAP of the vector-space model was 0.242, while the MAP of DFR was 0.329. The MAP of LSPR was 0.348, thus indicating a good performance comparable to the state-of-the-art.

## 3. Greedy algorithm

In this section an algorithm for increasing the MAP of the LSPR model is described.

Suppose there are a query $Q$ and a collection $C$ as input for LSPR, and the output is ranked list $R$, whose each document $i$ is associated with a power $Pw_i$: the less the power, the more the relevance.

The first step is to compute, for each document $i$ in the ranking list, a probability $p_i$ to be relevant. The informations given by LSPR can be very useful for this scope. Let $Pw_m$ and $Pw_M$ be respectively the power associated with the first and the last document in the ranking list $R$; a simple way to compute $p_i$ can be the following:

$$p_i = \frac{Pw_i - Pw_M}{Pw_m - Pw_M}. \tag{1}$$

It is easy to see that the probability is from 0 to 1: 0 if the power of the document is $Pw_M$ (that is the last document retrieved), 1 if the power of the document is $Pw_m$ (that is the first document retrieved.

[1] The collection can be found on `http://www.search-engines-book.com/collections`.

In order to maximize the MAP, we should maximize both precision and recall, leading to a multiobjective model. Furthermore, the recall depends on the number of relevant documents for a query, but usually this information is not available, so we have to semplify the problem.

Let $x_i \in \{0,1\}$ be a variable that is 1 if the document $i$ is selected, 0 otherwise. Precision and recall are rounded respectively as:

- precision= $\frac{\sum_{i=1}^{|R|} p_i x_i}{\sum_{i=1}^{|R|} x_i}$
- recall= $\frac{\sum_{i=1}^{|R|} p_i x_i}{N(Q)}$

where $N(Q)$ is the unknown number of relevant documents for the query, and the sums at the numerators play the role of the number of relevant documents retrieved.

At this point it is possible to semplify both the problem of the multiobjective function and the unknown value of $N(Q)$ by maximizing the product of recall and precision. Thus, the objective function to maximize is the following:

$$f(x) = \frac{\left(\sum_{i=1}^{|R|} p_i x_i\right)^2}{N(Q) \cdot \sum_{i=1}^{|R|} x_i}. \tag{2}$$

Since $N(Q)$ is a constant number, even if unknown, we can remove it from the objective function. The final MINLP model is

$$\begin{array}{ll} \max & \dfrac{\left(\sum_{i=1}^{|R|} p_i x_i\right)^2}{\sum_{i=1}^{|R|} x_i} \\ \text{s.t.} & x_i \in \{0,1\} \quad \forall i \in \{1,2,\ldots,|R|\} \end{array}$$

The greedy algorithm that solves this problem is very simple: first, the documents are ordered by decreasing probability to be relevant (i.e. $p_i \geq p_{i+1}, \forall i \in \{1,2,\ldots,|R|-1\}$). After that, we try to add the documents, from the first to the last in the ordered list, until the objective function increases. As soon as the objective function decreases, the algorithm stops; this is summarized in the following pseudo-code.

```
greedy_select {
    ⋆ call LSPR, to get the ranking list $R$ of the documents and the powers ⋆
    ⋆ starting from the powers, compute the probability, for example using Eq. (1) ⋆
    ⋆ order the documents by decreasing probability to be relevant ⋆
    $f \leftarrow 0$
    $x_i \leftarrow 0, \forall i \in \{1,2,\ldots,|R|\}$
    for $d \leftarrow 1$ to $|R|$ do
    {
      $x_d \leftarrow 1$
      $f_d \leftarrow \frac{\left(\sum_{i=1}^{d} p_i x_i\right)^2}{\sum_{i=1}^{d} x_i}$
      if ($f_d < f$)
          $x_d \leftarrow 0$
          break
      else
          $f \leftarrow f_d$
    }
    ⋆ return the documents $i$ for which $x_i = 1$, ordered by increasing $p_i$ ⋆
}
```

## 4. Conclusion and future work

This paper presents a possible way to increase the efficacy of an IR model. Actually, this technique could also be the base of a stand-alone IR model.

The important thing is, for each document, the computation of the probabilities to be relevant. In this paper a simple idea is proposed (see Eq. (1)), but more sophisticated techniques, which take account of the distribution of the powers, should led to better results.

Removing the documents from the ranking list has other advantages: the user has less document to check, and also the efficiency increases, because the memory occupation for the list of documents decreases.

Future work has 2 main objectives: First, this idea needs to be tested with some experimental collections. Second, a more precise mathematical description of the precision and the recall, and consequentely of the MAP, should be found. In this way the efficacy should increase, even if the model probably will be solved by some Nonlinear Global Optimization solver, such as BARON [6] or Couenne [1], instead of a simple greedy algorithm.

## Acknowledgments

The author is grateful to Digiteo Project 2009-55D "ARM" for financial support.

## References

[1] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, 24(4):597–634, 2009.

[2] A. Costa and M. Melucci. An information retrieval model based on discrete fourier transform. In H. Cunningham, A. Hanbury, and S. Rüger, editors, *Proceedings of the 1st Information Retrieval Facility Conference*, volume 6107 of *Lecture Notes in Computer Science*, pages 84–99, Vienna, 2010. Springer, Heidelberg.

[3] W.B. Croft and J. Lafferty, editors. *Language Modeling for Information Retrieval*. Springer, 2003.

[4] N. Fuhr. A probability ranking principle for interactive information retrieval. *Journal of Information Retrieval*, 11(3):251–265, 2008.

[5] S. Robertson. Salton award lecture: On theoretical argument in information retrieval. *SIGIR Forum*, 34(1):1–10, 2000.

[6] N.V. Sahinidis and M. Tawarmalani. *BARON 7.2.5: Global Optimization of Mixed-Integer Nonlinear Programs*, User's Manual, 2005.

[7] C.J. van Rijsbergen. *The geometry of information retrieval*. Cambridge University Press, 2004.

[8] E.M. Voorhees and D.K. Harman, editors. *TREC: Experiment and evaluation in information retrieval*. The MIT Press, Cambridge, MA, USA, 2005.

Proceedings of TOGO 2010, pp. 39 – 42.

# The Optimization Test Environment

Ferenc Domes,[1] Martin Fuchs,[2] and Hermann Schichl[1]

[1] *University of Vienna, Faculty of Mathematics, Vienna, Austria*

[2] *CERFACS, Parallel Algorithms Team, Toulouse, France*

**Abstract** Testing is a crucial part of software development in general, and hence also in mathematical programming. Unfortunately, it is often a time consuming and little exciting activity. This naturally motivated us to increase the efficiency in testing solvers for optimization problems and to automatize as much of the procedure as possible.

**Keywords:** test environment, optimization, solver benchmarking, solver comparison

The testing procedure typically consists of three basic tasks: a) organize test problem sets, also called test libraries; b) solve selected test problems with selected solvers; c) analyze, check and compare the results. The TEST ENVIRONMENT is a graphical user interface (GUI) that enables to manage the tasks a) and b) interactively, and task c) automatically.

The TEST ENVIRONMENT is particularly designed for users who seek to

1. adjust solver parameters, or
2. compare solvers on single problems, or
3. evaluate solvers on suitable test sets.

The first point considers a situation in which the user wants to improve parameters of a particular solver manually, see, e.g., [5]. The second point is interesting in many real-life applications in which a good solution algorithm for a particular problem is sought, e.g., in [10] (all for black box problems). The third point targets general benchmarks of solver software. It often requires a selection of subsets of large test problem sets (based on common characteristics, like similar problem size), and afterwards running all available solvers on these subsets with problem class specific default parameters, e.g., timeout. Finally all tested solvers are compared with respect to some performance measure.

In the literature, such comparisons typically exist for **black box** problems only, see, e.g., [17] for global optimization, or the large online collection [16], mainly for local optimization. Since in many real-life applications models are given as black box functions (e.g., the three examples we mentioned in the last paragraph) it is popular to focus comparisons on this problem class. However, the popularity of **modeling languages** like AMPL and GAMS, cf. [1], [9], that formulate objectives and constraints algebraically, is increasing. Thus first steps are made towards comparisons of global solvers using modeling languages, e.g., on the Gamsworld website [11], which offers test sets and tools for comparing solvers with interface to GAMS.

One main difficulty of solver comparison is to determine a reasonable criterion to **measure the performance** of a solver. For our comparisons we will count for each solver the number of global solutions found, and the number of wrong and correct claims for the solutions. Here we consider the term global solution as the best solution found among all solvers.

A severe showstopper of many current test environments is that it is uncomfortable to use them, i.e., the library and solver management are not very user-friendly, and features like automated LaTeX table creation are missing. Test environments like CUTEr [13] provide a test library, some kind of modeling language (in this case SIF) with associated interfaces to the solvers to be tested. The unpleasant rest is up to the user. However, our interpretation of the term test environment also requests to analyze and summarize the results **automatically** in a way that it can be used easily as a basis for numerical experiments in scientific publications. A similar approach is used in Libopt [12], available for Unix/Linux, but not tailored to optimization problems. It provides test library management, library subset selection, solve tasks, all as (more or less user-friendly) console commands only. Also it is able to produce performance profiles from the results automatically. The main drawback is the limited amount of supported solvers, restricted to black box optimization.

Our approach to developing the Test Environment is inspired by the experience made during the comparisons reported in [19], in which the Coconut Environment benchmark [22] is run on several different solvers. The goal is to create an easy-to-use library and solver management tool, with an intuitive GUI, and an easy, multi-platform installation. Hence the core part of the Test Environment is **interactive**. We have dedicated particular effort to the interactive library subset selection, determined by criteria such as a minimum number of constraints, or a maximum number of integer variables or similar. Also the solver selection is done interactively.

The modular part of the Test Environment is mainly designed as **scripts** without having fixed a scripting language, so it is possible to use Perl, Python, etc. according to the preference of the user. The scripts are interfaces from the Test Environment to solvers. They have a simple structure as their task is simply to call a solve command for selected solvers, or simplify the solver output to a unified format for the Test Environment. A collection of already existing scripts for several solvers is available on the Test Environment website [4]. We explicitly **encourage** people who have implemented a solve script or analyze script for the Test Environment to send it to the authors who will add it to the website. By the use of scripts the modular part becomes very flexible. For many users default scripts are convenient, but just a few modifications in a script allow for non-default adjustment of solver parameters without the need to manipulate code of the Test Environment. This may significantly improve the performance of a solver.

As **problem representation** we use Directed Acyclic Graphs (DAGs) from the Coconut Environment [14]. We have decided to choose this format as there already exist automatic conversion tools inside the Coconut Environment from many modeling languages to DAGs and vice versa. The Test Environment is thus designed to be independent from any choice of a modeling language. Nevertheless benchmark problem collections, e.g., given in AMPL such as COPS [3], can be easily converted to DAGs.

The summarizing part of the Test Environment is managing **automated tasks** which have to be performed manually in many former test environments. These tasks include an automatic check of solutions, and the generation of LaTeX tables that can be copied and pasted easily in numerical result sections of scientific publications. As mentioned we test especially whether global solutions are obtained and correctly claimed.

Using the Test Environment we have performed a benchmark of eight solvers on constrained global optimization and constraint satisfaction problems using three libraries with more than 1000 problems in up to about 20000 variables, arising from the Coconut Environment benchmark [22]. We have removed some test problems from the 2003 benchmark that had incompatible DAG formats. Thus we have ended up with in total 1286 test problems.

### Benchmark test results

The tested solvers in alphabetical order are: BARON 8.1.5 [20] (global solver), COCOS [14] (global), COIN with Ipopt 3.6/Bonmin 1.0 [15] (local solver), CONOPT 3 [7] (local), KNITRO 5.1.2 [2] (local), Lindoglobal 6.0 [21] (global), MINOS 5.51 [18] (local), Pathnlp 4.7 [8] (local). COCOS and KNITRO accepted (almost) all test problems. Also the other solvers accepted the majority of the problems. Minos accepted the smallest number of problems, i.e., $81\%$ of the problems. A typical reason why some solvers reject a problem is that the constraints of the objective function could not be evaluated at the starting point $x = 0$ because of the occurrence of expressions like $1/x$ or $\log(x)$. Some solvers like Baron also reject problems in which $\sin$ or $\cos$ occur in any expression.

Lindoglobal has the best score ($79\%$) in the number of correctly claimed global solutions among the global solutions found. COCOS is second with $76\%$, and Baron is third with $69\%$. But it should be remarked that Lindoglobal made $15\%$ wrong solution claims as opposed to Baron with $8\%$. Not surprisingly, the local solvers had only very bad scores in claiming global solutions, since they are not global solvers. On the other hand, they had a low percentage of wrong solutions, between $3\%$ and $8\%$ (except for KNITRO). The local solvers did not have zero score in claiming global solutions since for some LP problems they are able to claim globality of the solution.

Baron has found the most global solutions among all accepted problems ($71\%$). The local solver Coin also performed very well in this respect ($65\%$), at the same level as the global solver Lindoglobal. The other solvers are not far behind (except for KNITRO with $47\%$– however, it should be noted that for license reasons we used the quite old KNITRO version 5.1.2). New results with updated versions are continuously uploaded to the TEST ENVIRONMENT website [4]. For more details the interested reader is referred to [6].

## Acknowledgments

Partial funding of the project is gratefully appreciated: Ferenc Domes was supported through the research grant FS 506/003 of the University of Vienna. Hermann Schichl was supported through the research grant P18704-N13 of the Austrian Science Foundation (FWF).

Furthermore, we would like to acknowledge the help of Oleg Shcherbina in several solver and test library issues. We thank Nick Sahinidis, Alexander Meeraus, and Michael Bussieck for the support with several solver licenses. Thanks to Mihaly Markot who has resolved several issues with COCOS. We also highly appreciate Arnold Neumaier's ideas for improving the TEST ENVIRONMENT, and the comments by Yahia Lebbah.

## References

[1] A. Brooke, D. Kendrick, and A. Meeraus. *GAMS: A User's Guide*. The Scientific Press, 1988.

[2] R.H. Byrd, J. Nocedal, and R.A. Waltz. *Large-Scale Nonlinear Optimization*, chapter KNITRO: An Integrated Package for Nonlinear Optimization, pages 35–59. Springer, 2006.

[3] E.D. Dolan, J.J. Moré, and T.S. Munson. Benchmarking optimization software with COPS 3.0. Technical Report ANL/MCS-273, Mathematics and Computer Science Division, Argonne National Laboratory, 2004.

[4] F. Domes. TEST ENVIRONMENT website, `http://www.mat.univie.ac.at/~dferi/testenv.html`, 2009.

[5] F. Domes. GloptLab - A configurable framework for the rigorous global solution of quadratic constraint satisfaction problems. *Optimization Methods and Software*, 24(4-5):727–747, 2009.

[6] F. Domes, M. Fuchs, and H. Schichl. The optimization test environment. Submitted, 2010. Preprint available on-line at: `http://www.mat.univie.ac.at/~dferi/testenv.html`.

[7] A.S. Drud. CONOPT: A GRG code for large sparse dynamic nonlinear optimization problems. *Mathematical Programming*, 31(2):153–191, 1985.

[8] M.C. Ferris and T.S. Munson. Interfaces to PATH 3.0: Design, implementation and usage. *Computational Optimization and Applications*, 12(1):207–227, 1999.

[9] R. Fourer, D.M. Gay, and B.W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. Duxbury Press/Brooks/Cole Publishing Company, 2002.

[10] K.R. Fowler, J.P. Reese, C.E. Kees, J.E. Dennis, C.T. Kelley, C.T. Miller, C. Audet, A.J. Booker, G. Couture, R.W. Darwin, M.W. Farthing, D.E. Finkel, J.M. Gablonsky, G. Gray, and T.G. Kolda. Comparison of derivative-free optimization methods for groundwater supply and hydraulic capture community problems. *Advances in Water Resources*, 31(5):743–757, 2008.

[11] Gamsworld. Performance tools, `http://gamsworld.org/performance/tools.htm`, 2009.

[12] J.C. Gilbert and X. Jonsson. LIBOPT - An environment for testing solvers on heterogeneous collections of problems - The manual, version 2.1. Technical Report RT-331 revised, INRIA, 2009.

[13] N.I.M. Gould, D. Orban, and P.L. Toint. CUTEr and SifDec: A constrained and unconstrained testing environment, revisited. *ACM Transactions on Mathematical Software*, 29(4):373–394, 2003.

[14] H. Schichl et al. The COCONUT Environment, 2000–2010. Software.

[15] R. Lougee-Heimer. The Common Optimization INterface for Operations Research. *IBM Journal of Research and Development*, 47(1):57–66, 2003.

[16] H. Mittelmann. Benchmarks, `http://plato.asu.edu/sub/benchm.html`, 2009.

[17] M. Mongeau, H. Karsenty, V. Rouze, and J.B. Hiriart-Urruty. Comparison of public-domain software for black box global optimization. *Optimization Methods and Software*, 13(3):203–226, 2000.

[18] B.A. Murtagh and M.A. Saunders. MINOS 5.5 user's guide. Technical Report SOL 83-20R, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, California, 1983. Available on-line at: `http://www.sbsi-sol-optimize.com/manuals/Minos%20Manual.pdf`.

[19] A. Neumaier, O. Shcherbina, W. Huyer, and T. Vinko. A comparison of complete global optimization solvers. *Mathematical programming*, 103(2):335–356, 2005.

[20] N.V. Sahinidis and M. Tawarmalani. BARON 7.2.5: Global optimization of mixed-integer nonlinear programs. User's Manual, 2005. Available on-line at: `http://www.gams.com/dd/docs/solvers/baron.pdf`.

[21] L. Schrage. *Optimization Modeling with LINGO*. LINDO Systems, 2008.

[22] O. Shcherbina, A. Neumaier, D. Sam-Haroud, X.H. Vu, and T.V. Nguyen. *Global Optimization and Constraint Satisfaction*, chapter Benchmarking global optimization and constraint satisfaction codes, pages 211–222. Springer, 2003.

# Air traffic conflict resolution via light propagation modeling[*]

Nour Dougui,[1,2] Daniel Delahaye,[1] Marcel Mongeau[2,3] and Stephane Puechmorel[1]

[1] *Applied Mathematics Laboratory, ENAC, 7, Avenue Edouard Belin, 31055 Toulouse, France,*
{nour,delahaye,puechmor}@recerche.enac.fr

[2] *Université de Toulouse; UPS, INSA, UT1, UTM; Institut de Mathématiques de Toulouse; F-31062 Toulouse, France,*
mongeau@math.univ-toulouse.fr

[3] *CNRS; Institut de Mathématiques de Toulouse UMR 5219; F-31062 Toulouse, France*

**Keywords:** Branch and Bound, Geodesic

## 1. Introduction

The analysis of air traffic growth expects a doubling in the flights number over the next 20 years. The Air Traffic Management (ATM) will therefore have to absorb this additional burden and to increase the airspace capacity, while ensuring at least equivalent standards of safety.

The European project SESAR was initiated to propose solutions to this problem. It relies on a new concept of air traffic control, known as 4D (3D + time) trajectory planning, which consists in exploiting the new Flight Management System (FMS) abilities that ensure that the aircraft is at a given position at a given moment. For each flight, a reference trajectory, called Reference Business Trajectory (RBT), is requested by the operating airline. During the flight, conflict situations may nevertheless occur, in which two or several aircraft can dangerously approach each other. In this case, it is necessary to modify one or more trajectories to ensure that minimum separation standards (currently 5 Nm horizontally and 1000 ft vertically) are still satisfied. Moreover, it is desirable that proposed new trajectories deviate as little as possible from RBTs.

Several methods have been tested to find an optimal solution to adress this problem including genetic algorithm[1] and navigation function based approach[2]. The first approach can not gurantee a feasible (conflict-free) solution for a given time computing. The second one does not take into account the constraints imposed by ATM, such as bounded velocity.

## 2. Light Modeling Algorithm

We propose a new methodology, based on an optical analogy, which seeks to ensure sufficient separation between aircraft while producing flyable trajectories.

[*] ACKNOWLEDGEMENT: This work has been supported by French National Research Agency (ANR) through COSINUS program (project ID4CS n ANR-09-COSI-005)

## 2.1 Problem

The objective of our approach is to find for each aircraft a feasible (relevant to ATM constraints) optimal 4D trajectory, avoiding conflicts and wich minimizes a criterion based on a local metric. We consider here the following simplified problem: we want to determine the trajectory of one aircraft given that we know about the surrounding aircraft trajectories. In order to exploit future FMS capabilities,we represent an aircraft trajectory by a sequence of 4D points connected by line segments and by velocity 3D vectors (one such vector for each 4D point).

## 2.2 Light Modeling

We use light propagation analogy. Light propagates in space under *Descartes law* [5]: the trajectory of a light ray is the shortest path in time. The distance and travel time are correlated by a local metric called index. The analogy we use is to replace the index by a cost function for the aircraft trajectory: we consider the refractive index as a measure of congestion or so-called *traffic complexity*. We select a barrier index value in the prohibited areas, such as military areas, and in the protection volumes surrounding each aircraft. We compute the environment index associated to a given congested area (detail can be found in [3]). The optimal trajectory will be computed using a technique of ray tracing. The light will be slowed down in congested areas, but despite this, it can pass through. However, it will be completely blocked by aircraft protection volumes, which ensures conflict free-situations.The idea of our methodology consists in launching several light rays in various directions from the departure point of the aircraft, then the path of the first ray that reaches the arrival point corresponds to a geodesic approximation, hence a good flyable trajectory for the controlled aircraft.

## 2.3 Branch and Bound Algorithm

In order to compute this trajectory, we use a wavefront propagation algorithm in 3D with a time discretization (the wave propagation is done with a time step *dt*) from the departure point.

We implement the propagation with a *branch-and-bound algorithm*(B&B) [4], a classical framework for solving discrete optimization problems. The initial step of a B&B is to consider the set of all possible solutions, represented by the root of an enumeration tree. Procedures to obtain lower and upper bounds for the optimal value of our obective function (trajectory time travel) are applied to the root. If these two bounds are equals, then the optimal solution is found, and the algorithm stops. Otherwise, the solution set is partitioned into several sub-problems (new nodes). The method is then applied recursively on these sub-problems, generating a tree. If an optimal solution is found for a sub-problem, it is feasible but not necessarily optimal for the original problem. But, as a feasible solution, it can be used to eliminate partial solutions. The search goes on until all the nodes are explored or eliminated.

For the implementation of our light propagation case, a lower approximate bound for a given node is obtained as follows: we first compute a duration, "*TimeToDest*", for the remaining

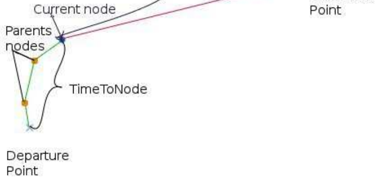


*Figure 1.* The lower bound computing.

time to reach the destination. This duration is a weighted sum of two terms (Formula 1 with $\alpha$ a weighting parameter). The first one, "*integTime*", is the time to reach destination considering the refractive index along the direct route. The second one, "*maxSpeedTime*", is the time needed to reach destination in direct route with the maximum speed.

$$TimeToDest := \alpha * integTime + (1 - \alpha) * maxSpeedTime. \tag{1}$$

The lower approximate bound is then the summation of *TimeToDest* and the time needed to reach the node from the origin (*TimeToNode*). It is given by (see Figure 1):

$$lowerBound := TimeToNode + TimeToDest.$$

Branching, in our context, involves launching rays as straight lines in a spatial cone of given radius $dt$, given steps $d\theta$ horizantally and $d\varphi$ vertically oriented towards the arrival point.

Browsing the search tree can be done in different ways. We choose a strategy whose priority is to find quickly a feasible solution (depth-first search or DFS). Here a node for which children have not yet been generated, with deepest level in the search tree, is chosen for exploration. DFS is then combined with a selection strategy. This consists in selecting the node that has the best lower bound among the nodes at the same level in the search tree (combination of DFS as the overall principle and best first search as a secondary selection criterion).

**1.** Set TrajSolution := null. Set upperBound := $\infty$

**2.** Discretize the cone towards the destination, whose center is Departure point and the radius is $dt$, with an angle steps $d\theta$ horizontally and an angle steps $d\varphi$ vertically.

**3.** While there is still unexplored nodes in the tree do:

- Choose a node N. If distance (N, destination point)$\leq \epsilon$ then TrajSolution := Set of points that leads to N and upperBound:= value of node N.
- Relaunch rays from node N in the cone towards the destination: For any light ray, if the light beam goes from a region with index $n_1$ into a region with index $n_2$ with an angle $i_1$, let it continue with a new angle $i_2$ such that $n_1$ sin$(i_1)$ = $n_2$ sin$(i_2)$ and with a velocity of $v = \frac{c}{n_2}$ where c is the light speed.
- Remove node N from the tree. Calculate node N's son values. Add them to the tree.

## 3. Numerical Results

Let us test our approach on a simplified instance of the problem, first in 2D then in 2D+time.

We use a coordinate system that is scaled with separation standards. Thus, we use an $(x, y)$ grid with a standard horizontal separation (5 Nm) unit. We set the radius $dt$ of the cone to the required time to travel a half standard separation distance. The cone maximum angle is set to $\frac{\Pi}{3}$. And the sampling angle $d\theta$ is set to $\frac{\Pi}{10}$. The weighting coefficients in the formula (1), is set to $\alpha := 0.9$.

### 3.1 Results in 2D

We first test our methodology on a 2D space instance to show it does find geodesics in simple cases.

Several refractive index functions were tested. For instance, index function used in Figures 2 is $\sum_{i=0}^{4} e^{-((x-a_i)^2+(y-b_i)^2)/k}$. It is a continuous function. High values (congested areas) are represented in red and low values (involving little traffic) in blue.

As can be seen in grey on Figure 2, the trajectory generated by our B&B algorithm avoids high index area and passes through "valleys", as one would expects. Thus, the aircraft avoids automatically congested areas.

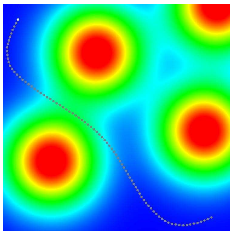


*Figure 2.* Resulting trajectory in 2D space.

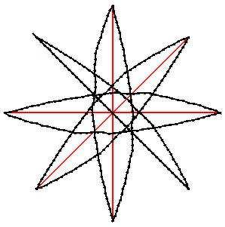


*Figure 3.* Conflict resolution with 8 aircraft

## 3.2 Results in 2D + time

Let us now consider a a 2D + time instance involving $P$ aircraft which are initially positioned along a circle of radius 100 Nm, converging at identical speed (450 knots velocity) towards the circle center. At any time, each of the $P$ aircraft has a position $(\overrightarrow{X_i})$. For any space point $\overrightarrow{Y}$, let us denate $\alpha := ||\overrightarrow{X_i} - \overrightarrow{Y}||$.

The used refractive index function we shall define must take into account avoidance of other aircraft protection zones. In order to ensure that the aircraft controlled by the algorithm avoids the other aircraft, we represent them by disks (whose radius is the standard distance separation),and we set the index function, $n$, to a very high constant value $N$ inside these disks and we make it decrease rapidly outside the disk. The index function $n$ is given by the following formula at any point $\overrightarrow{Y} \in \mathbf{R}^3$:

$$n(\overrightarrow{Y}) = \begin{cases} N & if \quad \alpha \leq R \\ 1 + \frac{N-1}{1+(\alpha-R)^q} & \quad otherwise. \end{cases}$$

with $R$ the standard distance separation and $q$ is a parameter that determines the speed with which the index decreases outside the separation zone. Our algorithm is sequentially applied to each aircraft until there is no conflict any more with P:= 8, N:= 2 and q:= 2.

We obtained a conflict free situation with the last aircraft that does not deviate from its direct route as displayed on Figure 3.

# 4. Conclusion

Our overall original light modelling methodology seems viable as it managed to resolve an academic conflict situation in (2D + time). Future work will concentrate on real-world instances and implementing a (3D + time) version of the algorithm.

# References

[1] N. Durand. *Optimisation de trajectoires pour résolution de conflits en route*, Phd thesis, ENSEEIHT, 1996.

[2] G. Roussos, G. Chaloulos, K. Kyriakopoulos, and J. Lygeros. *Control of multiple non holonomic air vehicles under wind uncertainty using model predictive control and decentralized navigation function*, IEEE Conference on Decision and Control, 2008.

[3] S. Puechmorel and D. Delahaye. *Dynamical systems complexity with a view towards air traffic management applications*, IEEE Conference on Decision and Control, 2009.

[4] E. Balas and P. Toth. *Branch and Bound Methods, In The Traveling Salesman Problem*, John Wiley & Sons, 361-401, 1985.

[5] Giancoli, Douglas C. *Ondes, optique et physique moderne*, DeBoeck, 361-401, 1993.

# Interval Branch and Bound Algorithms dealing with Black-Box Constraints

Julien Fontchastagner[1] and Frédéric Messine[2]

[1] *GREEN, Institut National Polytechnique de Lorraine, France,* julien.fontchastagner@ensem.inpl-nancy.fr

[2] *ENSEEIHT, IRIT-CNRS, University of Toulouse, France,* frederic.messine@n7.fr

**Abstract** Interval Branch and Bound Algorithms have shown their intrinsic interest to solve exactly some difficult mixed non-convex and non-linear programs. A code named IBBA was developed to solve some design problems for electromechanical actuators. The idea of this paper is to extend the exact global optimization algorithm by inserting the direct use of an automatic numerical tool (based on a finite element method). This new methodology permit to solve more rationally design problems. Some numerical examples validate the interest of this new approach.

**Keywords:** Interval Arithmetic, Branch and Bound, Black-box constraint

## 1. Introduction

*IBBA* is a code that we developed in order to solve some particular design problems of electromechanical actuators (such as electrical machines), [4]. Because *IBBA* is based on interval arithmetic, it needs all the expressions of the equations of the optimization problem. That was exploited by using analytical models of actuators developed by researchers in electrical engineering since the 80th. Associating analytical models and *IBBA* code yields to an efficient and rational approach to solve these design problems which can be formulated as non-homogeneous mixed constrained global optimization ones:

$$\left\{\begin{array}{ll} \min\limits_{(x,y)\in(X,Y)\subseteq I\!\!R^n\times I\!\!N^m} & f(x,y) \\ \text{u.c.} & g_i(x,y)\leq 0,\ \forall i\in P=\{1,...,p\} \\ & h_j(x,y)=0,\ \forall j\in Q=\{1,...,q\} \end{array}\right. \tag{1}$$

where $f$ is a real function, $I\!\!R$ and $I\!\!N$ are respectively the real and the positive integer sets. Normally the discrete variables can also be boolean or categorical ones [4]. However, it is sufficiently general for this paper to use only these classical mixed formulation.

A large part of these kind of design problems were solved efficiently by using *IBBA* code with analytical models [1]. Nevertheless, before making the prototype of the optimized actuator it is preferable to validate the optimal solution by using a finite element code (because for developing analytical models some assumptions have to be done and it is less the case for a numerical model). Thus, some differences generally occur between some important characteristical values yielding the designer to correct some parameters of the solution. In [2], we developed an extension of *IBBA*, named *IBBA+NUMT* to solve a problem of type (1) but where one constraint has to be satisfied via a finite element code *NUMT*. This was extended recently when more than one constraint have to be taken into account.

In this paper, we will present in a more general case this extension of *IBBA*, that we name *IBBA_BBC* for *IBBA* with Black-Box Constraints (constraints which do not have an explicit form but which has to be computed via an algorithm).

## 2. Formulations of the problem

The problem that we are interested with is the following:

$$\begin{cases} \min\limits_{(x,y)\in(X,Y)\subseteq \mathbb{R}^n\times \mathbb{N}^m} & f(x,y) \\ \text{u.c.} & g_i(x,y)\le 0,\ \forall i\in I\subseteq P \\ & h_j(x,y)=0,\ \forall j\in J\subseteq Q \\ & g_i^A(x,y)\le 0,\ \forall i\in P\setminus I \\ & h_j^A(x,y)=0,\ \forall j\in Q\setminus J \end{cases} \tag{2}$$

where $A$ on the constraints indicates that they are computed using an algorithm.

Unfortunately, this problem is actually impossible to be solved by a code based on *IBBA*; because for the black-box constraints, it is not possible to compute bounds. Hence, we introduce a new more constrained program but related to Program (2):

$$\begin{cases} \min\limits_{(x,y)\in(X,Y)\subseteq \mathbb{R}^n\times \mathbb{N}^m} & f(x,y) \\ \text{u.c.} & g_i(x,y)\le 0,\ \forall i\in I\subseteq P \\ & h_j(x,y)=0,\ \forall j\in J\subseteq Q \\ & g_i^A(x,y)\le 0,\ \forall i\in P\setminus I \\ & h_j^A(x,y)=0,\ \forall j\in Q\setminus J \\ & g_i(x,y)\le \overline{s_i},\ \forall i\in P\setminus I \\ & \underline{r_j}\le h_j(x,y)\le \overline{r_j},\ \forall j\in Q\setminus J \end{cases} \tag{3}$$

where $\overline{s_i}, \underline{r_j}$ and $\overline{r_j}$ associated with the analytical equations define a *zone* where the global solution has to be searched. Of course, analytical constraints $g_i, \forall i\in P\setminus I$ and $h_j, \forall j\in Q\setminus J$ as to be related to the numerical ones, respectively $g_i^A, \forall i\in P\setminus I$ and $h_j^A, \forall j\in Q\setminus J$.

## 3. IBBA_BBC Algorithm

The following corresponds to the iterations of *IBBA_BBC* Algorithm:

```
1. Set $(X,Y) :=$ the initial hypercube.
2. Set $\tilde{f} := +\infty$ and set $\mathcal{L} := (+\infty,(X,Y))$.
3. Extract from $\mathcal{L}$ the lowest lower bound.
4. Bisect the considered box chosen by its midpoint: $V_1, V_2$.
5. For j:=1 to 2 do
   (a) $V_j :=$Propagation of the analytical constraints on $V_j$, [3].
   (b) if $V_j \neq \emptyset$ then
       i. Compute $v_j := lb(f, V_j)$.
       ii. Compute all the lower and upper bounds of all the analytical constraints on $V_j$.
       iii. if $\tilde{f} \geq v_j$ and no analytical constraint is unsatisfied then
            ■ insert $(v_j, V_j)$ in $\mathcal{L}$.
            ■ set $m$ the midpoint of $V_j$
            ■ if $m$ satisfies all the analytical constraints and then if the black-box
              constraints are also satisfied then $\tilde{f} := \min(\tilde{f}, f(m))$.
            ■ if $\tilde{f}$ is changed then remove from $\mathcal{L}$ all $(z,Z)$ where $z > \tilde{f}$ and set $\tilde{y} := m$.
6. If $\tilde{f} - \min\limits_{(z,Z)\in\mathcal{L}} z < \epsilon$ (where $z = lb(f,Z)$) then STOP.
   Else GoTo Step 4.
```

In this algorithm, one has: -$\tilde{f}$ which represents the current best known solution of program (3); -$\mathcal{L}$ is the list which contain all the boxes which can possibly have the global solution at each stage of the iterations; -*lb* represents a technique using interval arithmetic which allows to compute lower bounds of a function (explicitly known) over a box; -the constraint propagation code which is used is detailed in [3] and it is based on the calculus trees of the constraints (it is only for constraints which are explicitly known). Remark that there a few differences between *IBBA* and *IBBA_BBC*; the differences are underlined in the above algorithm.

Now, we want to prove that this algorithm *IBBA_BBC* find the global solution of Program (3). In order to prove that point, we introduce the following relaxed program:

$$\left\{\begin{array}{ll} \min\limits_{(x,y)\in(X,Y)\subseteq I\!R^n\times I\!N^m} & f(x,y) \\ \text{u.c.} & g_i(x,y)\leq 0,\ \forall i\in I\subseteq P \\ & h_j(x,y)=0,\ \forall j\in J\subseteq Q \\ & g_i(x,y)\leq \overline{s_i},\ \forall i\in P\setminus I \\ & \underline{r_j}\leq h_j(x,y)\leq \overline{r_j},\ \forall j\in Q\setminus J \end{array}\right. \tag{4}$$

where the black-box constraints are eliminated from Program (3). In fact, the main principle of *IBBA_BCC* is to solve the relaxed program (4) and just to consider the Program (3) for improving the current solutions $\tilde{f}$. Such a Branch and Bound algorithm proceeds by exclusions of boxes when it is proven that the global solution cannot be inside it. Thus, suppose that a box $Z$ which contains the global minimum of Program (3) is discarded by *IBBA_BBC*. $Z$ has only 3 distinct possibilities to be eliminated: (i) $Z=\emptyset$ after the propagation step. This implies that at least one constraint of the relaxed problem (4) is unsatisfiable, see [3]. Hence, no point in $Z$ can satisfy all the constraints of problem (4) and therefore of problem (3). Thus, this possibility cannot occur; (ii) using interval arithmetic, an analytical constraint is proved to be unsatisfiable for all points in $Z$. It is also an impossible case such as in (i); (iii) $\tilde{f}\geq z:=lb(f,Z)$. This implies that no point in $Z$ can have a value lower than $\tilde{f}$, therefore $Z$ does not contain a global minimum. Therefore, we show that these 3 possibilities cannot occur and then that such a box $Z$ cannot exist. Hence, *IBBA_BBC* provides a global minimum of Program (3).

## 4. Numerical experiments

To illustrate our methodology, we consider the design of an electromechanical machine with permanent magnets. See [1] for details about its mixed non-convex analytical formulation. For the computations of the electromagnetic torque of the machine, we use *NUMT* which is an efficient finite element numerical algorithm that we developed, [2]. Thus, here we have just one black-box constraint.

In Table 1, parameters of the machine are: -$D$ its bore diameter, -$L$ its length, -$l_a$ the thickness of the magnets, -$E$ the winding thickness, -$C$ the thickness of yoke, -$\beta$ the polar arc factor,-$k_d$ the filling factor, -$p$ the number of poles pairs, -$m$ the number of slots per pole and phase, -$b_r$ the rotoric configuration, -$b_f$ the kind of electromotive force waveform, -$\sigma_m$ the type of permanent magnet, -$\sigma_{mt}$ the type of magnetic conductor. On this example, we minimize the mass (see [2] for complementary results), the main constraint is about the torque of the machine which must be equal to 10 N·m (with a tolerance about $\pm 0.2$ N·m). In the column named IBBA, we solve program (1) with only the analytical model and just using *IBBA*. Remark that the numerical value for the torque (computed using *NUMT*) is equal to $9.12$ which is quite far from imposed value $10$. In column *IBBA_BBC*, we solve Program (3) with a zone of $10\%$ around 10 for the torque and using *NUMT*. Then, we obtain $9.83$ N·m for the numerical torque; notice that the analytical value is $10.21$. We emphasize in bold all the parameters which have changed between these two resolutions (nine parameters have changed including the integer ones).

In Table 2, we represent the solutions found when the zone is progressively increased: $2\%, 5\%$ and so on. We note that a zone about $5\%$ is sufficient to solve efficiently this design problem.

*Table 1.* Results for the minimization of the Mass.

| Parameter | Bounds | Unit | *IBBA* | *IBBA_BBC* |
|---|---|---|---|---|
| $D$ | $[0.01, 0.3]$ | m | $\mathbf{0.1430}$ | $\mathbf{0.1320}$ |
| $L$ | $[0.01, 0.3]$ | m | $\mathbf{0.0491}$ | $\mathbf{0.0519}$ |
| $l_a$ | $[0.003, 0.01]$ | m | $\mathbf{0.0034}$ | $\mathbf{0.0047}$ |
| $E$ | $[0.005, 0.03]$ | m | $\mathbf{0.0075}$ | $\mathbf{0.0082}$ |
| $C$ | $[0.003, 0.02]$ | m | $\mathbf{0.0040}$ | $\mathbf{0.0039}$ |
| $\beta$ | $[0.7, 0.9]$ | | $\mathbf{0.73}$ | $\mathbf{0.79}$ |
| $k_d$ | $[0.4, 0.6]$ | | $\mathbf{0.5020}$ | $\mathbf{0.5024}$ |
| $p$ | $[\![3, 10]\!]$ | | $\mathbf{8}$ | $\mathbf{9}$ |
| $m$ | $\{1, 2\}$ | | 1 | 1 |
| $b_r$ | $\{0, 1\}$ | | $\mathbf{0}$ | $\mathbf{1}$ |
| $b_f$ | $\{0, 1\}$ | | 1 | 1 |
| $\sigma_m$ | $\{1, 2\}$ | | 2 | 2 |
| $\sigma_{mt}$ | $\{1, 2\}$ | | 2 | 2 |
| Mass | | $kg$ | $\mathbf{2.92}$ | $\mathbf{3.35}$ |
| Analytical Torque | | N·m | $\mathbf{9.81}$ | 10.21 |
| Numerical Torque | | N·m | 9.12 | $\mathbf{9.83}$ |
| CPU - Time | | min | 0′51 | 34′44 |
| Iterations | | | 152,126 | 216,623 |
| Iterations of NUMT | | | - | 2,898 |

*Table 2.* Results when the zone varies.

| zone: $pc =$ | Mass | $\Gamma_{em}$ | NUMT | Time | Its | Its of NUMT |
|---|---|---|---|---|---|---|
| IBBA | 2.92 | 9.81 | 9.12 | 0′51 | 152,126 | – |
| 2% | 3.44 | 10.18 | 9.92 | 7′59 | 223,769 | 585 |
| **5%** | **3.35** | **10.21** | **9.83** | **17′08** | **213,094** | **1,404** |
| 10% | = | = | = | 34′44 | 216,623 | 2,898 |
| 20% | = | = | = | 79′29 | 223,118 | 7,162 |
| 30% | = | = | = | 159′25 | 228,324 | 14,751 |
| 40% | = | = | = | 281′09 | 231,513 | 25,004 |

# 5. Conclusion

We show in this paper that it is possible to extend IBBA code in order to take into account global optimization problems with black-box constraints. We prove that our new algorithm provides the global minimum of such a problem. Our new method is validated by solving a problem of design of an electromechanical machine with permanent magnets.

# References

[1] E. Fitan, F. Messine, and B. Nogarede, *The Electromagnetic Actuator Design Problem : A General and Rational Approach*, IEEE Transactions on Magnetics, Vol. 40, N. 3, pp. 1579–1590, May 2004.

[2] J. Fontchastagner, F. Messine, Y. Lefèvre, *Design of Electrical Rotating Machines by Associating Deterministic Global Optimization Algorithm With Combinatorial Analytical and Numerical Models*, IEEE Transcations on Magnetics, Vol. 43, N. 8, pp. 3411-3419, 2007.

[3] F. Messine, *Deterministic Global Optimization using Interval Constraint Propagation Techniques*, RAIRO Operations Research, Vol. 38, N. 4, 2004.

[4] F. Messine, *A Deterministic Global Optimization Algorithm for Design Problems*, chapter in Essays and Surveys in Global Optimization, Editors Charles Audet, Pierre Hansen et Gilles Savard, pp. 267–294, 2005.

# Solving the Pooling Problem with LMI Relaxations*

Lennart Frimannslund, Mohamed El Ghami, Mohammed Alfaki and Dag Haugland

*Department of Informatics, University of Bergen, Postboks 7803, 5020 Bergen, Norway.*
{lennart, mohamed, mohammed.alfaki, dag}@ii.uib.no

**Abstract** We discuss solving the pooling problem, a bilinear programming problem, and show how the global optimum of a subclass of this problem can be found without using branch-and-bound techniques.

**Keywords:** pooling problem, LMI relaxations, bilinear programming, semidefinite programming

## 1. Introduction

We study the *pooling problem*, a nonconvex network flow problem. For example, consider the flow problem instance shown in Figure 1, which models the flow of oil contaminated by sulfur from the source nodes (squares) to the sinks/customers (triangles), with the objective to maximize the profit. As one can see from the figure, there are bounds on the relative sulfur content at the sinks (nodes 5 and 6) which are such that the source node 1 cannot alone service any of the sinks, and the source node 3 cannot alone service sink 6. In addition, there is a node in the network (number 4) where two incoming flows meet, and then split up again. Such a node is called a pool, and is what gives rise to the name pooling problem. The quality of the flow leaving the pool is a weighted average of the qualities entering the pool, where the flow quantities constitute the weights.

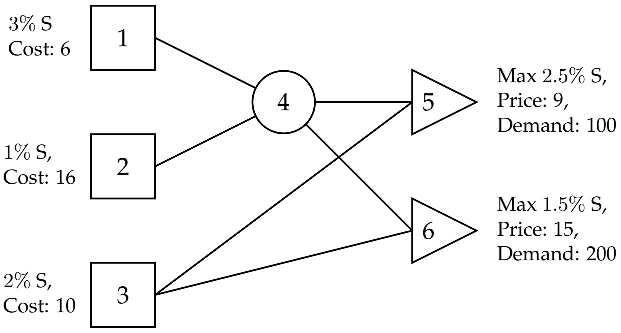


*Figure 1.* The Haverly1 pooling problem instance [2]. The direction of flow is from left to right.

Although there may be several quality components of interest (e.g. water in addition to sulfur), we will focus on the situation where there is only one component. Note also that the graph can be said to have three layers of nodes. First, there is a layer of sources, then a layer containing pools, and finally a layer of sinks. If we wish to maximize profit/minimize cost,

*This work was supported by the Norwegian Research Council, Gassco and StatoilHydro under contract 175967/S30.

then the problem instance in the figure can be written as:

$$\max\ 9(x_{45}+x_{35})+15(x_{36}+x_{46})-6x_{14}-16x_{24}-10(x_{35}+x_{36}),$$

such that

$$\begin{aligned}
x_{14}+x_{24}-x_{45}-x_{46} &= 0, && (1)\\
3x_{14}+x_{24}-q_4(x_{45}+x_{46}) &= 0, && (2)\\
q_4x_{45}+2x_{35} &\le 2.5(x_{45}+x_{35}),\\
q_4x_{46}+2x_{36} &\le 1.5(x_{46}+x_{36}),\\
x_{46}+x_{36} &\le 100,\\
x_{45}+x_{35} &\le 200,\\
\text{all variables} &\ge 0.
\end{aligned}$$

Here $q_4$ denotes the relative sulfur content of the oil exiting node 4, and $x_{ij}$ denotes the flow from node $i$ to node $j$. This is not the only possible objective function, for example if we want to maximize the flow to the sinks, the objective will be

$$\max x_{35}+x_{36}+x_{45}+x_{46}, \tag{3}$$

We will call this a max flow formulation. For both formulations there are bilinear terms, which are caused by the presence of the pool.

## 2. Solving the pooling problem

Existing algorithms for this problem (see e.g. [7] and the references therein) employ branch-and-bound techniques, and therefore have to solve an unknown, in the worst case exponential, number of subproblems. The reason for this is that there are exponentially many nodes in the branch-and-bound tree. We now present a methodology that does not make use of branching and bounding.

First, we need some background regarding a technique which can be used to solve constrained optimization problems which are polynomial. Since a pooling problem instance is a bilinear program and therefore is also polynomial (in fact, quadratic), we can use convex linear matrix inequality (LMI) relaxations [4, 5] to try to solve them. Let us write a general quadratic, quadratically constrained optimization problem as

$$\begin{aligned}
&\min_x f(x)\\
\text{subject to}\quad &g_j(x)\ge 0,\quad j=1,\dots,m.
\end{aligned} \tag{4}$$

Let $K$ denote the feasible domain. Solving (4) is equivalent to solving

$$\min_{\mu\in\mathcal{P}(K)}\int f(x)\,\mu(dx). \tag{5}$$

Here we minimize of over all possible Borel signed measures $\mu$ with support on $K$, which can be thought of as probability measures. Assuming, for simplicity, that $f$ has a unique global minimum $x^*$ on $K$, then the solution to (5), say, $\mu^*$ is the probability measure with its support only on $x^*$, and $x^*$ can be identified from the first-order moments of $\mu^*$. Finding a probability distribution with support on $K$ can be done by determining its moments. However, this is an infinite-dimensional optimization problem. It can be truncated by determining only moments up to a certain (finite) order. Finding such moments is precisely the (primal) LMI relaxation of (4), and the higher the order of the moments, the better the relaxation. See [4, 5] for a detailed description.

Instead of discussing the primal LMI relaxation, we focus on its dual. The dual can be viewed as finding polynomial weights so that one can write the expression $f(x) - f^*$ (the objective function minus its globally optimal value on the feasible domain) as a weighted sum of the constraints, where the weights are squared polynomials. That is, finding polynomials $q_j(x),\ j = 1, \ldots, r_0$ and $q_{kj}(x),\ j = 1, \ldots, r_{kj}$, if they exist, such that

$$f(x) - f^* = \sum_{j=1}^{r_0} [q_j(x)]^2 + \sum_{k=1}^{m} g_k(x) \left[ \sum_{j=1}^{r_k} [q_{kj}(x)]^2 \right]. \tag{6}$$

Since squared polynomials are always nonnegative and the constraint polynomials, as well as the expression $f(x) - f^*$ are always nonnegative on the feasible domain, finding the coefficients of the $q$-polynomials such that (6) holds for all $x$ on the feasible domain is equivalent to finding the global optimum $f^*$. The allowed degree of these polynomials is called the order of the relaxation, and is linked to how many moments are present in the primal problem. For quadratic objective functions and quadratic constraints (our case) and relaxation order $i$, $q_j$ may have at most degree $i$, and $q_{kj}$ at most degree $i - 1$, for all $k$ and $j$.

If the feasible set of (4) is compact and has a nonempty interior, then there exists a finite order for which (6) holds. Having found the $q$-polynomials and $f^*$, the corresponding optimal variables $x^*$ for the original problem (4) can be determined using the procedure decribed in [3], in most cases.

**Proposed Solution framework.** This enables us to solve the pooling problem with one quality in the following way:

1. Eliminate equalities from the formulation by variable substitution, *without* introducing rational expressions. Specifically, for each pool there will be a flow balance equation (1) and a similar equation regarding the flow of the quality component (2). Use these to eliminate two of the incoming edge flow variables.
2. If needed, remove obvious causes of empty interiors (e.g. sinks with unsatisfiable quality requirements).
3. Solve the reduced problem with LMI relaxations.
4. Assign values to eliminated variables.

Not all pooling problem instances can have their equalities removed while remaining polynomial. In particular, this is true for the case of multiple quality components, and for this reason we make no claim about the solvability of such instances. In addition, single-quality instances which cannot be made to have a nonempty interior cannot provably be solved for a finite LMI relaxation order. However, we note that the class where step 1 and 2 produce a useable formulation encompasses many instances, including the single-quality instances most commonly discussed in the literature.

After step 1 has been performed, min cost instances will have bilinear terms in the objective function, whereas the objective function of max flow instances will be unchanged. Since solving the problem amounts to constructing the expression $f(x) - f^*$, an expression which is linear for the max flow case, we suspect that solving max flow problems is actually easier than solving min cost problems.

**Computational Complexity.** Step 1, 2 and 4 have polynomial complexity. As for step 3, there are $O(n^{2i})$ decision variables in the LMI relaxation of order $i$, which can be cast as a semidefinite programming (SDP) problem. SDP problems can be solved in polynomial time using interior point methods. In other words, if there, for some subclass of the pooling problem, exists an $i_{\max}$ such that all instances in the class can be solved by an LMI relaxtion of order no larger than $i_{\max}$, then this class of problems is solvable in polynomial time. In addition we note that, being a network flow problem, the pooling problem should be suited to the sparsity-exploiting variant of LMI relaxations presented in e.g. [6], which allows for faster solution times.

## 3. Numerical experience

We test our solution framework on concrete instances to see if there is a trend regarding which relaxation order is needed to find the global optimum. First we solve the standard sinqle-quality test cases from the literature, all of which were solvable with a second-order LMI relaxation. The results are given in Table 1. All of these instances are defined in e.g. [1]. We have also solved randomly generated problem instances with nonempty interiors. In these experiments no min cost instance tested required a relaxation order larger than 3, and no max flow instance required a relaxation order larger than 2. Interestingly, the lower bound provided by relaxation order 1 was equal to the globally optimal objective function value for all maximum flow problem instances tested, but not all minimum cost problem instances.

| Name | Min cost | | Max flow | |
|---|---|---|---|---|
| | Order 1 | Order 2 | Order 1 | Order 2 |
| Haverly 1 | 600 | 400 | 300 | 300 |
| Haverly 2 | 1200 | 600 | 800 | 800 |
| Haverly 3 | 875 | 750 | 300 | 300 |
| Ben-Tal 4 | 600 | 450 | 300 | 300 |
| Foulds 2 | 1200 | 1100 | 600 | 600 |

*Table 1.* Results on standard test problem instances from the literature. The values are the objective function values corresponding to relaxation orders one and two, for both minimum cost and max flow variants of the test cases.

## 4. Summary

We use LMI relaxations for solving the single-quality pooling problem. Based on our experiments we suspect that most instances of this problem are solvable with a low LMI relaxation order, provided that they can be formulated in such a way that they have a nonempty interior.

## References

[1] N. Adhya, M. Tawarmalani, and N. V. Sahinidis. A Lagrangian approach to the pooling problem. *Industrial & Engineering Chemistry Research*, 38(5):1956–1972, 1999.

[2] C. A. Haverly. Studies of the behavior of recursion for the pooling problem. *ACM SIGMAP Bulletin*, 25:19–28, 1978.

[3] D. Henrion and J. B. Lasserre. Detecting global optimality and extracting solutions in GloptiPoly. In D. Henrion and A. Garulli, editors, *Positive Polynomials in Control*, pages 293–310. Springer Berlin/Heidelberg, 2005.

[4] J. B. Laserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

[5] J. B. Lasserre. New positive semidefinite relaxations for nonconvex quadratic programs. In H. Hadjisavvas and P. M. Paradalos, editors, *Advances in Convex Analysis and Global Optimization*, pages 319–331. Kluwer Academic Publishers, 2001.

[6] J. B. Lasserre. Convergent SDP-relaxations in polynomial optimization with sparsity. *SIAM Journal on Optimization*, 17(3):822–843, 2006.

[7] N. Sahinidis and M. Tawarmalani. Accelerating branch-and-bound through a modeling language construct for relaxation-specific constraints. *Journal of Global Optimization*, 32(2):259–280, 2005.

# Spectral unmixing based on a minimum volume simplicial enclosure model*

Inmaculada García,[1] Eligius M.T. Hendrix,[1] Javier Plaza,[2] and Antonio Plaza[2]

[1] *Universidad de Malaga, Dept. of Computer Architecture. 29080 Malaga, Spain,* {eligius, igarciaf}@uma.es

[2] *University of Extremadura, Dept. of Technology of Computers and Communications, Caceres, Spain,* aplaza@unex.es

**Abstract** This work describes the minimum volume enclosing simplex problem, which is known to be a multimodal Global Optimization problem. The problem has been used as a basis to estimate so-called endmember and abundance data in unmixing spectral data of hyperspectral sensors. This estimation problem is a big challenge. We explore the possibility of a new estimation algorithm using the minimum volume enclosing simplex problem. We investigate its behaviour numerically on designed instances comparing its outcomes with a maximum volume enclosed simplex approach which is used frequently in spectral unmixing.

**Keywords:** spectral unmixing, endmembers, principal components, optimization. minimum volume

## 1. Introduction

A challenging problem in having data from multispectral imaging sensors is to unfold them into components. We study here the possibility to do so using a minimum volume enclosing simplex approach. Hyperspectral sensors record scenes in which various disparate material substances contribute to the spectrum measured from a single pixel.

Spectral unmixing ([5]) is a term to denote a procedure to decompose a measured spectrum of a mixed pixel into a collection of constituent spectra (endmembers) and a set of corresponding fractions (abundances) that indicate the proportion of each endmember present in the pixel. Endmembers normally correspond to familiar macroscopic objects in the scene, such as water, soil, metal, or any natural or man-made material.

Many methods have been developed and tested to perform endmember extraction and unmixing, see [3] for an overview. We will focus on what is called linear unmixing and ask the question how one can recover the endmember and abundance data via unbiased estimators. One typically sees least squares approaches with the additional complication that the abundance estimate should lay on the unit simplex (nonnegativity). [4] takes an approach where two conflicting objectives, that of least squares and minimizing the volume of an enclosing simplex are combined in an objective function. Recently, [1] develop an approach where they apply sequential Linear programming to solve the minimum volume enclosing simplex problem. In this paper we use standard available nonlinear optimization algorithms to solve the problem.

The problem of enclosing a set of points with a minimum volume body leads usually to a Global Optimization problem; we will illustrate that for the generic simplicial enclosure

*This work has been supported by the Spanish Ministry of Science and Innovation through grants TIN2008-01117 and AYA2008-05965-C04-02. Eligius Hendrix is a fellow of the Spanish "Ramon y Cajal" contract program, co-financed by the European Social Fund.

this is the same. However, the use in spectral unmixing is far from worst case behaviour; instances are characterised by low noise and pixel data is well spread. A local search from a well designed starting body leads to the global optimum soon. We will take a hierarchical vision: First to minimize least squares using principal component analysis, which is very common in image data analysis and second, minimize the volume of an enclosing simplex in the reduced space. The question is how to use such an approach such that for linear mixture with white noise one obtains unbiased estimates of endmembers and abundance.

A benchmark method is to consider a maximum volume "inscribing" simplex looked for by the so-called N-FINDR algorithm [6]. Given the reduced data, in principle one looks for all combinations within the given pixels as candidate endmembers such that the resulting volume of the spanning simplex is maximum. If indeed the endmembers are present in the data and noise is low, the approach is very promising as analysed by [6]. We can use the results of such an approach to compare methods numerically.

## 2. Unmixing and minimum enclosing simplex

Let us assume that a hyperspectral scene contains $m$ spectral bands and $r$ pixel vectors. Each pixel vector in the scene ($y_k$) can be modeled using the following expression:

$$y_k = Xa_k + \epsilon \tag{1}$$

where $y_k$ is $m \times 1$ observation bands, $X$ is $m \times n$, bands of endmembers, $a_k$ is $1 \times n$ abundance and $\epsilon$ is $m \times 1$ white noise with a standard deviation of $\sigma$. Our goal is to design a method for recovering "real" matrix $X$ and abundance $a_k$ of observed pixels $y_k$. To do so, usually two objectives are minimized: noise in a least squares way and the volume of the simplex spanned by the columns of matrix $X$. Moreover, the abundance should be positive for each pixel. The question is how to deal with least squares and minimum volume in such a way that the estimation is unbiased, i.e. the expected value of the estimator is the real value.

One should keep in mind that instances of the problem consisting of real images are characterized by pixels being mixtures of less than 4 constituents, i.e. vectors $a_k$ have only a few positive values. The idea of least squares in the estimation procedure is, that often it is not know exactly how many endmembers, constituents, are involved in the data. Therefore application of principal component analysis is popular. Having $n$ endmembers gives that one should discover an $n-1$ dimensional subspace that is responsible for the main variation and the rest of the $m$ dimensional space is considered noise.

First of all the data are centralized by the mean $\overline{y}$, such that the columns of $Y$ consist of centralized observations $y_k - \overline{y}$. The observed variation in the spectral data $Y^TY$ is approximated by $(CZ)^TCZ$ where $C$ is an $m \times (n-1)$ matrix of principal components and $Z$ is $(n-1) \times r$ a so-called score matrix. In direction $c_1$ we have the biggest variation, in direction $c_2$ the second biggest etc. Essentially we have reduced model (1) to $z = Va + \xi$, where we expect the endmembers $X$ to lay in the space $< C > +\overline{y}$ spanned by the columns of $C$. $C$ represents an estimate of the space in which the endmember spectra $X$ are located, $X = CV + \overline{y}$. To say it in another way, with absence of noise the estimate of $C$ represents the space spanned by $X - \overline{y}\mathbf{1}^T$, where $\mathbf{1}$ is the all-ones vector of appropriate dimension. With noise, $\xi$ is now the projection of $\epsilon$ on $< C >$ and therefore its components also form white noise. To be consistent, we should theoretically notice that $y = Cz + \overline{y} + \zeta$ where $\zeta$ is the part of $\epsilon$ projected on the orthoplement of $< C >$; $\epsilon = \xi + \zeta$. We will use the idea that the noise of $z$ is componentwise independent.

We follow a two step approach often found in literature. First we estimate the space in which the $n$ endmembers are lying. Secondly, in that space, we minimize the volume of the resulting simplex such that it encloses the projections of the observed bands of the pixels. The N-FINDR algorithm follows an approach where on the projected plane the volume of a simplex is maximized.

## 3. Minimum volume versus maximum volume simplices

The estimate of the matrix of endmembers $X = CV + \overline{y}$ appears from an estimate of $V$ based on the projected bands (scores) $Z$. The problem of finding the minimum enclosing simplex of a set of points $z_k, k = 1, \ldots, r$ in $(n-1)$-dimensional space is

$$\min_{V}\{f(V) := \det \begin{pmatrix} V \\ \mathbf{1}^T \end{pmatrix}\} \texttt{ subject to } a_k = \begin{pmatrix} V \\ \mathbf{1}^T \end{pmatrix}^{-1} \begin{pmatrix} z_k \\ 1 \end{pmatrix} \geq 0, \;\; k = 1, \ldots, r \qquad (2)$$

Enclosing with shapes may lead to GO problems. [2] give several examples for enclosing with spheres (the Chebychev problem) and with hyper-rectangles. The use of the minimum volume problem for endmember identification is illustrated next.

In general, we will call $V$ the real values of endmembers defining simplex $S = conv(V)$ and use for the outer enclosing estimate $\hat{V}o$ and corresponding simplex $\hat{S}o$. In case all pixels would be convex combinations of (few) endmembers without any noise, the enclosing simplex $\hat{S}o$ obtains the endmembers $V$ as vertices despite they do not appear in the pixels. Literature on spectral unmixing also uses a maximum volume simplex perspective. The idea is that pure pixels representing the endmembers are present in the data set $Z$. Consider the pixel data as a set $\mathbb{Z}$. One wants to find a subset $\mathbb{V}$ with $|\mathbb{V}| = n$ such that the corresponding simplex has maximum volume; i.e. $\max_{\mathbb{V} \subset \mathbb{Z}} \; f(V)$, where $V$ is a matrix with the columns of $\mathbb{V}$. This defines a combinatorial optimization problem. The N-FINDR algorithm is a so-called local search heuristic in combinatorial optimisation context. We used a MATLAB implementation of N-FINDR as reference method to compare to MINVEST described in Section 4.

## 4. Minimum volume estimation procedure: MINVEST

The minimum volume simplex $\hat{S}o$ gives an accurate estimate of the endmembers if noise is absent. That is, sufficiently many pixels should lay on the boundary of $S$. Mathematically, this means that abundance values $a_{j,k} = 0$; i.e. pixel $k$ does not contain any constituent $j$. In the hyperspectral image area, it is known that a pixel spectrum consists of a mix of at most 4 constituents. As soon as noise is added, one can approximate with probability theory the chance that a pixel lays outside $S$. Let $\rho$ be an estimation of the fraction of pixels we expect to be interior with respect to $S$. An initial matrix $V$ that does not include all pixels is generated. Iteratively the endmembers $\hat{V}$ are estimated from the minimum volume problem by solving (2) and the active pixels at its boundary are removed up to a $\rho$ fraction is left over.

To recover the abundance values from the estimated endmembers $V$ the term *linear spectral unmixing* (LSU) is used when nonnegativeness of estimated abundance is not taken into account. For the fraction of pixels located within simplex $\hat{S}$ we have automatically positive abundance values. For pixels $z_k$ outside $\hat{S}$, we have at least one corresponding $a_{jk} < 0$. The term *fully constrained linear spectral unmixing* (FCLSU) is used if we want to force abundance values to be nonnegative. To do so we consider that the noise of $z_k$ is componentwise independent we choose to project $z_k$ on the facet of $\hat{S}$ closest to $z_k$ and determine the abundance for the endmembers in the plane of that facet.

## 5. Computer simulated data experiments and conclusions

Computer simulations have been carried out in order to evaluate the accuracy of MINVEST in comparison with N-FINDR in highly controlled analysis scenarios. The quality of estimation $\hat{V}$ $(\hat{A})$ of $V$ $(A)$ is measured as the standard deviation estimate assuming $\hat{V}$ $(\hat{A})$ is unbiased, also called root mean squared error (RMSE). To distinguish, we will use $\sigma_A$ if $\hat{A}$ is generated by LSU and $\sigma_{Ap}$ if $\hat{A}$ is generated by FCLSU We show the results obtained from a case with $n = 5$

endmembers and $r = 500$ pixels. To mimic the idea of combinations of a few constituents, a ground truth abundance matrix $A$ is generated consisting for $50\%$ of mixtures of 2 endmembers and for $50\%$ of mixtures of 3 endmembers. They are generated uniformly over the unit simplex. The score matrix $Z$ as input for the estimation is taken as $Z = VA + \sigma \cdot \xi$, where $\xi$ is standard white noise. The choice of the parameter value for $\rho$ is determined by the data $\rho = 18.25\%$. Given that performance indicators depend on pseudo-randomly drawn white noise, for each ground-truth matrix $A$ we replicated white noise 100 times.

*Table 1.* (RMSE) of endmembers $V$ and abundance $A$ obtained by N-FINDR and MINVEST given noise $\sigma$.

| | N-FINDR | | | | | MINVEST | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $\sigma$ | 0.01 | 0.1 | 0.2 | 0.5 | 0.7 | 0.01 | 0.1 | 0.2 | 0.5 | 0.7 |
| $\sigma_V$ | .030 | .118 | .233 | .857 | 1.359 | .013 | .111 | .194 | .486 | .922 |
| $\sigma_A$ | .011 | .063 | .114 | .259 | .323 | .007 | .058 | .105 | .204 | .266 |
| $\sigma_{Ap}$ | .008 | .048 | .092 | .224 | .281 | .005 | .048 | .086 | .174 | .234 |

The measured performance for N-FINDR and MINVEST is given in Table 1. It shows standard deviation estimates $\sigma_V$ of endmembers and $\sigma_A$ of fractional abundances calculated via LSU and via fully constrained spectral unmixing (FCLSU). One can observe that the standard deviation of the estimates is in the same order of magnitude as that of noise. This means that the procedures give results as accurate as the input data. Deviation of endmembers and abundances estimations provided by N-FINDR are higher than those obtained with MINVEST. Other scenarios with and without pure pixels have been generated and evaluated.

The following can be concluded: (1) The problem of unmixing hyperspectral data may be a hard to solve problem. (2) The minimum volume simplicial enclosure problem is a Global Optimization problem where the number of optima depends in worst case on the number of points in the convex hull of the instance. (3) The resulting simplex of the (combinatorial) maximum volume simplex problem is enclosed in the result of the minimum volume enclosing simplex problem. (4) Local search from a good starting simplex leads in general to the global optimum for the case of spectral unmixing due to well spread data in the originating simplex and low noise in practice. (5) The new MINVEST algorithm does not require pure pixels to be present in the scene of the instance unlike the N-FINDR algorithm. (6) In the case of having no noise and well spread data over the boundary of the spectral simplex, MINVEST recovers the original endmembers and ground truth abundance. (7) The RMSE performance indicator is sensitive to scaling in its use for measuring abundance discrepancies. (8) The results of MINVEST seems more correlated to ground truth abundance data than the ones of N-FINDR.

## References

[1] T.H. Chan, C.Y. Chi, Y.M. Huang, and W.K. Ma. Convex analysis based minimum-volume enclosing simplex algorithm for hyperspectral unmixing. In *Proceedings of the 2009 IEEE international conference on Acoustics, Speech and Signal Processing*, pages 1089–1092. IEEE, Piscataway, 2009.

[2] E. M.T. Hendrix and B.G. Toth. *Introduction to Nonlinear and Global Optimization*. Springer, Cambridge, 2010.

[3] N Keshava. A survey of spectral unmixing algorithms. *Lincoln Laboratory Journal*, 14:55–78, 2003.

[4] L. Miao and H. Qi. Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization. *IEEE Transactions on Geoscience and Remote Sensing*, 45:765–777, 2007.

[5] A. Plaza, P. Martinez, R. Perez, and J. Plaza. A quantitative and comparative analysis of endmember extraction algorithms from hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 42(3):650–663, 2004.

[6] M. E. Winter. N-FINDR: an algorithm for fast autonomous spectral end-member determination in hyperspectral data. *Proc. SPIE Image Spectrometry V*, 3753:266–277, 2003.

Proceedings of TOGO 2010, pp. 59 – 62.

# On Solving the Maximum Weighted Clique Problem via d.c. Constraint Problem

Tatyana V. Gruzdeva

*Institute for System Dynamics and Control Theory SB of RAS, Lermontov Str., 134, Irkutsk, 664033, Russia*
gruzdeva@icc.ru

**Abstract** This report is focused on computational solution approach for the Maximum Weighted Clique Problem (MWCP). The problem is formulated as continuous optimization problem with the nonconvex quadratic constraint given by difference of two convex functions (d.c. function). The approach is based on Global Optimality Conditions. The key ingredients of the approach are: local search algorithm, linearization at points of finite approximation of level surface of convex function. The effectiveness of the proposed algorith is illustrated by the computational results.

**Keywords:** global optimization, d.c. constraint, maximum weighted clique problem

## 1. Introduction

Let $G = (V, E, w)$ be a simple undirected weighted graph with the nodes set $V$ and the set of edges $E$. The node weights are given by the vector $w \in \Re^{|V|}$. Let $\overline{G} = (V, \overline{E}, w)$ be the complementary graph of $G$. We assume that the graph and its complement have no isolated nodes. A clique $C$ is a subset of $V$ such that the subgraph $G(C)$ is complete. The Weighted Maximum Clique Problem (MWCP) consists in a finding a clique of maximum weight. The particular case of MWCP is the Maximum Clique Problem (MCP), where $w_i = 1,\ \ i = 1, \dots, |V|$, i.e. the clique of maximum cardinality have to be found.

In this report we propose to consider the continuous formulation of MWCP as an optimization problem with d.c. constraint over the canonical simplex. For this formulation a Global Search Strategy (GSS) [1, 2] can be adopted. We show how the efficient implementation of GSS can be done taking into account the structure of WMCP. The efficiency of this application of GSS to the MWCP is demonstrated by a numerical experiment on DIMACS benchmarks and a comparison with algorithms from [3].

## 2. Continuous formulation of MWCP

Let $C \subset V$ be a clique, $W(C) = \sum\limits_{i \in C} w_i$ be the total weight of a clique $C$ and $|V| = n$.

The characteristic vector $z(C, w)$ of a clique $C$ is given by

$$z(C,w)_i = \begin{cases} \dfrac{w_i}{W(C)}, & \text{if } \ i \in C, \\ \quad 0, & \text{otherwise.} \end{cases} \tag{1}$$

We construct the matrix $\overline{B} = ||\overline{b_{ij}}||_{(n \times n)}$ by following rule [4]:

$$\overline{b_{ij}} = \begin{cases} \dfrac{1}{2w_i} + \dfrac{1}{2w_j}, & \text{if } i \neq j, \quad (i,j) \notin E; \\ \quad 0, & \text{otherwise.} \end{cases} \tag{2}$$

If $w_i = 1$, for all $i = 1, \dots, n$ then matrix $\overline{B}$ turns out to be the adjacency matrix of the complementary graph $\overline{G}$.

We propose a new formualtion of MWCP as a continuous nonconvex problem, as follows:

$$\phi(x) \overset{\triangle}{=} \sum_{i=1}^{n} \frac{1}{w_i} x_i^2 \downarrow \min, \;\; x \in S, \;\; \Phi_0(x) \overset{\triangle}{=} \langle x, \overline{B}x \rangle \leq 0, \qquad (\mathcal{P}_0)$$

where $S \overset{\triangle}{=} \{x \in I\!R^n \;\mid\; x \geq 0, \; \sum\limits_{i=1}^{n} x_i = 1\}$.

Denote $supp(x) = \{i \in \{1, \dots, n\} \;\mid\; x_i > 0\}$. It is not so difficult to prove

**Lemma 1.** *[5] Suppose $x \in S$. Then $\langle x, \overline{B}x \rangle = 0$ iff $supp(x)$ is a clique in the graph $G$.*

Consider now the following function of parameters $\alpha$ and $\gamma$:

$$\Phi(x) \overset{\triangle}{=} \langle x, [\alpha\overline{B} + \gamma(B + D)]x \rangle - \gamma \langle d, x \rangle, \qquad (3)$$

where $d \in I\!R^n, \;\; d_i = \dfrac{1}{w_i}, \; i = 1, \dots, n$, $D = diag\{d_1, \dots, d_n\}$ and the matrix $B = ||b_{ij}||_{(n \times n)}$ is defined by the rule:

$$b_{ij} = \begin{cases} \dfrac{1}{2w_i} + \dfrac{1}{2w_j}, & \text{if } (i,j) \in E; \\ 0, & \text{otherwise.} \end{cases} \qquad (4)$$

**Lemma 2.** *[5] Suppose $x \in S$, $\alpha \neq \gamma$. Then the set $supp(x)$ is a clique of graph $G$ iff $\Phi(x) = 0$.*

Then we turn our attention to the question on the inequality with the function $\Phi(x)$. If $\langle x, \overline{A}x \rangle \leq 0$, then it can be readely seen, that $\Phi(x) \leq 0$, when $\alpha > \gamma$, and $\Phi(x) \geq 0$ when $\alpha < \gamma$. So, $sign(\alpha - \gamma)\Phi(x) \leq 0$ when $\alpha \neq \gamma$.

Let us consider the generalization of $(\mathcal{P}_0)$ with parameters $\alpha \neq \gamma$:

$$\phi(x) \downarrow \min, \;\; x \in S, \;\; sign(\alpha - \gamma)\Phi(x) \leq 0. \qquad (\mathcal{P})$$

Note that Problem $(\mathcal{P}_0)$ is a particular case of Problem $(\mathcal{P})$ when $\gamma \equiv 0$. If $\alpha \equiv 0$, we have the continuous formulation of MWCP with matrix $B$, corresponding to graph $G$, as follows

$$\sum_{i=1}^{n} \frac{1}{w_i} x_i^2 \downarrow \min, \;\; x \in S, \;\; \langle x, (B + D)x \rangle - \langle d, x \rangle \geq 0. \qquad (\mathcal{P}_1)$$

**Theorem 3.** *Let Problem $(\mathcal{P})$ corresponds to graph $G(V, E, w)$ and $\alpha \neq \gamma$. Then the following statements hold:*

*i) $C$ is a maximal weighted clique of $G$ and $z(C, w)$ is its characteristic vector;*

*ii) $z$ is a strict local minimizer of $(\mathcal{P})$;*

*iii) $z$ is a local minimizer of $(\mathcal{P})$, $\;z \in Arglocmin(\mathcal{P})$.*

## 3. Local search

In this section we present a Local Search Algorithm (LSA) to find a maximal clique $C$, which we call $\overline{C}$-procedure. Here we are concerned the idea of $C$-procedure from [6].

For the sake of clarity suppose, that $\alpha > \gamma \geq 0$. $\overline{C}$-procedure consists of two parts.

1) The first one begins at some point $x^0 \in S$ and generates the sequence $\{x^m\}$ of infeasible points $x^m$: $\;\Phi(x^{m+1}) < \Phi(x^m), \;\; m = 0, 1, 2, \dots$ The work of the first part is terminated when the set $supp(x^m)$ is a clique, i.e. $\Phi(x^m) = 0$.

2) In the second part one constructs a maximal clique $C$ containing $supp(x)$, where $x$ is the final point of the first part.

$\overline{C}$-procedure

**Step 0.** Put $m := 0$, choose $x^0 \in S$.

**Step 1.** Construct $supp(x^m)$.

**Step 2.** If $supp(x^m)$ is a clique then $x := x^m$ and go to step 5.

**Step 3.** Find vertex $q$ and $p$ from $supp(x^m)$ such that $(q,p) \notin E, \ \ \Phi(\overline{x}(q,p)) < \Phi(x^m)$, where $\overline{x}(q,p) = x^m + x_p^m(e^q - e^p)$.

**Step 4.** Put $x^{m+1} := \overline{x}(q,p), \ \ m := m+1$ and go to step 1.

**Step 5.** Find a maximal weighted clique $C \supset supp(x^m)$. Put $K := |C|$.

**Step 6.** Construct $z(C,w)$ and Stop.

The choice of pair $(q,p)$ and constructing of maximal clique $C$ may be implement by different methods, for example by methods from [6]. So we could obtain different variants of $\overline{C}$-procedure.

## 4. Global search

It can be easily seen that problems $(\mathcal{P}_0), \ (\mathcal{P})$ and $(\mathcal{P}_1)$ are not convex optimization problems since matrices $B$ and $\overline{B}$ are not nonnegative definite. But it is well-known that an indefinite matrix can be represented as a difference of two nonnegative definite matrices.

As a consequence, problems $(\mathcal{P}_0), \ (\mathcal{P})$ and $(\mathcal{P}_1)$ turns out to be a particular cases of problem with d.c. constraint [1, 2]:

$$f(x) \downarrow \min, \ \ x \in S, \ \ h(x) - g(x) \leq 0, \tag{5}$$

where $f(\cdot), \ g(\cdot), \ h(\cdot)$ are convex functions and $S$ is a convex set.

There are many ways to represent a matrix as a difference of two nonnegative definite matrices. For example, when $\alpha > \gamma > 0$ the following representation of matrix $A = \alpha\overline{B} + \gamma(B+D)$ in problem $(\mathcal{P})$ may be used: $A = A_1 - A_2, \ \ \ A_1 = \alpha\overline{\Lambda} + \gamma(\Lambda + D), \ \ \ A_2 = \alpha(\overline{\Lambda} - \overline{B}) + \gamma(\Lambda - B)$, where $\Lambda = diag\{\lambda_1, \dots, \lambda_n\}, \ \ \lambda_i = \sum\limits_{j=1}^{n} b_{ij}; \ \ \ \overline{\Lambda} = diag\{\overline{\lambda_1}, \dots, \overline{\lambda_n}\}, \ \ \overline{\lambda_i} = \sum\limits_{j=1}^{n} \overline{b_{ij}}$.

It is obvious, that $\lambda_i, \overline{\lambda_i} > 0, \ i = 1, \dots, n$, since graph $G$ and its complement $\overline{G}$ have no isolated nodes. Hence, the matrix $A_1$ is positive definite and matrix $A_2$ has the dominant diagonal. Then functions $h(x) = \langle x, A_1 x\rangle - \gamma\langle d, x\rangle, \ \ g(x) = \langle x, A_2 x\rangle$ are convex functions. So we obtained the d.c. decomposition of function $\Phi(\cdot)$ in problem $(\mathcal{P})$.

Now let us describe a Global Search Algorithm for solving MWCP. Let a point $x^0 \in S$ be given.

$\Re$-strategy

**Step 0.** Put $k := 0$. Starting from $x^0$ obtain by $\overline{C}$-procedure a point $z^0$ of local minimum to $(\mathcal{P})$.

**Step 1.** Put $\beta_k := h(z^k), \ \zeta_k := \phi(z^k)$. Construct an approximation

$$\mathcal{R}_k = \{y^i = \lambda_i e^i \mid g(y^i) = \beta_k, \ i = 1, \dots, n\}.$$

**Step 2.** For every $i = 1, \dots, n$ solve the Linearized Problem

$$h(x) - \langle\nabla g(y^i), x\rangle \downarrow \min, \ \ x \in S, \ \ \phi(x) \leq \zeta_k. \tag{$\mathcal{PL}_i$}$$

Let $u^i$ be a solution of the Linearized Problem.

**Step 3.** Starting from $u^i \in S$ $(i = 1, \dots n)$ obtain by $\overline{C}$-procedure a point $v^i \in Arglocmin(\mathcal{P})$.

**Step 4.** From all points $v^i, \ \ i = 1, \dots n$, choose a point $v^j : \ \ \ \phi(v^j) = \min\limits_{1 \leq i \leq n} \phi(v^i)$.

**Step 5.** If $\phi(v^j) < \phi(z^k)$, then put $z^{k+1} := v^j, \quad k := k+1$ and go to step 1. Otherwise put $z := z^k$, STOP: $supp(z)$ is a maximal clique.

To illustrate the efficiency of the proposed algorithm, computational experiments were carried out on DIMACS benchmark graphs ("C"-graph data, for instance, see in Table 1). The weights of the nodes are all in the range $[1, 10]$ and are generated following way [3]. The nodes $i$ of graph $G(V, E, w)$ are numbered from 1 to $n$ and $w_i = i \ mod \ 10, \quad i = 1, \ldots, n$. In Table 1 ”$W$” is the weight of the solutions for each algorithm, ”$T$”(sec.) is the computation time and ”$W_*$” is the maximum weight (optimal) of the clique computed by Xpress-MP. It can be readily seen from Table 1 that $\mathcal{R}$–algorithm has reached better solutions on the instances, where $n \geq 1000$.

*Table 1.* Testing on DIMACS benchmark "C" graphs

| $graph$ | $W_*$ | $MS-alg$ | | $\mathcal{R}-alg$ | |
|---|---|---|---|---|---|
| | | $W$ | $T$ | $W$ | $T$ |
| C125.9 | **215** | **215** | 5 | **215** | 3 |
| C250.9 | **304** | **304** | 13 | 292 | 7 |
| C500.9 | $\geq$**385** | **385** | 3364 | 379 | 48 |
| C1000.9 | $\geq$**483** | 470 | 554 | **483** | 375 |
| C2000.9 | $\geq$**561** | 531 | 2431 | **561** | 1007 |
| C2000.5 | $\geq$**126** | 113 | 324 | **126** | 605 |

## 5. Conclusion

In the paper we considered the well-known combinatorial problem of finding a maximum weighted clique (MWCP) as the continuous problem with nonconvex quadratic constraint given by difference of two convex functions.

For solving MWCP we applied an approach based on Global Optimality Conditions for problems with d.c. constraint. Developing the proposed Global Search Strategy for problems with d.c. constraint, we obtained $\mathcal{R}$–algorithm.

The extensive computational experiments were carried out on the DIMACS benchmark graphs. The obtained computational results stimulate the future investigations.

**Acknowledgments.** The author wish to thank Prof. Alexander Strekalovsky for their encouragement and support.

## References

[1] Strekalovsky, A.S. (2005). *Minimizing sequences in problems with d.c. constraints.* Comput. Math. Math. Phys., V.45, pages 418–429.

[2] Gruzdeva, T.V., Strekalovsky, A.S. (2007). *Local search in problems with nonconvex constraints.* Comput. Math. Math. Phys., V. 47, pages 381–396.

[3] Mannino, C., Stefanutti, E. (1999). *An augmentation algorithm for the maximum weighted stable set problem.* Comput. Optimization and Applications, V. 14, pages 467–371.

[4] Bomze, I.M., Budinich, M., Pardalos, P.M., Pelillo, M. (1999). *The maximum clique problem.* Handbook of Combinatorial Optimization, Suppl. V. A, Kluwer, pages 1–74.

[5] Gruzdeva, T.V. (2008). *On solving the clique problems via d.c. constraints problems.* Discrit analisys and oper. research, V. 15, pages 20-34 (in russian).

[6] Kuznetsova, A., Strekalovsky, A.S. (2001). *On solving the maximum clique problem.* J. Global Optim., N 3, pages 265–288.

# Finding largest small polygons with GloptiPoly

Didier Henrion[1,2] and Frédéric Messine[3]

[1] *LAAS-CNRS, University of Toulouse, France,* henrion@laas.fr

[2] *Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic,* henrion@fel.cvut.cz

[3] *ENSEEIHT, IRIT-CNRS, University of Toulouse, France,* frederic.messine@n7.fr

**Abstract** A small polygon is a convex polygon of unit diameter. We are interested in small polygons which have the largest area for a given number of vertices $n$. Many instances are already solved in the literature, namely for all odd $n$, and for $n = 4, 6$ and $8$. Thus, for even $n \geq 10$, instances of this problem remain open. Finding those largest small polygons can be formulated as nonconvex quadratic programming problems which can challenge state-of-the-art global optimization algorithms. We show that a recently developed technique for global polynomial optimization, based on a semidefinite programming approach to the generalized problem of moments and implemented in the public-domain Matlab package GloptiPoly, can successfully find largest small polygons for $n = 10$ and $n = 12$. Therefore this significantly improves existing results in the domain. When coupled with accurate convex conic solvers, GloptiPoly can provide numerical guarantees of global optimality, as well as rigorous guarantees relying on interval arithmetic.

**Keywords:** extremal convex polygons, global optimization, nonconvex quadratic programming, semidefinite programming

## 1. Introduction

The problem of finding the largest small polygons was first studied by Reinhardt in 1922 [13]. Reinhardt solved the problem by proving that the solution corresponds to the regular polygons but only when the number of vertices $n$ is odd. He also solved the case $n = 4$ by proving that a square with diagonal length equal to 1 is a solution. However, it exists an infinity of other different solutions (it is just necessary that the two diagonals intersect with a right angle). The hexagonal case $n = 6$ was solved numerically by Graham in 1975 [6]. Indeed, Graham studied possible structures that the optimal solution must have. He introduced the diameter graph of a polygon which is defined by the vertices of the polygon and by edges with length one (if and only if the corresponding two vertices of the edge are at distance one). Using a result due to Woodall [14], he proved that the diameter graph of the largest small polygons must be connected, yielding 10 distinct possible configurations for $n = 6$. Discarding 9 of these 10 possibilities by using standard geometrical reasonings plus the fact that all the candidates must have an area greater than the regular small hexagon, he determined the only possible diameter graph configuration which can provide a better solution. He solved this last case numerically, yielding the largest small hexagon. Following the same principle, Audet et al. in 2002 found the *largest small octagon* [4]. The case $n = 8$ is much more complicated than the case $n = 6$ because it generates 31 possible configurations and just a few of them can be easily discarded by geometrical reasonings. Furthermore, for the remaining cases, Audet et al. had to solve difficult global optimization problems with 10 variables and about 20 constraints. These problems are formulated as quadratic programs with quadratic constraints [4]. Audet et al. used for that a global solver named QP [1]. Notice that optimal solutions for $n = 6$ and $n = 8$ are not the regular polygons [4, 6]. In 1975, Graham proposed a conjecture which is the following: when $n$ is even and $n \geq 4$, the largest small polygon must have a diameter graph with a cycle with $n-1$ vertices and with an additional edge attached to a vertex of the cycle; this is true for $n = 4, 6$ and also $n = 8$, see Figure 1. Therefore, this yields only one possible diameter graph configuration that must have the optimal solution. In 2007, Foster and Szabo proved Graham's conjecture [5]. Thus to solve the following open cases $n \geq 10$, it is just necessary to solve one global optimization problem defined by the configuration of the diameter graph with a cycle with $n-1$ vertices and an additional pending edge. In order to have an overview of these subjects, refer to [2, 3].

## 2. Nonconvex quadratic programming

As mentioned above, for an even $n \geq 4$, finding the largest small polygon with $n$ vertices amounts to solving only one global optimization problem. All these problems depending on $n$ can be formulated as nonconvex quadratic programs under quadratic constraints [4]. For illustration, here is the problem corresponding to the case $n = 8$ (the definitions of the variables are given in Figure 1):

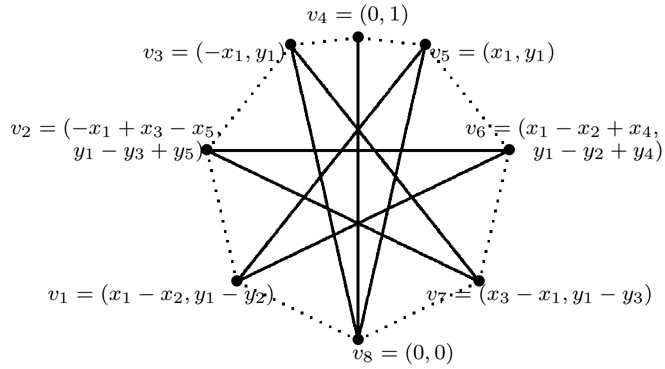

$v_4 = (0,1)$
$v_3 = (-x_1, y_1)$
$v_5 = (x_1, y_1)$
$v_2 = (-x_1 + x_3 - x_5,$
$y_1 - y_3 + y_5)$
$v_6 = (x_1 - x_2 + x_4,$
$y_1 - y_2 + y_4)$
$v_1 = (x_1 - x_2, y_1 - y_2)$
$v_7 = (x_3 - x_1, y_1 - y_3)$
$v_8 = (0,0)$

*Figure 1.* Case of $n = 8$ vertices. Definition of variables following Graham's conjecture.

$$\left\{\begin{array}{lll} \max\limits_{x,y} & \frac{1}{2}\{(x_2 + x_3 - 4x_1)y_1 + (3x_1 - 2x_3 + x_5)y_2 + (3x_1 - 2x_2 + x_4)y_3 & \\ & \qquad\qquad +(x_3 - 2x_1)y_4 + (x_2 - 2x_1)y_5\} + x_1 & \\ \text{s.t.} & \|v_i - v_j\| \leq 1, \forall (i,j) \in \{1, \cdots, 8\}, i \neq j & \\ & \|v_2 - v_6\| = 1 & \\ & x_i^2 + y_i^2 = 1 & i = 1,2,3,4,5 \\ & x_2 - x_3 \geq 0 & y \geq 0 \\ & 0 \leq x_1 \leq 0.5 & 0 \leq x_i \leq 1, \quad i = 2,3,4,5. \end{array}\right. \qquad (1)$$

Without loss of generality we can insert the additional constraint $x_2 \geq x_3$ which eliminates a symmetry axis. In program (1), all the constraints are quadratic. The quadratic objective function corresponds to the computation of the area of the octagon following Graham's diameter graph configuration. This formulation is easy to extend to the cases $n \geq 10$ with $n$ even.

## 3. GloptiPoly

In 2000, Lasserre proposed to reformulate nonconvex polynomial optimization problems (POPs) as linear moment problems, in turn formulated as linear semidefinite programming (SDP) problems [10]. Using results on flat extensions of moment matrices and representations of polynomials positive on semialgebraic sets, it was shown that under some relatively mild assumptions, solving nonconvex POPs amounts to solving a sufficiently large linear hence convex SDP problem. In practice, a hierarchy of embedded SDP relaxations of increasing size are solved gradually. Convergence and hence global optimality can be guaranteed by examining a certain rank pattern in the moment matrix, a simple task of numerical linear algebra. A user-friendly Matlab interface called GloptiPoly was designed in 2002 to transform a given POP into an SDP relaxation of given size in the hierarchy, and then to call SeDuMi, a general-purpose conic solver [7]. A new version 3 was released in 2007 to address generalized problem of moments, including POPs but also many other decision problems. The interface was also extended to other public-domain conic solvers [8]. Almost a decade after the initial spark [10], Lasserre summarized the theoretical and practical sides of the approach in a monograph [11].

## 4. Numerical experiments

We applied GloptiPoly 3 and SeDuMi 1.1R3 to solve the quadratic problem in the cases $n = 8$ and 10. In order to obtain accurate solutions, we let SeDuMi minimize the duality gap as much as possible. We also tightened the tolerance parameters used by GloptiPoly to detect global optimality and extract globally optimal solutions. We used a 32 bit desktop personal computer with a standard configuration.

For the case $n = 8$ we obtain the solution (with 8 significant digits) $x_1 = 0.26214172$, $x_2 = 0.67123417$, $x_3 = 0.67123381$, $x_4 = 0.90909242$, $x_5 = 0.90909213$ whose global optimality is guaranteed numerically (the moment matrix has approximately rank one) at the second SDP relaxation in the hierarchy. This SDP problem is solved by SeDuMi in less than 5 seconds. The objective function of the SDP relaxation is equal to $0.72686849$, and this is an upper bound on the exact global optimum. The quadratic objective function evaluated at the above solution is the same to 11 significant digits. Symmetry considerations indicate that $x_2 = x_3$ and $x_4 = x_5$ at the optimum, and we see that the above solution achieves this to 5 digits for $x_2$ and to 6 digits for $x_4$.

These results can be rigorously guaranteed by using Jansson's VSDP package which uses SDP jointly with interval arithmetic [9]. The solution of an SDP problem can be guaranteed at the price of solving a certain number of SDP problems of the same size. In our case, VSDP solved 8 instances of the second SDP relaxation to provide the guaranteed lower bound $0.72686845$ and guaranteed upper bound $0.72686849$ on the objective function, namely the area of the octagon.

In the case $n = 10$, we obtain the solution $x_1 = 0.21101191$, $x_2 = 0.54864468$, $x_3 = 0.54864311$, $x_4 = 0.78292524$, $x_5 = 0.78292347$, $x_6 = 0.94529290$, $x_7 = 0.94529183$ whose global optimality is guaranteed numerically at the second SDP relaxation. This SDP problem is solved by SeDuMi in less than 5 minutes. The objective function of the SDP relaxation, an upper bound on the exact global optimum, is equal to $0.74913736$. The quadratic objective function evaluated at the above solution is the same to 10 significant digits.

For $n = 12$, we obtain the following solution without using the rigorous method of SDP: $x_1 = 0.17616131$, $x_2 = 0.46150224$, $x_3 = 0.46150519$, $x_4 = 0.67623091$, $x_5 = 0.67623301$, $x_6 = 0.85320300$, $x_7 = 0.85320328$, $x_8 = 0.96231370$, $x_9 = 0.96231344$. This SDP problem is solved within 1h06. The objective function of the SDP relaxation, an upper bound on the exact global optimum, is equal to $0.76072988$. The solutions for the optimal decagon and dodecagon are drawn in Figure 2.

## 5. Conclusion

GloptiPoly can be efficiently used to find some largest small polygons with an even number of vertices. The case $n = 8$ is most efficiently solved than in [4]: (i) the accuracy on the value of the area is now $10^{-10}$ in place of $10^{-5}$ and (ii) the required CPU time is about 5 seconds in

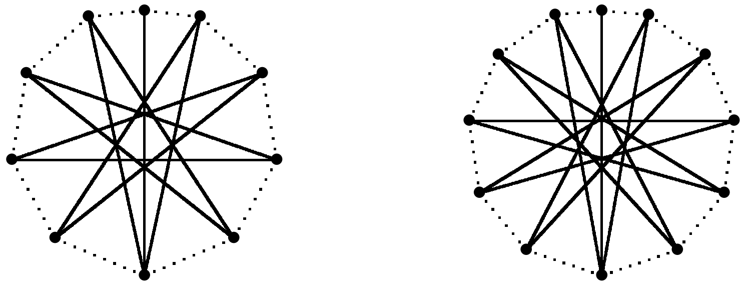


*Figure 2.* Largest Small Decagon and Dodecagon.

place of 100 hours. Furthermore, the next open instance for $n = 10$ is solved using GloptiPoly in only 5 minutes with always an accuracy of $10^{-10}$. These two results are obtained with a certified guarantee on 10 digits. For the case $n = 12$, GloptiPoly found the global solution, but for the moment without a certified guarantee. In future works, we have to certified and guarantee the solution obtained for the case $n = 12$. It seems to be also possible to solve the next open case $n = 14$. Note that all the found largest small polygons with an even number $n$ of vertices (from $n = 4$ to 12) own a symmetry axis on the pending edge of their corresponding optimal diameter graph configurations.

# References

[1] C. Audet, P. Hansen, B. Jaumard, G. Savard, *A branch and cut algorithm for nonconvex quadratically constrained quadratic programming*, Mathematical Programming, Vol. 87, No. 1, Series A, pp. 131–152, 2000.

[2] C. Audet, P. Hansen, F. Messine, *Extremal problems for convex polygons*, Journal of Global Optimization, Vol. 38, pp. 163–179, 2007.

[3] C. Audet, P. Hansen, F. Messine, *Extremal problems for convex polygons - an update*, In **Lectures on Global Optimization** (P. M. Pardalos, T. F. Coleman, Eds.), **Fields Institute Communications**, AMS, pp. 1–16, 2009.

[4] C. Audet, P. Hansen, F. Messine, J. Xiong, *The largest small octagon*, Journal of Combinatorial Theory, Series A, Vol. 98, No. 1, pp. 46–59, 2002.

[5] J. Foster, T. Szabo, *Diameter graphs of polygons and the proof of a conjecture of Graham*, Journal of Combinatorial Theory, Series A, Vol. 114, No. 8, pp. 1515–1525, 2007.

[6] R. L. Graham, *The largest small hexagon*, Journal of Combinatorial Theory, Series A, Vol. 18, pp. 165–170, 1975.

[7] D. Henrion, J. B. Lasserre, *GloptiPoly: global optimization over polynomials with Matlab and SeDuMi*. ACM Trans. Math. Software, Vol. 29, No. 2, pp. 165–194, 2003.

[8] D. Henrion, J. B. Lasserre, J. Löfberg. *GloptiPoly 3: moments, optimization and semidefinite programming*. Optim. Methods and Software, Vol. 24, No. 4-5, pp. 761–779, 2009.

[9] C. Jansson, *VSDP: a Matlab software package for Verified Semidefinite Programming*. Nonlinear Theory and its Applications, pp. 327–330, 2006.

[10] J. B. Lasserre, *Optimisation globale et théorie des moments*. Comptes Rendus de l'Académie des Sciences, Paris, Vol. 331, Série 1, pp. 929-934, 2000.

[11] J. B. Lasserre, *Moments, Positive Polynomials and Their Applications*, Imperial College Press, London, UK, 2009

[12] M.J. Mossinghoff, *Isodiametric Problems for Polygons*, Discrete and Computational Geometry, Vol. 36, pp. 363–379, 2006.

[13] K. Reinhardt, *Extremale Polygone gegebenen Durchmessers*, Jahresbericht der deutschen Mathematiker Vereinigung, Vol. 31, pp. 251–270, 1922.

[14] D.R. Woodall, *Thrackles and Deadlock*, In *Combinatorial Mathematics and Its Applications* (D. J. A. Welsh, Ed.), Academic Press, New York, 1971.

# Extensions of Interval Branch-and-Bound Algorithms for Mixed Global Optimization Problems with Real and Categorical Variables

Bernard Jeannet[1] and Frédéric Messine[2]

[1] *LIUPPA - EA 3000, University of Pau (UPPA), Anglet, France,* bernard.jeannet@univ-pau.fr

[2] *ENSEEIHT, IRIT-CNRS, University of Toulouse, France,* frederic.messine@n7.fr

**Abstract** Real global optimization problems, such as the design of electromechanical actuators, often imply to take into account different kinds of variables: discrete and continuous. In this paper, we study mixed problems which combine real and categorical variables (discrete variables without ordering). Four methods are presented and discussed in order to compute bounds for the categorical variables. This yields some properties and permits some extensions of classical interval branch-and-bound global optimization algorithms. Numerical tests will validate our approaches on two examples.

**Keywords:** Interval Arithmetic, Branch and Bound, mixed variables, categorical variable.

## 1. Introduction

Interval analysis introduced by Moore has already been successfully used in many branch-and-bound algorithms for global optimization of univariate or multivariate non-linear and non-convex analytical functions, possibly subject to constraints [1, 3–6]. These methods provide precise enclosures for the optimal value and for one or all optimal solutions with an absolute guarantee, i.e. the errors can be bounded by an arbitrary degree specified by the user.

In this paper, constrained global optimization problems with mixed (real and categorical) variables are addressed. Let us name $x$ the vector of real variables where $x \in X \subset I\!R^n$, and $k$ the vector of categorical variables where $k \in \Pi_{j=1}^{m} I\!K_j$. $I\!K_j$ are enumerated sets which permit to list some possibilities, for example the kind of magnet for electrical machines [5]. These sets are discrete but without any ordering. Obviously, the choice of a category changes some physical parameters which are taken into account in the associated model. A real function $c$ is then introduced in order to manipulate these categorical variables; $c : \prod_{j=1}^{m} I\!K_j \longrightarrow I\!R^t$.

The optimization problems studied in this paper, are formulated as follows:

$$\left\{\begin{array}{ll} \displaystyle\min_{x\in X\subset I\!R^n, k\in\Pi_{j=1}^{m} I\!K_j} & f(x, c(k)) \\ u.c. & g_i(x, c(k)) \leq 0, \forall i \in \{1, \ldots, p\}, \\ & h_j(x, c(k)) = 0, \forall j \in \{1, \ldots, q\}, \end{array}\right. \tag{1}$$

where $f$, $g_i$ and $h_j$ are real functions.

In all the real examples that we had to solve [5], all the models of design always use functions $c_i$ depending on only one categorical variable. This is the reason why, for simplifying this paper, only $t$ univariate functions are considered: $c_i : I\!K_j \longrightarrow I\!R$, where the category index $j$ belongs to $\{1, \cdots, m\}$ and $i$ belongs to $\{1, \cdots, t\}$. Nevertheless, into the general case, the functions $c_i$ depend on some different categorical variables and it is possible and relatively easy to extend our results.

## 2. Extension of Interval Branch-and-Bound Algorithm

The exact method developed for solving problems (1) is an extension of Interval Branch-and-Bound algorithms, [1, 3, 5, 6], named *IBBA*. All these algorithms are based on *interval analysis*, which is the tool for computing the bounds of a continuous function over a box; a box is in fact an interval vector. Generally, these algorithms work with homogeneous real variables according to an exclusion principle: when a constraint cannot be satisfied in a considered box or when it is proved that the global optimum cannot occur in. In this work, it is necessary to extend these *interval Branch-and-Bound algorithms* in order to deal with *mixed real and categorical* variables.

Representation of the variables: (i) for real variables: one considers the interval compact set where the global solution is searched and (ii) for categorical variables: categories are represented by enumerate sets. Therefore, a distinction must be introduced between continuous and discrete variables.

*IBBA* works according to the four following phases: (i) Bisection Rules: we bisect a real or a categorical set into two parts following a simple heuristic principle based on a weight factor depending on the sizes of the intervals or of the enumerated sets ; (ii) Computation of the Bounds: this represents one of the fundamental parts of a Branch-and-Bound algorithm, because all the techniques of exclusion and of propagation are depending on it. In the case of the formulation (1), functions $f(x, c(k)), g_i(x, c(k)), h_j(x, c(k))$ must be considered, in place of continuous functions which always appear in such techniques. In Section 3, we discus the way to compute bounds for the intermediate univariate function $c(k)$ over a vector of enumerate sets $K \subset \prod_{j=1}^{m} I\!K_j$. Assuming that these bounds can be calculated and therefore, they can be considered as continuous interval vectors; they are denoted by $C(K)$. Thus interval arithmetic can be applied directly and that yields the bounds; (iii) Exclusion principle: these techniques of exclusion are based on the fact that it is proved that the global optimum cannot occur in a box. By computing bounds, this leads to two main possibilities: (a) a constraint is proved to be unsatisfiable; (b) the lower bound of $f$ is greater than a current solution previously computed (the midpoint test is used, refer to [6]); (iv) Constraint Propagation Techniques: it is based on *interval analysis* and makes it possible to reduce the size of a considered hypercube (interval vector) by using the implicit relations of some variables in the expressions of the constraints, see [1, 4].

## 3. Methods for bounding categorical functions

Now, it is necessary to develop methods for the computation of real bounds $C_i^L$ and $C_i^U$ for each categorical function $c_i$ in order to obtain inclusion functions $C_i\left(K_j\right) = [C_i^L, C_i^U]$, $\forall K_j \subseteq I\!K_j$. The bounds calculated in this section must satisfy the only following constraint: $\forall k \in K_j,\ c_i(k) \in C_i(K_j) = \left[C_i^L, C_i^U\right]$ .

Four methods are developed as follows: (i) Method $M_1$: the exact range of each real function $c_i$, $i \in \{1, \cdots, t\}$ over a subset $K_j \subseteq I\!K_j$ is:

$$C_i(K_j) := \left[\min_{k \in K_j} c_i(k), \max_{k \in K_j} c_i(k)\right]. \tag{2}$$

This is the definition that would allow the inclusion functions to be computed with accurate bounds. But it requires an ascending sort algorithm that penalizes each computation of the *Branch-and-Bound method*. We propose the following method that avoids this systematic enumeration. Only one enumeration is performed during the initialization phase, providing

$C_i(I\!\!K_j)$. Hence, for any subset $K_j \subseteq I\!\!K_j$, we associate the following inclusion function:

$$C_i(K_j) := \begin{cases} [c_i(1), c_i(1)] \text{ if } K_j = \{1\}, \\ \vdots \\ [c_i(|I\!\!K_j|), c_i(|I\!\!K_j|)] \text{ if } K_j = \{|I\!\!K_j|\}, \\ C_i(I\!\!K_j) \text{ in any other case.} \end{cases} \tag{3}$$

(ii) Method $M_2$ : the accurate bounds are computed at each iterations following equation (2); (iii) Method $M_3$ : in the very particular case when the categorical function is independent (is not dependent on the same variables $k_i$) and only depend on a single categorical variable, one can initially sort all the values $c_i(k_j)$, $i \in \{1, \cdots, t\}$, $k_j \in \{1, \cdots, |I\!\!K_j|\}$, by changing the numbering of the corresponding categorical variables. But it is not a general method; (iv) Method $M_4$ : searching for the optimum of the function $f(x, c(k))$, what really matters are the values taken by the categorical function $c$. Thus, it is more judicious to bisect each one of the inclusion functions $C_i(K_j)$ into two intervals (interval of lower $c_i(k_j)$ values and interval of higher values) so that optimization algorithm may explore with more efficiency, values taken by $f$ on each of the disjoint sub-intervals. But as categorical functions don't take values on a continuum, we must not forget that they are enforced, through the choice of discrete categorical variables, to take only established values. Those ideas allow us to give the following equivalent formulation of the optimization problem.

$$\begin{cases} \displaystyle\min_{x \in X \subset I\!\!R^n, k \in \Pi_{j=1}^m I\!\!K_j, y \in Y \subset I\!\!R^t} & f(x, y) \\ & g_i(x, y) \leq 0, \ \forall i \in \{1, \ldots, p\}, \\ & h_j(x, y) = 0, \ \forall j \in \{1, \ldots, q\}, \\ & y_i = c_i(k_j), \ \forall i \in \{1, \ldots, t\}, j \in \{1, \ldots, m\}. \end{cases} \tag{4}$$

Initially without constraint the so-reformulated problem involves new real variables $y_i$ which depend on new equality constraints $y_i = c_i(k_j)$. In fact method $M_4$ can be derived as three distinct possibilities: (a) Method $M_{4v1}$ where bounds $C_i\left(K_j^l\right)$, $l = 1, 2$ are computed using method $M_1$, (3); (b) Method $M_{4v2}$ where bounds $C_i\left(K_j^l\right)$, $l = 1, 2$ are computed using method $M_2$, (2); (c) Method $M_{4v3}$ where a complete constraint propagation technique for $Y_i = C_i(K_j)$ is used, [2, 4]. This fourth method $M_4$ yields to the two following properties:

**Proposition 1.** *Let $K_j$ be a categorical set in which a potential solution is evaluated. Let by definition be $Y_i = C_i(K_j)$ the inclusion function associated to $K_j$ and the categorical function $c_i$. It does not exist a favorable bisection of the categorical set $K_j$ into $K_j^1$ and $K_j^2$ for which the two inclusion functions $C_i\left(K_j^1\right)$ and $C_i\left(K_j^2\right)$ should be more narrow (concerning a bounds point of view) than intervals computed after a bisection of interval $Y_i$ into $Y_i^1$ and $Y_i^2$.*

**Proposition 2.** *When the categorical functions values $c_i(k_j)$, with $k_j \in K_j$ are independently and uniformly distributed, the average percentage of intervals $Y_i^l$, $l = 1, 2$ (resulting from bisection of interval $Y_i := C_i(K_j)$) included in the inclusion functions $C_i(K_j^l)$, $l = 1, 2$ (resulting themselves from a bisection of categorical set $K_j$) is:*

$$I\!\!E = 1 - \frac{|K_j|}{(|K_j| - 1)\sqrt{2}^{|K_j|}}.$$

$I\!\!E$ tends quickly towards $100\%$ when the number of categorical values increases; $|K_j| = 6 \Rightarrow I\!\!E = \frac{17}{20} \simeq 0.85\%$ and $|K_j| = 8 \Rightarrow I\!\!E = \frac{13}{14} \simeq 0.92\%$. In average, it is more interesting to bisect intervals $Y_i$ rather than sets $K_j$. This result allows to state the following remark: *when the categorical functions values $c_i(k_j)$ are independently and uniformly distributed, it is more efficient to solve the problem (4) rather than problem (1) in spite of adding new variables $y_i$ and new constraints* $y_i = c_i(k_j)$. For details on these properties and their proofs refer to [2].

## 4. Numerical tests

Consider the two following functions:
$f_1(x_1, x_2, c(k_1)) = 20c_1(k_1)x_1^2 + 2c_2(k_1)x_1x_2$ and $f_2(x_1, x_2, c(k_1)) = 20\frac{y_1^2}{(1-c_1(k_1))^2} + 2c_2(k_1)x_1x_2$, where $k_1 \in I\!\!K_1$ with $|I\!\!K_1| = 6$, and $x_1 \in [-15, 25]$, $x_2 \in [3, 10]$. Following $k_1$, the two univariate functions $c_1$ and $c_2$ take the values described in the following table:

| $k_1$ | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| $c_1$ | 0.5 | 0.3 | 0.8 | 0.1 | 0.9 | 0.12 |
| $c_2$ | $-0.5$ | 0.6 | 0.1 | 1.5 | $-1$ | 0.8 |

The global minimum for $f_1$ is 112.5 corresponding to the solution $k_1^* = 4$, $x_1^* = -7.5$ and $x_2^* = 10$ for the corresponding solution. In the two following tables of result,the number of iterations and the CPU-time are presented for all the method from $M_0$ (we enumerate all the continuous global optimization problems by enumerating the categorical sets and we solve them iteratively) to $M_{4v3}$.

| $Method$ | $M_0$ | $M_1$ | $M_2$ | $M_{4v1}$ | $M_{4v2}$ | $M_{4v3}$ |
|---|---|---|---|---|---|---|
| Number of iterations | 45799 | 9914 | 8378 | 4210 | 3271 | 3148 |
| CPU-time (PC 2GHz (s)) | 42.24 | 3.2 | 3.31 | 0.71 | 0.45 | 0.27 |

The result for $f_2$ is $9.1$ for the minimum value and $(4, -0.6, 10)$ for its corresponding solution.

| $Method$ | $M_0$ | $M_1$ | $M_2$ | $M_{4v1}$ | $M_{4v2}$ | $M_{4v3}$ |
|---|---|---|---|---|---|---|
| Number of iterations | 77928 | 35019 | 29107 | 15230 | 10794 | 6466 |
| CPU-time (PC 2GHz (s)) | 169 | 43.01 | 35.35 | 2.57 | 1.46 | 0.86 |

In the two above tables, we remark that Method $M_4$ inserted in *IBBA* yields clearly the most efficient computational results. This is due to the fact that introducing new real variables, we impose a kind of order to the categorical sets.

## 5. Conclusion

In this paper, we extend IBBA code to solve mixed categorical-real global optimization problems. Four methods were proposed and the fourth one (introducing additional real variables and constraints) is proved to be the most efficient theoretically and numerically.

## References

[1] E. Hansen, *Global Optimization Using Interval Analysis*, MARCEL DEKKER, Inc. 270 Madison Avenue, New York 10016, 1992.

[2] B. Jeannet, F. Messine, *Interval Branch and Bound Algorithms for Optimization Problems with Real and Categorical Mixed Variables*, internal report of LMA, University of Pau, (available on http://messine.perso.enseeiht.fr/publi.html).

[3] R. B. Kearfott, *Rigorous Global Search: Continuous Problems*, Kluwer Academic Publishers, Dordrecht, Boston, London, 1996.

[4] F. Messine, *Deterministic Global Optimization using Interval Constraint Propagation Techniques*, RAIRO-OR, Vol. 38, N. 4, 2004.

[5] F. Messine, *A Deterministic Global Optimization Algorithm for Design Problems*. To appear as chapter of the book "Essays and Surveys in Global Optimization", Kluwer Academic Publishers, 2005.

[6] H. Ratschek, J. Rokne, *New Computer Methods for Global optimization*, ELLIS HORWOOD LIMITED Market Cross House, Cooper Street, Chichester, West Sussex, PO19 1EB, England, 1988.

# Solution of general problem quadratic optimization

Anatolii Kosolap

*Dnipropetrovsk University, Dnipropetrovsk, Ukraine,* anivkos@ua.fm

**Abstract** Let us consider the general quadratic optimization problem. This multiextreme problem is broken into two classes of complexity by quadratic regularization method: convex and concave optimization problems. In order to solve the concave optimization problem let us use the method of positive orthant displacement with space stretching.

**Keywords:** global optimization, quadratic optimization, semidefinite relaxation, quadratic regularization, concave minimization.

## 1. Introduction

The general quadratic problems include one of the most important domains of nonlinear optimization. Many problems referring to economy, finance, project optimization, planning, computer graphics, management of difficult systems can be transformed to quadratic optimization problems in finite-dimensional space when the objective function and constraints contain the general quadratic functions. Such problems contain the set of local minima and belong to NP-difficult class. The feasible set of such problems can be inconsistent and even discrete.

One of the general approaches to solving the problems referring to this class is semidefinite relaxation [1, 3]. In this case quadratic function $x^TAx$ is represented in the form of $Axx^T$ or $A \bullet X$ where is positively semidefined matrix of unit rank. Such transformation allows reducing the general quadratic problem to the linear semidefinite optimization problem, in which the semidefined matrix is unknown quantity. The semidefinite optimization problem can be effectively solved. However, the semidefinite relaxation is the ap-proximate transformation without the requirement that the rank of required matrix is equal to unit. The set of all semidefined matrixes forms the convex cone. Its extreme beams are semidefined matrixes of unit rank. However, not every boundary beam of the semidefinite cone is extreme. Therefore, if the solution of the transformed semidefinite optimization problem is reached on an extreme beam of semidefined matrix cone, the semidefinite relaxation will be exact.

Other general approaches to solving the general quadratic optimization problem use the schemes of branch and bound methods which provide for construction of a subtask tree by means of splitting the feasible domain into the set of subareas and comparing solutions in each of these subareas. Process of splitting of a subarea is finished if the global minimum point is found on it. It is obvious that such approach can be effective only for problems of small dimension which do not have any practical importance. There are also other approaches using duality [4] or decomposition [2]. However, in these cases only approximate estimations can be received for sure. Therefore, the problem of solving the general quadratic problems effectively is still actual.

## 2. Two complexity classes of general quadratic optimization problems

Let us consider the general quadratic optimization problem

$$\min\{f_0(x)|f_i(x) \leq 0, i = 1, \dots, m, x \in E^n\}, \tag{1}$$

where all functions $f_i(x) = x^T A_i x + b_i^T x + c_i$ are quadratic, $b_i, x$ are vectors of $n$-dimensional Euclidean space, $c_i$ are constants, and all matrixes $A_i$ are symmetric. Let us transform problem (1) to the following

$$\min\{x_{n+1}|f_0(x) + s \leq x_{n+1}, f_i(x) \leq 0, i = 1, \dots, m, x \in E^n\}, \tag{2}$$

where value $s$ is chosen so that $f_0(x)+s > 0$. Further, using replacement $x = Az$ where matrix $A$ of order $(n+1 \times n+1)$ looks like

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ z_1 & z_2 & \dots & z_{n+1} \end{pmatrix}$$

problem (2) is transformed to the following one

$$\min\{||z||^2|f_0(Az) + s \leq ||z||^2, f_i(Az) \leq 0, i = 1, \dots, m\}. \tag{3}$$

There is the minimum value $r > 0$ so that all functions $f_0(Az) + s + (r-1)||z||^2$ and $f_i(Az) + r||z||^2$ would be convex. It follows from the fact that Hessians of these functions are positively defined matrixes (matrixes with dominant main diagonal) at the corresponding choice of parameter $r > 0$. Let us use quadratic regularization for transforming problem (3) to the following

$$\min\{||x||^2|g_i(x) \leq d, i = 0, \dots, m, r||x||^2 = d\}, \tag{4}$$

where all $g_i(x)$ are convex functions ($g_0(x) = f_0(Ax)+s+(r-1)||x||^2, g_i(x) = f_i(Ax)+r||x||^2$). It is obvious that Karush-Kuhn-Tucker (KKT) optimality points of problem (1) are also KKT points of problem (4).

Let us designate feasible sets of problem (4) through $S_1(d) = \{x|g_i(x) \leq d, i = 0, \dots, m\}$ and $S_2(d) = \{x|r||x||^2 \leq d\}$. It is obvious that set $S_2(0)$ consists of one point $x = 0$, and set $S_1(0)$ can be either empty, or contain point $= 0$ or not contain it. It is easy to determine value $d_0$ so that $S_1(d_0)$ also consisted of one point (generally it can be a convex set). It will be the minimum value $d$ so that $S_1(d) \neq \oslash$. In order to find $d_0$, let us consider the following algorithm. Let us define the centre of convex set $S_1(d)$, and let it be point $x^0$. Let us set $d = S_1(x_0) + 1$ and find the centre of convex set $S_1(d)$ once more. Let $d = S_1(x^k) + 1/(k+1)$ on $k$-th iteration. Value $d_0$ is the limiting point of sequence $S_1(x^k)$ as $S_1(x_k)$ decreases and is limited from below if the feasible set of problem (4) is limited. Further let us determine the minimum feasible value $d$ according to formula $d_m = \max\{0, d_0\}$.

Studying the properties of sets $S_1(d)$ and $S_2(d)$ allows breaking problem (4) into two complexity classes. If $S_1(d_m) \cap S_2(d_m) = \oslash$ or $0 \in S_1(d_m)$ problem (4) is equivalent to the following convex quadratic problem

$$\min\{||x||^2|g_i(x) \leq d_{min}, i = 0, \dots, m\}. \tag{5}$$

In this case it is necessary to find the tangency point of two convex sets $S_1(d)$ and $S_2(d)$ at the minimum value $d$. It is not difficult to check condition $S_1(d_m) \cap S_2(d_m) = \oslash$ as one of these sets is a point. If $S_1(d_m) \cap S_2(d_m) \neq \oslash$, problem (4) is equivalent to the following concave optimization problem

$$\max\{||x||^2|g_i(x) \leq d_{min}, i = 0, \dots, m\}, \tag{6}$$

where it is necessary to find the minimum value $d$ so that set $S_1(d)$ touched set $S_2(d)$ from within. Let us consider the algorithm of solving problem (6) which has shown practical efficiency at solving many test problems.

## 3. The method of positive orthant displacement for solving concave quadratic optimization problem

To restrict the search domain of the global minimum point in problem (6), let us transform problem (1) so that the variables possessed only positive values. For this purpose it is enough to present the variables of problem (6) as the difference of two positive variables. Therefore, let us consider the following problem

$$\max\{||x||^2 | g_i(x) \leq d, i = 0, \dots, m, x \geq 0\}, \tag{7}$$

at the fixed value of parameter $d$ then $S_1(d) = \{x | g_i(x) \leq d, i = 1, \dots, m, x \geq 0\}$. Let us solve the auxiliary problem

$$\max\{e^T x | x \in S_1(d)\},$$

where $e = (1, \dots, 1)$. If condition $r||x^0||^2 \geq d$ in is satisfied solution point $x^0$, value $d$ is to be reduced ($d_{max} = d$). If $e^T x^0 \leq \sqrt{d/r}$, value $d$ is to be increased ($d_{min} = d$). In other cases it is necessary to check whether set $S_1(d)$ crosses the border of set $S_2(d)$.

Let us divide a spherical segment $\{x | e^T x \geq d/r, x \in S_2(d)\}$ into $n$ spherical subsegments

$$W_k = \{x | \sum_{i \neq k} x_i + (\sqrt{n} - n + 1)x_k \geq \sqrt{d/r}, x \in S_2(d)\}, k = 1, \dots, n.$$

Let us solve $n$ convex optimization problems

$$\max\{\sum_{i \neq k} x_i + (\sqrt{n} - n + 1)x_k | x \in S_1(d)\}, k = 1, \dots, n.$$

It is obvious that if $\exists k, x^k \notin S_2(d)$ it is necessary to reduce value $d$ and if $\forall k, x^k \notin W_k$, it is necessary to increase it. Otherwise, in order to continue searching of maximum it is enough to consider only such values $k$ for which $x^k \in W_k$.

Let us narrow domain of searching the maximum point in $k$-th segment by displacing the coordinate hyperplanes until they touch domain $W_k \cap S_1(d)$. For this purpose it is necessary to solve $n$ convex optimization problems. In the received positive orthant let us continue the procedure of searching the maximum point considered above.

Searching the maximum point in problem (6) is finished when point $x^k \notin S_2(d)$ or $\forall k, x^k \notin W_k$ is found. It leads to changing of parameter $d$. Let us find the optimum value of this parameter by the dichotomy method.

The considered procedure of dividing a spherical segment can be considerably reduced if the following method of displacing a positive orthant is used for the problem solution:

$$\min\{^T x | x \in S_1(d), e^T x \geq \sqrt{d/r}, r||x||^2 = d\}, \tag{8}$$

where all $c_i > 0$. The method can be used for searching the maximum point of problem (7) in each subsegment.

1. Designate the coordinate origin through $x^0$. Find the intersection points of coordinate axes with the border of set $S_2(d)$. Designate the segment lengths on coordinate axes (the distance from point $x^0$ to the intersection point) through $q_i$. If the feasible point of problem (8) corresponds to the minimum value $q_i$, the global minimum point is found. It is equal to $(x_1^0, \dots, x_i^0 + q_i, \dots, x_n^0)$. Otherwise, solve convex programming problem

$$\min\{c^T x | x \in S_1(d), e^T x \geq \sqrt{d/r}, x \in S_2(d)\}.$$

If its solution lies on the border of set $S_2(d)$, it is the solution of problem (8).

2. Solve the sequence of convex optimization problems

$$\min\{x_i | x \in S_1(d), e^T x \geq \sqrt{d/r}, \sum_{i=1}^{n} x^i/q_i \geq 1\}, i = 1, \dots, n. \tag{9}$$

Define the coordinates of new coordinate origin: $x_i^0 = x_i^0 + x_i^*$ where $x_i^*$ is the solution of the $i$-th problem (9). Pass to step 1.

The algorithm finds the global minimum point on the border of sphere $S_2(d)$. For checking the feasible points out of the sphere it is enough to solve problem (8) once again having replaced condition $r||x||^2 = d$ by $r||x||^2 = d + \varepsilon$. If the feasible set of transformed problem (8) is empty, the method of displacing a positive orthant finds the global minimum point of problem (1).

While realizing the numerical algorithm it was found that the algorithm might not converge if the displacement of coordinate planes is not performed on any of the iterations. It means that it is necessary to search for the solution on one of coordinate hyperplanes. But it leads to the branch and bound method. It is proposed to perform space stretching in the direction of vector $c$ (see below). It often allows continuing the displacement of the coordinate origin. It is equivalent to the following linear coordinate transformation $y = [(\alpha - 1)cc^T/n + I]x$, where $\alpha > 1$ is the factor of space stretching, $c$ is the objective function vector and $I$ is the unit matrix. The polyhedron points on coordinate hyperplanes will be displaced into the positive orthant in the result of stretching. Only in case the method of displacing a positive orthant does not allow finding the global minimum point, the procedure of dividing a spherical segment on $n$ subsegments considered above is used.

## 4. Summary

The new approach to solving the general quadratic problems which is alternative to semidefinite relaxation is considered. Semidefinite relaxation methods can be used for getting the initial approximation in the considered algorithms and estimation of solution accuracy. The semidefinite relaxation can be used for checking of the found solution optimality. If the solution of semidefinite optimization problem has unit rank in the given point, the exact solution is found.

Let's notice, that the considered method is easily generalised for the solution of the general problems of nonlinear optimization.

Computational experiments with the proposed algorithms on problemsfrom the literature and also randomly generated instances suggest that our algorithms are efficient for solving general quadratic programs.

## References

[1] Ding Y. (2007). *On Efficient Semidefinite Relaxations for Quadratically Constrained Quadratic Programming*. Waterloo, Ontario, Canada.

[2] Floudas C.A., Visweswaran V. (1995). *Quadratic optimization* Princeton, NY: Princeton University Press.

[3] Lasserre J. (2001). *Global optimization with polynomials and the problem of moments*. SIAM J.Optim., Vol. 11, N. 3.

[4] Shor N.Z. (1998). *Nondifferentiable optimization and polinomial problems*. Dordrecht: Kluwer Academic Publishers.

# A "joint+marginal" algorithm for polynomial optimization

Jean B. Lasserre

*LAAS-CNRS and Institute of Mathematics, Toulouse, France,* lasserre@laas.fr

**Keywords:** polynomial optimization, semidefinite relaxation.

We present a new algorithm for solving a polynomial program P based on the recent "joint + marginal" approach proposed by the author for parametric polynomial optimization. The idea is to first consider the variable $x_1$ as a parameter and solve the associated (n-1)-variable $(x_2, \ldots, x_n)$ problem $P(x_1)$ where the parameter $x_1$ is fixed and takes values in some interval $Y_1 \subset I\!R$, with some probability $\varphi_1$ uniformly distributed on $Y_1$. Then one considers the hierarchy of what we call "joint+marginal" semidefinite relaxations, whose duals provide a sequence of univariate polynomial approximations $x_1 \mapsto p_k(x_1)$ that converges to the optimal value function $x_1 \mapsto J(x_1)$ of problem $P(x_1)$, as $k$ increases. Then with $k$ fixed a priori, one computes $x_1^* \in Y_1$ which minimizes the univariate polynomial $p_k(x_1)$ on the interval $Y_1$, a convex optimization problem that can be solved via a single semidefinite program. The quality of the approximation depends on how large $k$ can be chosen (in general for significant size problems $k = 1$ is the only choice). One iterates the procedure with now an (n-2)-variable problem $P(x_2)$ with parameter $x_2$ in some new interval $Y_2 \subset I\!R$, etc. so as to finally obtain a vector $\tilde{x} \in R^n$. Preliminary numerical results are provided.

# Global search for pessimistic solution in bilevel problems

Anton V. Malyshev and Alexander S. Strekalovsky

*Institute for System Dynamics and Control Theory SB RAS, Irkutsk, Russia,* anton@irk.ru, strekal@icc.ru

**Abstract** A quadratic-linear bilevel optimization problem in its pessimistic statement is considered. It is reduced to a series of bilevel optimization problems in its optimistic statement and then to nonconvex optimization problems. Global and local search algorithms for the latter problems are developed.

**Keywords:** bilevel optimization, pessimistic solution, nonconvex optimization problems, local search, global search, computational simulation

## 1. Introduction

At the present hierarchical optimization problems seem to be one of the most attractive fields for many experts [1]–[3]. In particular, bilevel optimization problems represent extremum problems, which – side by side with ordinary constraints such as equalities and inequalities – include a constraint described as an optimization subproblem [2], [3], called the lower-level problem.

In course of investigation of bilevel optimization problems the difficulty arises already at the stage of defining the concept of solution. The optimistic and pessimistic (guaranteed) definitions of a solution are known to be the most popular [1]–[3]. During the three decades of intensive investigation of bilevel optimization problems there were proposed various methods for finding an optimistic solution by different authors (see the survey [4]). But there was a few for finding pessimistic solutions [5], [6].

This paper is concerned with a new approach for finding a pessimistic solution to bilevel problems, where the upper-level goal function is a convex quadratic function of upper-level variables and concave quadratic w.r.t. lower-level variables, besides the lower-level goal function is linear.

Here, the techniques from [5] are developed to reduce such bilevel optimization problems to a series of bilevel optimization problems in its optimistic setting. Latter bilevel problems may be reduced to a series of single-level problems via the KKT-rule (see, for example, [2]) and penalty method. For the purpose of solving the single-level problems obtained, which turn out to be nonconvex, we construct an algorithm based on the Global Search Theory developed in [7]–[10].

The paper is organized as follows. In section 2 we present the statement of the bilevel problem and its reduction. Next, in the section 3 algorithms of local and global search for reduced problems are considered.

## 2. Problem statement and its reduction

Consider the bilevel optimization problem in its pessimistic statement, as follows:

$$\left.\begin{array}{c} \sup\limits_y\{F(x,y) \mid y \in Y_*(x)\} \downarrow \min\limits_x, \\ x \in X,\ \ Y_*(x) \overset{\triangle}{=} \operatorname*{Argmin}\limits_y\{G(y) \mid y \in Y(x)\}, \end{array}\right\} \qquad (\mathcal{BP})$$

where $F(x,y) \overset{\triangle}{=} \frac{1}{2}\langle x, Cx\rangle + \langle c, x\rangle - \frac{1}{2}\langle y, C_1 y\rangle + \langle c_1, y\rangle$, $G(y) \overset{\triangle}{=} \langle d, y\rangle$, $X \overset{\triangle}{=} \{x \in I\!R^m \mid Ax \leq a, x \geq 0\}$, $Y(x) \overset{\triangle}{=} \{y \in I\!R^n \mid A_1 x + B_1 y \leq b, y \geq 0\}$, $A \in I\!R^{p\times m}$, $A_1 \in I\!R^{q\times m}$, $B_1 \in I\!R^{q\times n}$, $C \geq 0$, $C_1 \geq 0$, $a, b, c, c_1, d$ — the vectors of corresponding dimensions.

Also consider the following auxiliary bilevel optimization problem in its optimistic statement:

$$\left.\begin{array}{c} F(x,y) \downarrow \min\limits_{x,y},\ \ x \in X, \\ y \in \operatorname{Argmin}\{G(y) - \nu F(x,y) \mid y \in Y(x)\}, \end{array}\right\} \qquad (\mathcal{BP}_o(\nu))$$

where $\nu > 0$ is a penalty parameter.

Suppose, feasible sets of upper level and lower level are bounded so that

$$(\mathcal{H}):\ X \text{ is a bounded set, and } \exists Y : Y \supseteq Y(x)\ \ \forall x \in X,\ \ Y \text{ is a compact set.} \qquad (1)$$

The following theorem develops corresponding result from [5].

**Theorem 1.** *Suppose the condition $(\mathcal{H})$–(1) takes place and number sequences $\{\nu_k\}, \{\tau_k\}$ tend to zero: $\nu_k \downarrow 0$, $\tau_k \downarrow 0$. Then any limit point of the sequence $\{x^k, y^k\}$ of approximate $\tau_k$-solutions of problems $(\mathcal{BP}_o(\nu_k))$ is a pessimistic solution to problem $(\mathcal{BP})$.*

So, in order to solve bilevel problem in pessimistic statement $(\mathcal{BP})$ it's proposed in theorem 1 to solve the series of bilevel problems in optimistic statement $(\mathcal{BP}_o(\nu_k))$ corresponding to the sequence $\{\nu_k\} : \nu_k \downarrow 0$.

Further more, replacing the lower-level problem with its Karush–Kuhn-Tukker (KKT) conditions and using penalty method, problem $(\mathcal{BP}_o(\nu))$ with a fixed value $\nu$ can be reduced to a series of the problems (see, for example, [2], [3], [11]), as follows:

$$\left.\begin{array}{c} \Phi(x,y,v) \overset{\triangle}{=} F(x,y) + \mu h(x,y,v) \downarrow \min\limits_{x,y,v}, \\ (x,y,v) \in D \overset{\triangle}{=} \{(x,y,v) \mid Ax \leq a,\ \ x \geq 0,\ \ A_1 x + B_1 y \leq b,\ \ y \geq 0, \\ d - \nu c_1 + \nu C_1 y + v B_1 \geq 0,\ \ v \geq 0\}, \end{array}\right\} \qquad (\mathcal{P}(\mu,\nu))$$

where $h(x,y,v) \overset{\triangle}{=} \nu\langle y, C_1 y\rangle + \langle d - \nu c_1, y\rangle + \langle b - A_1 x, v\rangle$, $v \in I\!R^q$ is a vector of Lagrange multipliers for the lower-level problem, $\mu > 0$ is a penalty parameter. In what follows, we assume that $\mu\nu \geq \frac{1}{2}$.

It can be readily seen that problem $(\mathcal{P}(\mu,\nu))$ is a nonconvex quadratic optimization problem. Its objective function $\Phi(\cdot)$ is a d.c. function, i.e. it can be represented as difference of two convex functions, for example, as follows:

$$\Phi(x,y,v) = g(x,y,v) - f(x,y,v), \qquad (2)$$

where $g(x,y,v) = \frac{1}{2}\langle Cx, x\rangle + \langle c, x\rangle + \Big(\mu\nu - \frac{1}{2}\Big)\langle C_1 y, y\rangle + \mu\Big(\langle a_1, v\rangle + \frac{1}{4}\|v - A_1 x\|^2\Big)$ and $f(x,y,v) = \langle(\mu\nu - 1)c_1 - \mu d, y\rangle + \frac{1}{4}\mu\|v + A_1 x\|^2$ are convex functions.

We propose to apply the Global Search Strategy for d.c. optimization problems developed in [7]–[10] for solving problem $(\mathcal{P}(\mu,\nu))$ with fixed values $\nu > 0$, $\mu > 0$. A global search algorithm based on this strategy and one of its key elements – local search – are described in the next section.

## 3. Local and global search

The idea of local search for problem $(\mathcal{P}(\mu,\nu))$ consists in consecutive solving of the problem with respect to different groups of variables (see [8], [11]) in the case of joined variables. For a fixed value of variable $v$ problem $(\mathcal{P}(\mu,\nu))$ becomes a convex quadratic optimization problem (QP), and for a fixed pair $(x,y)$ we obtain a problem of linear programming (LP). These auxiliary problems can be solved by means of standard software packages. So, we obtain the following special local search method.

Let $(x_0,y_0,v_0)\in D$ be a starting point.
**Step 0**. Put $s:=0,\;\; v^s:=v_0$.
**Step 1**. Apply a method of quadratic optimization which yields a $\frac{\rho_s}{2}$-solution $(x^{s+1},y^{s+1})$ to the problem:

$$\Phi(x,y,v^s)\downarrow\min_{x,y},\quad (x,y,v^s)\in D.$$

**Step 2**. Obtain a $\frac{\rho_s}{2}$-solution $v^{s+1}$ of the linear problem

$$\Phi(x^{s+1},y^{s+1},v)\downarrow\min_{v},\;\; (x^{s+1},y^{s+1},v)\in D,$$

**Step 3**. Put $s:=s+1$, and go to Step 1. #

The following theorem provides us the convergence of this method, which is called V-procedure.

**Theorem 2.** *i) If* $\rho_s>0, s=0,1,2...,\ \sum\limits_{s=0}^{\infty}\rho_s<+\infty$ *then the sequence* $\{\Phi_s\}, \Phi_s\overset{\triangle}{=}\Phi(x^s,y^s,v^s)$, *of values of function* $\Phi(\cdot)$, *which is generated by the V-procedure, converges.*
*ii) If* $(x^s,y^s,v^s)\rightarrow(\hat{x},\hat{y},\hat{v})$, *then the limit point* $(\hat{x},\hat{y},\hat{v})$ *satisfies the following inequalities:*

$$\Phi(\hat{x},\hat{y},\hat{v})\le\Phi(x,y,\hat{v})\quad\forall(x,y)\in Z,\tag{3}$$
$$\Phi(\hat{x},\hat{y},\hat{v})\le\Phi(\hat{x},\hat{y},v)\quad\forall v\in V.\tag{4}$$

The triple $(\hat{x},\hat{y},\hat{v})$ satisfying inequalities (3) and (4) shall henceforth be called a critical point of problem $(\mathcal{P}(\mu,\nu))$. If the inequalities (3) and (4) are satisfied with certain accuracy for some point, we call this point *approximately critical*.

Stopping criteria for V-procedure have been substantiated in order to detect an approximately critical point with given accuracy.

To obtain the local search procedure for problem $(\mathcal{P}(\mu,\nu))$, according to the logic from [8], [11], it is possible to consider another variant of its implementation, in which auxiliary problems are solved in a different order (initially – with respect to $v$, and after that – with respect to $(x,y)$). This version of the local search method is called XY-procedure.

Since local search methods do not, generally speaking, yield a global solution even for a small-dimensional problem, we developed a global search algorithm (GSA) for problem $(\mathcal{P}(\mu,\nu))$ based on Global Search Strategy for d.c. optimization problems [7]–[11]. GSA consists of two principal stages: a) finding a critical point by means of a local search method and b) escaping from a current critical point with the help of procedures based on global optimality conditions [7], [9].

The local and global search algorithms described above have been programmed in C++. For constructing test problems we developed the approach proposed in [12] in the case of pessimistic solution. The approach implies constructing bilevel problems of various complexities and dimensions from so-called kernel bilevel problems of small dimension with known local and global solutions. We used the following kernel problems solved analytically:

$$\left.\begin{array}{c} \sup\limits_{y}\{x^2 - 8x + py_1 - 2y_2^2 \mid y \in Y_*(x)\} \downarrow \min\limits_{x}, \\ x \in [0;6], \;\; Y_*(x) \overset{\triangle}{=} \operatorname*{Argmin}\limits_{y}\{-y_1 \mid y_1 + y_2 \leq x, \;\; y_1 \leq 3, \;\; y_1 \geq 0, \;\; y_2 \geq 0\}, \end{array}\right\} \qquad (\mathcal{KP})$$

where $p$ is a real-valued parameter which determines the complexity of the problem.

Preliminary testing of the global search program demonstrated its rather competitive effectiveness. For example, we were successful to solve all problems generated up to the dimension $m = 20, n = 40$ in a reasonable time.

## 4. Summary

A quadratic-linear bilevel optimization problem in its pessimistic statement was considered. Using its reduction to a series of nonconvex optimization problems, global search algorithm for the problem was proposed. Preliminary computational testing of global search algorithm demonstrated its efficiency on generated problems.

## References

[1] Germeyer, Yu.B. (1976). *Games with nonantagonistic interests (in Russian)*. Moscow, Russia: Nauka.

[2] Bard, J.F. (1998). *Practical Bilevel Optimization*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

[3] Dempe, Stephan (2002). *Foundations of Bilevel Programming*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

[4] Colson, Benoit; Marcotte, Patrice; Savard, Gilles (2007). *An overview of bilevel optimization*. Annals of operations research, V.153, No.1, pages 235–256.

[5] Molodtsov, D.A. (1976). *The solution of a certain class of non-antagonistic games*. Computational Mathematics and Mathematical Physics, V.16, No.6, pages 1451–1456.

[6] Tsoukalas, Angelos; Wiesemann, Wolfram; Rustem, Berc (2009). *Global Optimisation of Pessimistic Bi-Level Problems*. Pardalos, Panos M. (ed.) et al., Lectures on global optimization. Papers based on the presentations at the workshop on global optimization: methods and applications, Toronto, Canada, May 11–12, 2007. Providence, RI: American Mathematical Society (AMS); Toronto: The Fields Institute for Research in Mathematical Sciences (ISBN 978-0-8218-4485-4/hbk). Fields Institute Communications 55, pages 215–243.

[7] Strekalovsky, A.S. (2003). *Elements of Nonconvex Optimization (in Russian)*. Novosibirsk, Russia: Nauka.

[8] Strekalovsky, A.S.; Orlov, A.V. (2007). *Bimatrix games and bilinear programming (in Russian)*. Moscow, Russia: FizMatLit.

[9] Strekalovsky, A.S. (2003). *On the Minimization of the Difference of Convex Functions on a Feasible Set*. Computational Mathematics and Mathematical Physics, V.43, No.3, pages 380–390.

[10] Strekalovsky, A.S.; Orlov, A.V. (2007). *A new approach to nonconvex optimization*. Numerical Methods and Programming (internet-journal: http://num-meth.srcc.msu.su/english/index.html), V.8, pages 160–176.

[11] Strekalovsky, A.S; Orlov, A.V.; Malyshev, A.V. (2010). *On computational search for optimistic solutions in bilevel problems*. Journal of Global Optimization, Special Issue celebrating the 70th birthday of Roman G. Strongin (in print; available online, DOI: 10.1007/s10898-009-9514-z).

[12] Calamai, Paul; Vicente, Luis (1994). *Generating Quadratic Bilevel Programming Test Problems*. ACM Transactions on Mathematical Software, V.20, pages 103–119.

# The averaged problem[*]

Ken McKinnon[1] and Marcel Mongeau[2,3]

[1]*Department of Mathematics & Statistics, University of Edinburgh, Edinburgh EH9 3JZ, UK,* ken@maths.ed.ac.uk

[2]*Université de Toulouse; UPS, INSA, UT1, UTM; Institut de Mathématiques de Toulouse; F-31062 Toulouse, France,* mongeau@math.univ-toulouse.fr

[3]*CNRS; Institut de Mathématiques de Toulouse UMR 5219; F-31062 Toulouse, France*

**Abstract** We consider a particular global optimization problem whose special structure permits the use of a decomposition algorithm optimizing globally only in search spaces of lower dimension. Roughly speaking, this problem involves the optimization of a weighted average of objective function values, subject to the same weighted average of constraints, where the weights are to be determined. We present theoretical results and an algorithmic approach to solve problems of this class.

**Keywords:** Gibbs free energy, decomposition, convexity, phase equilibrium

## 1. Definition of the problem

We consider in this talk a particular global optimization problem which, first, has various applications (for instance, in chemical engineering and in resource allocation), and, secondly, whose special structure permits the use of a decomposition algorithm optimizing globally only in search spaces of lower dimension. We call this problem the *averaged problem*. It has the following form:

$$\begin{array}{rrcl} \displaystyle\min_{K,x,y} F(K,x,y) := & \displaystyle\sum_{k=1}^{K} y_k f(x_k) & & \\ \text{subject to} & \displaystyle\sum_{k=1}^{K} y_k g(x_k) & = & b, \\ & y & \in & S_K, \\ & x & \in & X^K, \\ & K & \in & I\!N^+, \end{array} \qquad \text{(P)}$$

where

$$S_K := \{y : \sum_{k=1}^{K} y_k = 1, y_k \geq 0, \quad k = 1, 2, \ldots, K\};$$

$X \subseteq I\!R^n$; $f$ and $g$ are given functions $f : X \to I\!R$, $g : X \to I\!R^m$; $m$ is a given positive integer; and $b$ is a given vector of $I\!R^m$. The optimization variables are: $K$, a positive integer, $y \in I\!R^K$, where $y_k$ denotes the $k$th component of vector $y$, and $x_k \in X$, $x$ being the vector

[*] ACKNOWLEDGEMENT: This work has been supported by French National Research Agency (ANR) through COSINUS program (project ID4CS nřANR-09-COSI-005)

$(x_1, x_2, \ldots, x_K) \in X^K := X \times \cdots \times X$. We note that the dimension, $K$, of the optimization variables is itself an optimization variable.

Roughly speaking, this problem involves the optimization of a weighted average of objective function values, subject to the same weighted average of constraints, where the weights, $y_k$'s, are to be determined. By analogy with the specific chemical engineering instances of the averaged problem, we shall say that $K$ is the *n*umber of phases and we shall call $y_k$, and $x_k$ respectively the *weight*, and the *composition* of the *k*th phase.

## 2. Motivation

This problem was introduced by Whittle [5] four decades ago in order to take into account Lagrangian duality when convexity is lacking. To our knowledge, no one worked on the practical resolution of the averaged problem.

Various applications motivate our study. Special instances of the averaged problem are:

PEP The phase equilibrium problem.

Case where $f(x) := \min\{f_{\text{Liquid}}(x), f_{\text{Vapour}}(x)\}$, $X$ is an open simplex, $g(x) := x$, and $f_{\text{Liquid}}$ (respectively $f_{\text{Vapour}}$) is some given coercive smooth function whose specific form depends on the thermodynamic model chosen to describe a liquid phase (respectively a vapour phase). One must determine the number, types, proportions and compositions of each phase. See for instance [4] and references therein;

CEP The chemical equilibrium problem.

The same case except for the fact that $g$ has rather the form $g(x) := Ax$, where $A$ is an $m \times n$ real matrix of rank $m$ related to the amount of each chemical element per unit amount of each substance involved in the chemical reaction (see again for instance [4] and references therein);

RA Resource allocation.

A very simple example involves choosing among many processes for producing a material. Given the production costs, $f_i(x)$ of producing at a rate $x$ per hour with process $i$, $i \in \tau$ (where $\tau$ is some finite index set), find the least-cost way of producing at a given rate $b$, assuming that only one process can operate at the time, assuming that adequate storage is available, and that the costs of storage, changing production process or production rate are negligible. In such a case, we consider $f(x) := \min_{i \in \tau}\{f_i(x)\}$.

Another example is the allocation of discharge of water among generating units of a hydro-electric generating station so as to maximize power output within operating constraints. To each type of unit corresponds a particular power-output function of the discharge. Here $b$ is the average discharge through the station in $m^3/s$.

LP Classical linear programming is itself a special case of the averaged problem. Indeed, in the case where a positive integer $K$, some scalars $c_k$, and some vectors $a_k$ are given, and where $X := \{1, 2, \ldots, K\}$ (discrete index set), $f(x) = c_k$ if $x = k$ and $g(x) = a_k$ if $x = k$ ($k = 1, 2, \ldots, K$), then the averaged problem (P) simply reduces to the linear programming problem.

Incidentally, we shall see in this talk that the chemical and phase equilibrium global optimization algorithm GILO, introduced in [4] and described in the next section, can be viewed as an extension of the simplex method for linear programming. This is obvious when substituting here $y$ for $x$ in order to respect the notational conventions of linear programming:

$$\begin{array}{rcl} \min\limits_{x} \sum_{k=1}^{K} x_k c_k & := & z, \\ \text{subject to} \quad Ax & = & b, \\ \sum_{k=1}^{K} x_k & = & 1, \\ x & \geq & 0, \end{array} \qquad \text{(LP)}$$

where $A$ is an $m \times n$ matrix having $a_k$ as $k$th column. From Dantzig's fundamental book [1], when evoking the simplex interpretation of the simplex method:

> It was in this geometry that the simplex method was first seriously proposed after it had been earlier set aside as unpromising. The variables $x_k$ were interpreted as non-negative weights to be assigned to a system of points $A_k = (a_k, c_k)$ in the space $(u, v)$, $u \in I\!R^m$ and $v \in I\!R$, so that their weighted average (center of gravity) is a point $(b, \text{Min } z)$. That is to say the $x_k \geq 0$ are chosen so that the center of gravity lies on the "requirement line" $u = b$ (constant), and such that the $v$-coordinate is minimum.

## 3. Algorithm GILO and its extension to the averaged problem

The algorithm GILO (Global Improvement Local Optimization) was introduced as an alternative to solving the phase or chemical equilibrium problem as a single optimization problem. GILO rather iterates between a local minimization problem and a global search in a *lower* dimensional space, taking advantage of the special structure of the problem. Under mild assumptions, the GILO algorithm is guaranteed to find a global minimizer [3, 4]. The proofs of convergence are based on results in [2] which derives the global optimality conditions for the problem.

The global optimization algorithm we shall present is a further generalization. The proofs behind GILO relied on the fact that $X$ be an open set, and that $f$ tend towards infinity as its argument approaches the frontier of $X$. It is hence not readily extendable to the more general case of the averaged problem, where no such assumptions are made.

## 4. Summary

The phase and chemical equilibrium problem involving multiple phase classes is a difficult global optimization problem. Necessary and sufficient conditions for global optimality based on the tangent-plane criterion has been derived as well as an algorithmic approach, called GILO, that reduces this global optimization problem to a *finite* sequence of local optimization (LO) steps and global optimization (GI) steps in a smaller search space. This algorithmic approach can be extended to the more general averaged problem which has further applications such as resource allocation.

## References

[1] George B. Dantzig. *Linear Programming and Extensions.* Princeton University Press, 1963.

[2] Y. Jiang, W. R. Smith, and G. R. Chapman. Global optimality conditions and their geometric interpretation for the chemical and phase equilibrium problem. *SIAM Journal on Optimization*, 5(4):813–834, 1995.

[3] K. I. M. McKinnon, C. G. Millar, and M. Mongeau. Global optimization for the chemical and phase equilibrium problem using interval analysis. In C. Floudas and P.M. Pardalos, editors, *State of the Art in Global Optimization*, pages 365–382. Kluwer, 1996.

[4] K. I. M. McKinnon and M. Mongeau. A generic global optimization algorithm for the chemical and phase equilibrium problem. *Journal of Global Optimization*, 12(4):325–351, 1998.

[5] P. Whittle. *Optimization Under Constraints*. Wiley-Interscience, 1971.

Proceedings of TOGO 2010, pp. 85 – 88.

# Toward Global Minimum Solutions for the Problem of the Energy Consumption of an Electrical Vehicle

Abdelkader Merakeb[1,2] and Frédéric Messine[2]

[1] *University of Tizi-Ouzou, Algérie* Abdelkader.Merakeb@n7.fr

[2] *ENSEEIHT-IRIT, Université PRES Toulouse, France* Frederic.Messine@n7.fr

**Abstract** In this paper we discuss about the way to approximate the global solution of an optimal control problem of Bang-Bang type via a discretization technique associated with a Branch and Bound algorithm. The problem that we focuss on is the minimization of the consumption of energy of an electrical car during some imposed displacements.

**Keywords:** Discretization techniques, Optimal Control, Bang-Bang problem, Branch and Bound

## 1. Introduction

In this paper we discuss about the way to solve efficiently the problem of the minimization of the energy which is consummated by an electrical car during an imposed displacement, see [1] for an overview on this type of problems.

The problem that we are interested with can be formulated as follows:

$$\left\{\begin{array}{ll} \min_{i_m(t),\Omega(t),pos(t),u(t)} & E(i_m,u)=\int_0^{t_f} u(t)i_m(t)V_{alim}+R_{bat}u^2(t)i_m^2(t)dt \\ u.c. & \\ & \dot{i_m}(t)=\frac{u(t)V_{alim}-R_m i_m(t)-K_m\Omega(t)}{L_m} \\ & \dot{\Omega}(t)=\frac{1}{J}\left(K_m i_m(t)-\frac{r}{K_r}\left(MgK_f+\frac{1}{2}\rho SC_x\left(\frac{\Omega(t)r}{K_r}\right)^2\right)\right) \\ & \dot{pos}(t)=\frac{\Omega(t)r}{K_r} \\ & |i_m(t)|\leq 150 \\ & u(t)\in\{-1,1\} \\ & (i_m(0),\Omega(0),pos(0))=(i_m^0,\Omega^0,pos^0)\in I\!R^3 \\ & (i_m(t_f),\Omega(^{t}_f),pos(t_f))\in\mathcal{T}\subseteq I\!R^3 \end{array}\right. \tag{1}$$

where $E$ represents the electrical energy consummated during the displacement. The state variables are: (i) $i_m$ the current inside the motor; (ii) $\Omega$ the angular speed, with $V(t)=\frac{3.6\times r}{K_r}\times\Omega(t)$ which represents the speed of the car in $km/h$ ($r$ is the radius of the wheel); (iii) $pos$ is the position of the car. The control $u$ is in $\{-1,1\}$ (a Bang-Bang control); the physical system can switch in $10^{-9}s$. In this problem, we have a constraint on a state variable to limit the current inside the motor in order to discard the possibility to destroy it. The other terms are fixed parameters and represent some physical things: -$K_r=10$, the coefficient of reduction; -$\rho=1.293kg/m^3$, the air density; -$C_x=0.4$, the aerodynamic coefficient; -$S=2m^2$, the area in the front of the car; -$r=0.33m$, the radius of the wheel; -$K_f=0.03$, the constant representing the friction of the wheels on the road; -$K_m=0.27$, the coefficient of the motor torque; -$R_m=0.03\Omega$, the inductor resistance; -$L_m=0.05$, inductance of the rotor; -$M=250kg$, the mass; -$g = 9.81$, the gravity constant; -$J = M \times r^2/K_r^2$; -$V_{alim} = 150$, the battery voltage; -$R_{bat} = 0.05\Omega$, the resistance of the battery. The initial conditions are given but the target set $\mathcal{T}$ is free and depends on the instances of the problem; it could be a point of $I\!R^3$ but one or two variables could not be fixed: for example just the final position equal to $100m$ is required (see the numerical section).

This problem is hard to solve directly by using classical optimal control techniques. We try to solve it by using the Pontriaguine method based on shooting techniques and also by using a direct shooting algorithm, [3]. For the moment, the fact that we have a constraint on the state associated with the fact that it is a Bang-Bang control involves a lot of difficulties which does not permit to obtain solutions (even local ones) to this optimization problem using those two well-known methods. The dynamic programming of Bellman is also difficult to apply to this problem, [2].

Thus, in this paper we propose another original methodology to solve this problem yielding to some discretized problems which are solved using an exact Branch and Bound algorithm. This new method provide exact results for the discretized formulations which correspond to approximations of the global solutions of Problem (1).

## 2. Approximation of Problem (1)

First we remark that if we discretize all the interval of time $[0, t_f]$ by fixing the value of the control $u$, it is necessary to have very small steps about $10^{-3}$ else the value of the current will change too roughly. That will generate a very huge mixed integer non-linear global optimization problem which is, for the moment, impossible to solve using direct methods of optimal control.

Another idea, which directly comes from the numerical simulation of the behavior of the car, is to impose during some short laps of time the value of the current inside the electrical motor of the vehicle. This is possible using the control parameter $u(t)$. Thus, if we impose a reference current *iref*, if $i_m(t) > iref + \frac{\Delta}{2}$ then $u(t) := -1$ and if $i_m(t) < iref - \frac{\Delta}{2}$ then $u(t) := 1$. This technique is just a way to construct a regulator of current which is a first step before making a speed regulator for an electrical car. Hence, using this, the following differential system of equations can be solved:

$$VS(iref, t_0, t_f) := \begin{cases} \dot{E}(t) = u(t)i_m(t)V_{alim} + R_{bat}u^2(t)i_m^2(t) \\ \dot{i_m}(t) = \frac{u(t)V_{alim} - R_m i_m(t) - K_m\Omega(t)}{L_m} \\ \dot{\Omega}(t) = \frac{1}{J}\left(K_m i_m(t) - \frac{r}{K_r}\left(MgK_f + \frac{1}{2}\rho S C_x\left(\frac{\Omega(t)r}{K_r}\right)^2\right)\right) \\ \dot{pos}(t) = \frac{\Omega(t)r}{K_r} \\ u(t) := \begin{cases} -1 \text{ if } i_m(t) > iref + \frac{\Delta}{2} \\ +1 \text{ if } i_m(t) < iref - \frac{\Delta}{2} \\ u(t) \text{ else.} \end{cases} \\ (E(t_0), i_m(t_0), \Omega^(t_0), pos(t_0)) = (E^{t_0}, i_m^{t_0}, \Omega^{t_0}, pos^{t_0}) \in I\!R^4 \\ u(t_0) := 1; \end{cases} \quad (2)$$

where $t_0$ is the initial time which is not necessary equal to $0$. This system of differentiable equations can be efficiently solved using a classical differentiable integrator such as for example *Euler, RK2, RK4* with a step of time less than $10^{-3}$. The function $VS(iref, t_0, t_f)$ will compute in theory all the values for $E(t), i_m(t), \Omega(t), pos(t)$, for all $t \in [t_0, t_f]$ but in practice only values for a discretized time $t_i \in [t_0, t_f]$ is available. Here, we are interested by the final values of the state variables, hence we define a function:

$$VSF(iref, t_0, t_f) := (E(t_f), i_m(t_f), \Omega(t_f), pos(t_f)) \in I\!R^4,$$

all the computations are performed using function *VS* which solves the system of differential equations (2).

The main idea of this work is to subdivide the cycle of time $[0, t_f]$ into $P$ subintervals. In each step of time $[t_{p-1}, t_p]$ with $p \in \{1, \cdots, P\}$ $(t_p = p \times \frac{t_f}{P})$, we apply a reference current $iref_p$ which takes values in $[-150, 150]$ in order to directly satisfy the constraint on the state variable of Problem (1).

Thus, we focuss on the resolution of the following global optimization problem:

$$\left\{\begin{array}{ll} \min_{iref \in [-150,150]^P} & \sum_{k=1}^{P} E_k \\ u.c. & \\ & \\ & (E_k, i_k, \Omega_k, pos_k) := VSF(iref_k, t_{k-1}, t_k) \\ & (E_0, i_0, \Omega_0, pos_0) = (E^0, i_m^0, \Omega^0, pos^0) \in I\!R^4 \\ & (i_P, \Omega_P, pos_P) \in \mathcal{T} \subseteq I\!R^3 \end{array}\right. \tag{3}$$

Problem (3) is a good approximation of the initial problem (1) which generates just a few number of variables: $P$. In fact, we use a current regulator system to control the vehicle; this is also interesting in itself for a future implementation of the system in the car.

## 3. Dedicated Branch and Bound Algorithm

For the moment, we are not able to solve exactly the global optimization problem (3), thus we need to discretize also the possible values for the reference current: $iref \in \{-150, -150 + s, -150 + 2 \times s, \cdots, 150\}^P$; we will take integer values for $s$ which divide exactly $[-150, 150]$. Therefore, the set of solution becomes finite and could be enumerated. Nevertheless, if we want to have a good approximation for the resolution of the global optimization problem (3) we have to discretize into small steps and the finite set of possible points becomes rapidly too huge to be entirely enumerated in a reasonable CPU-time.

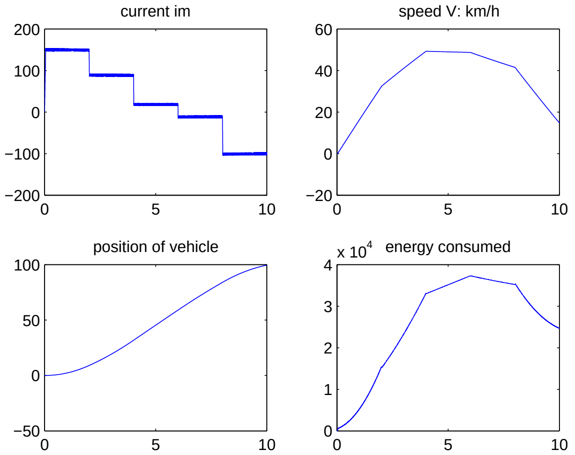
The idea is then to use a Branch and Bound algorithm in order to not explore all the finite set of solutions. For using such an algorithm, we have to elaborate a technique to compute bounds for the four main parameters: $E_k, i_k, \Omega_k, pos_k$ over a box $IREF \subseteq \{-150, -150 + s, -150 + 2 \times s, \cdots, 150\}^P$ and for given $t_0$ and $t_f$. In order to be more efficient, in a previous step, we compute 4 matrices: $M_E, M_i, M_\Omega, M_{pos}$ where the columns corresponds to values when $iref$ is fixed with $i_m^{t_0} = iref$ and the lines provides values for the entities when a speed $\Omega^{t_0}$ is given (we discretize also the possible values of the speed). For example $m_E(i, j)$ represents the value of the energy which is consummated during a step of time $t_p - t_{p-1}$ when $iref$ is equal to the $j$th components in $\{-150, -150 + s, -150 + 2 \times s, \cdots, 150\}$ with $i_m^{t_0} = iref$ and the $i$th discretized value for the speed, the other initial values are taken equal to 0: i.e., $E^{t_0} = pos^{t_0} = 0$.

When a box $IREF$ is considered, we can compute bounds for $E, i, \Omega$ and $pos$ by computing the integer sets $I$ and $J$ of the indices corresponding to the possible values of the speed at the previous step and the possible values of $iref$. Then, we have to compute the bounds which correspond to the minimal and maximal values of $m_E(i, j), m_i(i, j), m_\Omega(i, j), m_{pos}(i, j)$ with $(i, j) \in I \times J$. To obtain the final value for $E$ and $pos$, we have to sum all the lower and upper bounds. The rest of the Branch and Bound algorithm that we develop is simple and uses the following classical principle: (i) subdivision into two (distinct) parts of the enumerate set $IREF$ (which represents the possible values for $iref$); (ii) the upper bound is updated by taking the middle of the box $IREF$ if the constraints are satisfied and if its value is better than the previous one (we start with $+\infty$);(iii) we branch following the heuristic of lowest lower bound of the energy.

## 4. Numerical Experiments

To illustrate our method, we simulated it for a displacement of 100 meters, and a cycle $t_f = 10$ seconds: $(i_m(0), \Omega(0), pos(0)) = (0, 0, 0)$; $(i_m(t_f), \Omega(t_f), pos(t_f)) \in \mathcal{T} = I\!R \times I\!R \times \{100\}$.

For the resolution of the VSF function, we used the integrator Runge-Kutta at the order 4, with a step of time equal to $10^{-3}$, simulated on MatLab 7 on a standard PC Laptop with 2GB of RAM. The parameters for our code are fixed to $P = 5$, $s = 5$, $0.1km/h$ for the step of the discretization of the speed (to compute the matrices $M$) and $\Delta = 1$. Thus, we obtain the exact (for the discretized problem) solution $iref^* = (150, 90, 25, -15, -110)$ corresponding to the minimal value $E^*(10) = 24430.21J$. Moreover, we have $pos^*(10) = 100.03m$. The CPU-time computation is about $966s$ corresponding to 201830 iterations of the Branch and Bound algorithm. This long CPU-time strongly depends on the parameter $s$ and also $P$ which is understandable for a Branch and Bound code (the complexity of such an algorithm depends on $\left(\frac{2\times150}{s}+1\right)^P$). Thus, if we take $s = 10$, we obtain the following results: $iref^* = (150, 90, 20, -10, -100)$ corresponding to the minimal value $E^*(10) = 24589.90J$ with a position of $100.15m$ in only $17.62s$ for 31090 iterations. This solution is represented in the following figure:



Therefore an idea to obtain much more precise solutions, is simply to run the Branch and Bound code iteratively by defining more and more precise zones around the previous exact solutions and by increasing parameter $P$ and decreasing $s$. We remark that the current $i_m$ remains trapped around $iref$ with respect to the tolerance $\Delta$. The values of $u$ switches many times between $-1$ and $+1$; this is due to the fact that the current in the motor increases too quickly (average of $3A$ every $10^{-3}s$). Note that the final speed is not equal to zero because the final time is too short. Moreover, we remark that the curve of the energy decreases at the end of the cycle because this corresponds to the phase of deceleration with a negative period for the reference current $iref$.

## 5. Conclusion

In this paper, we show an original way based on discretization and a Branch and Bound algorithm to solve a hard global optimization problem which is an approximation of an optimal control problem. In a future work, we want to improve the efficiency of our Branch and Bound algorithm. Furthermore, we are interested by the resolution of Problem (3) directly by computing bounds.

## References

[1] A. Sciarreta, L. Guzzella, *Control of Hybrid Electric Vehicles - A Survey of Optimal Energy-Management Strategies*, IEEE Control Systems Magazine, Vol. 27, N. 2, pp. 60–70, 2007.

[2] C. Musardo, G. Rizzoni, Y. Guezennec, B. Staccia, *A-ECMS: An Adaptive Algorithm for Hybrid Electric Vehicle Energy Management*, European Journal of Control, N. 11 (4–5), pp. 509–524, 2005.

[3] J. Bernard, S. Delprat, T.M. Guerra, F. Buechi, *Fuel Cell Hybrid Vehicles: Global Optimization based on Optimal Control Theory*, International Review of Electrical Engineering, 1, 2006.

# Global Optimization of Large-Scale Extended and Generalized Pooling Problems: Mixed-Integer Nonlinearly Constrained Models

Ruth Misener and Christodoulos A. Floudas

*Department of Chemical Engineering, Princeton University, Princeton, NJ,* floudas@titan.princeton.edu

**Abstract** The pooling problem, an optimization challenge of maximizing profit subject product availability, storage capacity, demand, and product specification constraints, has applications to petroleum refining, wastewater treatment, supply-chain operations, and communications. Our recent work studies two classes of pooling problems: (i) an extended pooling problem that incorporates the Environmental Protection Agency (EPA) *Title 40 Code of Federal Regulations Part 80.45: Complex Emissions Model* into the constraint set and (ii) a generalized pooling problem that treats the network topology as a decision variable.

**Keywords:** pooling problem, quadratically-constrained quadratic program, EPA Complex Emissions Model

## 1. Introduction

Allocating limited resources in process synthesis is a major problem that is best approached using recent theoretical advances in global optimization. In the past year, we studied *pooling problems* to minimize cost while complying with environmental standards in petroleum refining and wastewater treatment. Specifically, we addressed (i) an extended pooling problem that incorporates the Environmental Protection Agency (EPA) *Title 40 Code of Federal Regulations Part 80.45: Complex Emissions Model* into the constraint set and (ii) a generalized pooling problem that treats the network topology as a decision variable.

The backbone of both these classes is the *standard* pooling problem, where flow rates on a predetermined network structure of feed stocks, pooling tanks, and final products are optimized to maximize profit subject to quality constraints on the final product composition [5, 13, 22, 23]. Nonconvex bilinear terms arise in the pooling problem from tracking the concentration of key components or *qualities* when multiple streams mix in intermediate storage nodes under the assumption of linear blending. The standard pooling problem is a non-convexifiable Quadratically-Constrained Quadratic Program (QCQP).

## 2. Extended Pooling Problems with the EPA Complex Emissions Model

Environmental Protection Agency (EPA) *Title 40 Code of Federal Regulations Part 80.45: Complex Emissions Model* codifies a mathematical model of volatile organic, nitrous oxide ($NO_X$), and airborne toxic emissions gasoline emissions for reformulated gasoline (RFG) using a function of eleven fuel qualities. The RFG program, which impacts roughly 75 million people, was developed to reduce smog and airborne toxic pollutants in accordance with the Clean Air Act. Final products exiting an oil refinery must comply with these standards, or upper bounds, on volatile organic, nitrous oxide ($NO_X$), and airborne toxic emissions. The *extended pooling problem* appends the EPA Complex Emissions Model and associated constraints to a standard pooling problem. The goal is to comply with RFG standards while maximizing profitability.

*Table 1.* Overview of the Three Case Studies

| | # Variables | | # Nonlinear Terms | |
|---|---|---|---|---|
| | Contin | Binary | Bilinear Only | All |
| Case 1 | 214 | 30 | 62 | 108 |
| Case 2 | 331 | 45 | 111 | 180 |
| Case 3 | 1104 | 150 | 410 | 640 |

The extended pooling problem is a mixed-integer nonlinear model (MINLP) [14], and the nonconvexities consist of bilinear, multilinear, exponential, and power law terms that participate in the constraints. The portion of the MINLP representing the EPA Complex Emissions Model is similar to that of Furman and Androulakis [6] [6]. We also developed a linear relaxation of the MINLP using piecewise-linear [4, 7–9, 12, 16, 17, 24] and edge-concave [11, 20, 21] relaxations. We integrated these relaxations into a branch-and-bound algorithm and solved the three test cases summarized in Table 1 to global optimality.

## 3. Large-Scale Generalized Pooling Problems

The *generalized pooling problem* increases the complexity of the pooling problem by transforming the network topology into a decision variable [2, 12, 15]. Choosing the interconnections between process units and storage tanks, or *pools*, is combinatorially complex. Because the activation or deactivation of each pipe or intermediate node is a discrete decision and the linear mixing at the intermediate nodes leads to bilinear terms, the generalized pooling problem is a mixed-integer nonconvex program (nonconvex MINLP) with quadratic equalities and inequalities which exhibits multiple locally optimal solutions. The major challenge in this problem is developing of rigorous global optimization methods that address large scale problems to global optimality.

The generalized pooling problem was introduced by Audet at al. [2] [2], who addressed the problem using the algorithm proposed in [1]. We revisited the generalized pooling problem test cases of Meyer and Floudas [12] [12, 15]. These test cases posit a set of wastewater sources containing regulated qualities that must be treated before release into the environment [3]. Representing the challenges that industry faces, we then considered as many as twenty treatment options and allowed the possibility of interconnections between all the treatment plants. We exploited recent advances in piecewise-linear underestimation of bilinear terms [4, 7–9, 12, 16, 17, 24] within a branch-and-bound algorithm and globally optimized these test cases. Table 2 summarizes the specific sizes of the four test cases we addressed [15].

To design a good global optimization algorithm for this class of pooling problems, we explored strategies including McCormick convex/concave envelopes [10], reformulation-linearization technique (RLT) relaxations [18, 19], piecewise-linear underestimators [7], branching strategies, and bounds tightening. We made significant computational improvements over previous work by using our proposed alternative formulation, piecewise underestimators, and a good branch-and-bound algorithm [12]. The best previous result reaches a 1.2% optimality gap in 1561.6 seconds when run on a modern hardware and software, but the new results reach a 0.1% optimality gap in as few as 38.2 seconds (a forty-fold time improvement to get an optimality gap that is more than an order of magnitude tighter).

The four-plant test case, with 150 equations, 63 continuous variables, 55 binary variables, and 48 bilinear terms, was solved to a 0.1% gap in 38.25 seconds. The ten-plant instance,

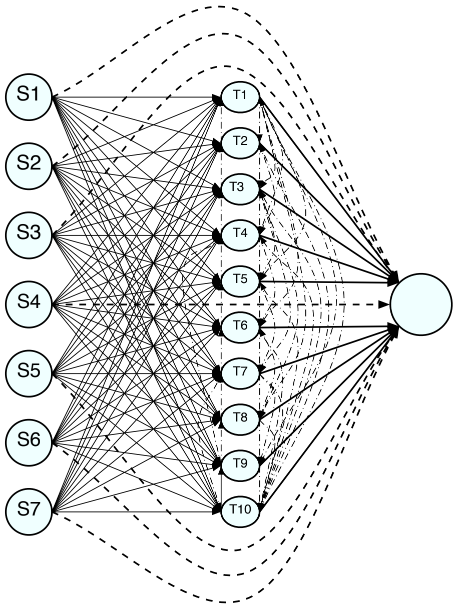


*Figure 1.* Superstructure of the Mid-Size (10-Plant) Generalized Pooling Test Case

*Table 2.* Sizes of the Four Test Cases

| | # Eqs | # Variables | | # Bilinear |
|---|---|---|---|---|
| | | Contin. | Binary | Terms |
| 4-Plant | 150 | 63 | 55 | 48 |
| 10-Plant | 516 | 207 | 187 | 300 |
| 15-Plant | 986 | 382 | 352 | 675 |
| 20-Plant | 1663 | 634 | 594 | 1260 |

with 516 equations, 207 continuous variables, 187 binary variables, and 300 bilinear terms, was solved in 680.66 seconds. We were also able to globally optimize a fifteen-plant test case with 986 equations, 382 continuous variables, 352 binary variables and 675 bilinear terms in 2489.76 seconds on a desktop workstation or 784.81 seconds on an eight-threaded parallel processor. The industrially-sized 15-Plant test case was solved to a 0.1% optimality gap after some experience with the particular topology by predicting a good partitioning level for the problem. For other topologies, we suggest that similar experience with small instantiations of a problem may lead to insight in dealing with industrially-sized cases. Finally, we were able to address two exceptionally large 20-Plant test instances with 1663 equations, 634 continuous variables, 594 binary variables, and 1260 bilinear terms within 0.9% and 2.3% of optimality.

## 4. Summary

We have recently addressed two major classes of pooling problems: extended pooling problems that incorporate the EPA Complex Emissions Model into the constraint set and generalized pooling problems that treat the network topology as a decision variable. These large-scale problems are best approached using rigorous global optimization methods [14, 15].

# References

[1] C. Audet, P. Hansen, B. Jaumard, G. Savard. A branch and cut algorithm for nonconvex quadratically constrained quadratic programming. *Math. Program.*, 87(1): 131–152, 2000.

[2] C. Audet, J. Brimberg, P. Hansen and S. Le Digabel, and N. Mladenovic. Pooling Problem: Alternate Formulations and Solution Methods. *Manage. Sci.*, 50(6): 761–776, 2004.

[3] M. Bagajewicz. A review of recent design procedures for water networks in refineries and process plants. *Comput. Chem. Eng.*, 24: 2093–2113, 2000.

[4] M. L. Bergamini, I. E. Grossmann, N. Scenna, and P. Aguirre. An improved piecewise outer-approximation algorithm for the global optimization of MINLP models involving concave and bilinear terms. *Comput. Chem. Eng.*, 32(3): 477–493, 2008.

[5] C. A. Floudas. Deterministic Global Optimization: Theory, Methods and Applications. Nonconvex Optimization and Its Applications, Kluwer Academic Publishers, Dordrecht, Netherlands, 2000.

[6] K. C. Furman and I. P. Androulakis. A novel MINLP-based representation of the original complex model for predicting gasoline emissions. *Comput. Chem. Eng.*, 32: 2857–2876, 2008.

[7] C. E. Gounaris, R. Misener, and C. A. Floudas. Computational Comparison of Piecewise-Linear Relaxations for Pooling Problems. *Ind. Eng. Chem. Res.*, 48(12): 5742–5766, 2009.

[8] M. M. F. Hasan and I. A. Karimi. Piecewise linear relaxation of bilinear programs using bivariate partitioning. *AIChE J.* Forthcoming (DOI: 10.1002/aic.12109), 2010.

[9] R. Karuppiah and I. E. Grossmann. Global Optimization for the Synthesis of Integrated Water Systems in Chemical Processes. *Comput. Chem. Eng.*, 30: 650-673, 2006.

[10] G. P. McCormick. Computability of global solutions to factorable nonconvex programs: Part 1-convex underestimating problems. *Math. Program.*, 10(1): 147–175, 1976.

[11] C. A. Meyer and C. A. Floudas. Convex envelopes for edge-concave functions. *Math. Program.*, 103(2): 207–224, 2005.

[12] C. A. Meyer and C. A. Floudas. Global Optimization of a Combinatorially Complex Generalized Pooling Problem. *AIChE J.*, 52(3): 1027–1037, 2006.

[13] R. Misener and C. A. Floudas. Advances for the Pooling Problem: Modeling, Global Optimization, and Computational Studies. *Applied and Computational Mathematics*, 8(1): 3–22, 2009.

[14] R. Misener, C. E. Gounaris, and C. A. Floudas. Mathematical modeling and global optimization of large-scale extended pooling problems with the (EPA) complex emissions constraints. *Comput. Chem. Eng.*, Forthcoming (DOI: 10.1016/j.compchemeng.2010.02.014), 2010.

[15] R. Misener and C. A. Floudas. Global Optimization of Large-Scale Generalized Pooling Problems: Quadratically Constrained MINLP Models. Submitted, 2010.

[16] V. Pham and C. Laird and M. El-Halwagi. Convex Hull Discretization Approach to the Global Optimization of Pooling Problems. *Ind. Eng. Chem. Res.*, 48: 1973–1979, 2009.

[17] Y. Saif, A. Elkamel, M. Pritzker. Global optimization of reverse osmosis network for wastewater treatment and minimization. *Ind. Eng. Chem. Res.*, 47(9): 3060–3070, 2008.

[18] H. D. Sherali and W. P. Adams. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*. Nonconvex Optimization and Its Applications. Kluwer Academic Publishers, Dordrecht, Netherlands, 1999.

[19] H. D. Sherali and A. Alameddine. A new reformulation-linearization technique for bilinear programming problems. *J. Global Optim.*, 2:379–410, 1992.

[20] F. Tardella. On a class of functions attaining their maximum at the vertices of a polyhedron. *Discret. Appl. Math.*, 22: 191–195, 1988/89.

[21] F. Tardella. On the existence of polyhedral convex envelopes. In C. A. Floudas and P. M. Pardalos, editors, *Frontiers in Global Optimization*, pages 563–573. Kluwer Academic Publishers, 2003.

[22] M. Tawarmalani and N. V. Sahinidis. Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Applications, Software, and Applications. Nonconvex Optimization and Its Applications, Kluwer Academic Publishers, Norwell, MA, 2002.

[23] V. Visweswaran. MINLP: Applications in Blending and Pooling. In C. A. Floudas and P. M. Pardalos, editors, *Encyclopedia of Optimization, Second Edition*, Springer Science, 2114–2121, 2009.

[24] D. S. Wicaksono and I. A. Karimi. Piecewise MILP under-and overestimators for global optimization of bilinear programs. *AIChE J.*, 54(4): 991–1008, 2008.

# Strategies for Solving Distance Geometry Problems with Inexact Distances by Discrete Approaches

Antonio Mucherino,[1] Carlile Lavor,[2] Leo Liberti,[3] and Nelson Maculan[4]

[1] *INRIA Lille Nord Europe, France,* antonio.mucherino@inria.fr

[2] *IMECC, UNICAMP, Campinas, Brazil,* clavor@ime.unicamp.br

[3] *LIX, Ecole Polytechnique, Palaiseau, France,* liberti@lix.polytechnique.fr

[4] *COPPE, Federal University of Rio de Janeiro, Brazil,* maculan@cos.ufrj.br

**Abstract** We consider the distance geometry problem in the particular case in which it can be discretized and formulated as a combinatorial optimization problem. A Branch & Prune (BP) algorithm is employed for an efficient solution of this problem when exact values for the known distances are available. We discuss some strategies for extending the BP algorithm to the case in which inexact distances are instead provided.

**Keywords:** distance geometry, discrete formulation, protein conformation, Branch & Prune, inexact distances

## 1. Introduction

The Distance Geometry Problem (DGP) is the problem of finding the coordinates of a set of points from some relative distances between such points. This problem arises in many practical applications. In particular, we are interested in DGPs arising in biology, where the conformation of a molecule must be determined by exploiting some known distances between pairs of its atoms. In the scientific literature, the DGP related to molecules is usually referred to as the Molecular DGP (MDGP). MDGPs are rather difficult, because the necessary distances are obtained through experimental techniques such as the Nuclear Magnetic Resonance (NMR), which are able to provide only a limited information on the distances.

The MDGP is a constraint satisfaction problem usually reformulated as a global optimization problem, where a penalty function, measuring the satisfaction of the constraints on the distances, needs to be minimized. When some assumptions are satisfied [2], the domain of the penalty function becomes discrete, and, in particular, it can be seen as a binary tree containing positions for the atoms of the considered molecule. In this case, we refer to the problem as the Discretizable MDGP (DMDGP). Both the MDGP and the DMDGP are NP-hard [1, 5].

The DMDGP is a combinatorial optimization problem. To solve instances of this problem, we employ the Branch & Prune (BP) algorithm [2], which is strongly based on the binary tree structure of the penalty function domain. At each iteration, two new nodes of the tree are added, which represent two new positions for the current atom $x_i$. Then, the feasibility of the two positions is checked, and branches of the tree containing infeasible positions are pruned. This pruning phase allows for reducing the binary tree very quickly, and for solving the DMDGP in a reasonable amount of time.

In this short paper, we investigate the possibility of solving DMDGPs in which a lower and an upper bound on the distances are provided instead of an exact value. This assumption is realistic, because the information on the distances are supposed to come from experimental techniques that are not able to provide very precise information. Preliminary studies in this direction were presented in [3].

The rest of the paper is organized as follows. In Section 2 we will discuss the main idea behind the discretization of the DMDGP and the BP algorithm, while in Section 3 we will provide some possible solutions for extending such an algorithm in the case in which exact distances are not available. Conclusions are given in Section 4.

## 2. The DMDGP with exact distances

Let $G = (V, E)$ be an undirected graph, where vertices in $V$ correspond to the atoms of the considered molecule, and there is an edge between two vertices if and only if the corresponding distance is known. Weights associated to edges can provide the numerical value of the known distances. The discretization of the MDGP is possible when the following two assumptions are satisfied, for a given ordering on $V$:

**Ass. a1** for each atom $x_i \in V$ with rank $i > 3$, the distances between $x_i$ and the three preceding atoms $x_{i-1}$, $x_{i-2}$ and $x_{i-3}$ must be known; in other words, the set of edges $E$ must contain $(i-1, i)$, $(i-2, i)$ and $(i-3, i)$;

**Ass. a2** for each triplet of consecutive atoms $x_i$, $x_{i-1}$ and $x_{i-2}$, the triangular inequality on the corresponding distances must hold strictly:

$$d_{i-2,i} < d_{i-2,i-1} + d_{i-1,i}.$$

Let $H$ be the set of pairs of atoms whose distances must be known because of Assumption a1. Other distances that are not required by this assumption could however be available. Let $F = E - H$ be the subset containing such distances.

The idea behind the discretization is the following. Let us suppose that all the atoms with rank smaller than $x_i$ have been already placed somewhere and that a position for the atom $x_i$ needs to be found. By Assumption a1, the distances $d_{i-3,i}$, $d_{i-2,i}$, $d_{i-1,i}$ are known. As a consequence, three spheres having center in $x_{i-3}$, $x_{i-2}$ and $x_{i-1}$ and radius $d_{i-3,i}$, $d_{i-2,i}$ and $d_{i-1,i}$, respectively, can be defined. By Assumption a2, the intersection of these three spheres can result in one single point but more likely in two different points: there are at most two possible atomic positions for the atom $x_i$. If this procedure is applied recursively for all the atoms of the molecule, a binary tree of atomic positions can be built and explored for finding solutions to the DMDGP.

The basic idea behind the BP algorithm is to compute, at each iteration, the two possible positions for the current atom $x_i$. Moreover, the distances associated to the pairs of atoms in $F$ are exploited for checking the feasibility of the computed positions by verifying that, for all $j < i - 3$ for which the edge $(j, i) \in F$, the following inequalities hold:

$$|\, ||x_j - x_i|| - d_{ji} \,| < \varepsilon, \tag{1}$$

where $\varepsilon$ is a small positive real number. For more details on the discretization and on the BP algorithm, the reader is referred to [1, 2].

## 3. The DMDGP with inexact distances

When inexact distances are available, the BP algorithm, as described in Section 2, cannot be used. An instance of the DMDGP with inexact distances is basically a list of intervals defined by the lower and the upper bound on the actual value of the distances.

The assumptions for the discretization can be easily adapted for the inexact DMDGP. Let $G = (V, E)$ be an undirected graph associated to an instance of the DMDGP with inexact distances. In this case, not only one weight is associated to each edge in $E$, but two weights $l$ and

$u$ representing the lower and the upper bound, respectively, on the corresponding distance. Assumption a1 can remain unchanged, the only difference being that the distances between the generic atom $x_i$ and the three preceding atoms are intervals. Assumption a2 can be instead modified as follows. Let us consider a triplet of consecutive atoms $x_i$, $x_{i-1}$ and $x_{i-2}$, and the three intervals $[l_{i-2,i}, u_{i-2,i}]$, $[l_{i-2,i-1}, u_{i-2,i-1}]$ and $[l_{i-1,i}, u_{i-1,i}]$.

**Ass. b2** there exists a triplet $(c_{i-2,i}, c_{i-2,i-1}, c_{i-1,i})$ such that

$$c_{i-2,i} \in [l_{i-2,i}, u_{i-2,i}], \quad c_{i-2,i-1} \in [l_{i-2,i-1}, u_{i-2,i-1}], \quad c_{i-1,i} \in [l_{i-1,i}, u_{i-1,i}],$$

and

$$c_{i-2,i} < c_{i-2,i-1} + c_{i-1,i}.$$

Note that triplets of distances $(c_{i-2,i}, c_{i-2,i-1}, c_{i-1,i})$ that do not satisfy the triangular inequality should not be considered because they define local subsets of incompatible distances. This property can be exploited for refining the available intervals before solving the inexact DMDGP. In the following we will suppose that all possible triplets $(c_{i-2,i}, c_{i-2,i-1}, c_{i-1,i})$ satisfy the strict triangular inequality.

As previously explained, the distances related to the edges in $H$ and the other distances related to the edges in $F$ are used for different purposes during the execution of the BP algorithm. The distances associated to the pairs of atoms in $F$ are not exploited for discretizing the problem, but rather for checking the feasibility of the atomic positions at each iteration of the algorithm. If intervals are available instead of exact distances [3], the inequalities (1) can then be substituted by:

$$l_{ji} < ||x_j - x_i|| < u_{ji}. \tag{2}$$

Note that no tolerances $\varepsilon$ are needed in this case.

While the inequalities (1) can be trivially substituted by the inequalities (2), the procedure for generating the binary tree cannot be generalized so easily to the case of the inexact DMDGP. Exact distances are not known but rather a lower and an upper bound. By Assumption a1, for each $i \in V$ with rank $i > 3$, the edges $(i-1,i)$, $(i-2,i)$, $(i-3,i) \in E$, and therefore the three intervals $[l_{i-1,i}, u_{i-1,i}]$, $[l_{i-2,i}, u_{i-2,i}]$, $[l_{i-3,i}, u_{i-3,i}]$ are known.

A first approach for building the binary tree from the information on these intervals is to attempt an approximation of the actual distance values. In this way, exact distances can substitute the intervals and the same discretization procedure described in Section 2 can be applied as is. One possibility is to consider the average value defined by each interval:

$$c_{i-1,i} = \frac{l_{i-1,i} + u_{i-1,i}}{2}, \quad c_{i-2,i} = \frac{l_{i-2,i} + u_{i-2,i}}{2}, \quad c_{i-3,i} = \frac{l_{i-3,i} + u_{i-3,i}}{2},$$

and use these three values as radii for the three intersecting spheres. However, we noted that small errors introduced on the distances in $H$ are able to propagate very quickly along the branches of the tree and to spoil the experiments. Therefore, even though the average of an interval is the distance that statistically better represents the interval, we must be aware that the actual distance could be quite far from this average value, and that the maximum error that might occur is:

$$\max_{k=1,2,3} \{c_{i-k,i} - l_{i-k,i}, u_{i-k,i} - c_{i-k,i}\}.$$

Naturally, the larger is the length of the intervals, the larger the occurring error can be.

Instead of approximating the intervals related to $H$ with their average value only, other randomly chosen values can also be selected. This strategy has been already used in other approaches to distance geometry [4]. It has been applied to all the intervals defining an instance, whereas we need to apply it only to the intervals related to the subset $H$. By using this strategy, more than one sequence of exact distances can be associated to the original set of intervals. Since distances are randomly chosen, a heuristic component is in this way added to our BP algorithm.

It is important to note that, when replacing the set of original distances by a sequence of discrete values, we need to check if the triangular inequality is still satisfied for all the triplets of atoms. Distances can be modified in order to have all the inequalities satisfied, but, when they are modified, they should not be allowed to get out of their own interval.

We implemented this strategy and we tested it on a set of artificially generated instances. The BP algorithm was able to find the solutions related to the generated sequences of exact distances only in correspondence with the smallest instances. For larger instances, the errors introduced on the binary tree by picking random representatives for the intervals propagated along the tree and made it incompatible with the intervals in $F$, so that the whole tree was pruned. We noted that the probability of catching a sequence of exact distances for which the binary tree is compatible to the intervals in $F$ is quite low, and therefore generating a larger number of such sequences does not solve the problem.

Therefore, other efficient strategies for the DMDGP need to be developed. Approximating the set of intervals in $H$ with various sequences of randomly selected distances does not seem to be a good approach in our case, even though it has been widely used in other approaches to distance geometry. As a consequence, we should probably avoid to introduce approximations and consider the entire intervals in $H$. This brings to the formulation of the following problem. As already remarked, Assumption a1 ensures, in the case of the inexact DMDGP, that all the intervals $[l_{i-1,i}, u_{i-1,i}]$, $[l_{i-2,i}, u_{i-2,i}]$, $l_{i-3,i}, u_{i-3,i}]$ are known. Therefore, three spherical shells having center in $x_{i-3}$, $x_{i-2}$ and $x_{i-1}$, inner radius $l_{i-3,i}$, $l_{i-2,i}$ and $l_{i-1,i}$ and outer radius $u_{i-3,i}$, $u_{i-2,i}$ and $u_{i-1,i}$, respectively, can be defined. If the lengths of the intervals were infinitesimal, the intersection of the three spherical shells would be at most two points by Assumption b2. Since, in general, the lengths of the intervals are not infinitesimal, such intersection must be formed by two three-dimensional geometrical objects, which are probably disjoint, and where the actual positions for $x_i$ are contained. In this case, then, two geometrical objects should be associated to the two new nodes that are added to the binary tree, and not two single positions. However, to the best of our knowledge, the shape and the mathematical expressions of these two objects are unknown.

## 4. Conclusions

We discussed possible strategies for solving DMDGPs with inexact distances. Unfortunately, we are not able yet to provide a good solution for this problem: we presented some subproblems that, if solved, could help solving DMDGPs in the case in which lower and upper bounds on the distances are available. Particularly interesting is the following problem: given three spherical shells, how to represent mathematically the two geometrical objects that their intersection defines?

## References

[1] C. Lavor, L. Liberti, and N. Maculan, *Discretizable Molecular Distance Geometry Problem*, Tech. Rep. q-bio.BM/0608012, arXiv, 2006.

[2] 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), 1–17, 2008.

[3] A. Mucherino, C. Lavor, *The Branch and Prune Algorithm for the Molecular Distance Geometry Problem with Inexact Distances*, Proceedings of World Academy of Science, Engineering and Technology **58**, International Conference on Bioinformatics and Biomedicine (ICBB09), Venice, Italy, 349–353, 2009.

[4] R. Reams, G. Chatham, W. Glunt, D. McDonald, T. Hayden, *Determining Protein Structure using the Distance Geometry Program APA*, Computers & Chemistry **23**, 153–163, 1999.

[5] J.B. Saxe, *Embeddability of Weighted Graphs in $k$-space is Strongly NP-hard*, Proceedings of $17^{th}$ Allerton Conference in Communications, Control, and Computing, Monticello, IL, 480–489, 1979.

# VXQR: Derivative-free unconstrained optimization based on QR factorizations

Arnold Neumaier, Hannes Fendl, Harald Schilly, and Thomas Leitner

*Fakultät für Mathematik, Universität Wien Nordbergstr. 15, A-1090 Wien, Austria,* Arnold.Neumaier@univie.ac.at

**Abstract** This paper presents basic features of a new family of algorithms for unconstrained derivative-free optimization in high dimensions, based on line searches along directions generated from QR factorizations of past direction matrices. Emphasis is on fast descent with a low number of function values, so that the algorithm can be used for fairly expensive functions. The theoretical total time overhead needed per function evaluation is of order $O(n^2)$, where $n$ is the problem dimension, but the observed overhead is much smaller.

The particular algorithm tested is typically very fast on smooth problems with not too rugged graphs, and on problems with a roughly separable structure. It typically performs poorly on problems where the graph along many directions is highly multimodal without pronounced overall slope (e.g., for smooth functions with superimposed oscillations of significant size), where the graphs along many directions are piecewise constant (e.g., for problems minimizing a maximum norm), or where the function overflows on the major part of the search region and no starting point with finite function value is known.

**Keywords:** Derivative-free optimization, black box optimization, scalability, high-dimensional, global optimization, line search, expensive objective function

## 1. Introduction

In derivative-free optimization (also known as black-box optimization), the goal is to optimise a function defined on a subset of $\mathbf{R}^n$ for which derivative information is neither symbolically available nor numerically computable, and bounds on Lipschitz constants are not known. More specifically, we consider the unconstrained optimization problem of minimizing a real-valued function $f$ defined on a subset of $\mathbf{R}^n$ by an oracle that returns for a given $x \in \mathbf{R}^n$ the function value $f(x)$ if it is defined, and otherwise one of the surrogate values `inf` or `NaN`. In addition, scaling information is assumed to be available in the form of a search box The expectation (which may or may not be true) is that the minimizer of interest lies in the interior of the search box; but the search may lead out of the box. The best point found is therefore not guaranteed to lie in the box.

VXQR stands for *valley exploration based on QR factorizations*, emphasizing the linear algebra tool (QR factorization) that dominates the part of the execution cost of the algorithms that is independent of the cost of the function evaluation routine.

The VXQR algorithm is designed for the case when a call to the oracle is fairly expensive, so that one wants to keep the number of function evaluations needed as small as possible, aiming for a rapid decrease of the objective function rather than for necessarily reaching the global minimum. Such a fast decrease is achievable only if the dimension of the problem is small, or if the function to be optimized has an appropriate structure that can be exploited by the algorithms. In particular, our algorithms are based on the expectation that the objective function resembles a separable function in so far as the variation of single coordinates is a reasonable option. Another expectation is that the function is smooth (twice continuously differentiable) and not strongly oscillating along typical rays. This enables one to make fast progress using line searches, a basic tool from traditional deterministic optimization algorithms. However, we also exploit the robustness that can be gained from a stochastic approach, and thus combine deterministic and stochastic features in a novel way. Thus none of the structural expectations is essential for the algorithm to work, although the performance of problems not matching the stated expectations may be erratic or poor.

The VXQR class of algorithms we consider proceed in several phases. After the initial scaling phase 1, scout phase 2 and subspace phase 3 alternate until a stopping criterion is reached.

Phase 1 **(scaling phase)**: Search for a well-scaled initial point $x$ with finite $f(x)$. This is simply done by a uniform random search in the specified search box, followed by a line search in the direction of the point in the search region with the absolutely smallest components to get an initial point with an appropriate order of magnitude.

Phase 2 **(scout phase)**: Search for a direction of good expected progress. This is done by a sequence of line searches from the best point, either in coordinate directions (to be efficient for approximately separable problems), or (to be efficient for nonseparable smooth problems) in a direction chosen from an orthonormal basis that adapts itself during the course of the algorithm. Line searches are done by function evaluation along a simple 1-dimensional grid, followed by safeguarded parabolic interpolation steps to approximately locate local minima along the direction searched. This is the place where the algorithm expects that the objective function is smooth; performance is degraded if this is not the case. The orthonormal basis is created at the beginning of each scout phase by taking the columns of the orthogonal factor $Q$ of a QR factorization of the matrix formed by the differences of the best point from the results of the scout line searches of the previous scout phase, but chosen randomly before the first scout phase.

Phase 3 **(subspace phase)**: The direction generated by the scout phase is used to extend a low-dimensional search subspace until a saturation dimension (in VXQR1, this dimension is 10 when $n > 20$) is reached; in this case, the scout direction replaces an old subspace direction. In the new subspace, a local quadratic model is created and minimized, subject to some safeguards. Then a line search is performed from the best point found so far to the model minimizer.

Because all line searches start from the best point available at the time, the algorithms have a greedy tendency. This makes them efficient for a low number of function values, but puts them at a disadvantage for highly oscillating functions where the greedy strategy often confines progress to a small neighborhood of the best point, with escape to a different valley often being the result of pure chance rather than strategy.

The work outside the function evaluations is dominated by the cost of the QR factorizations, the only expensive step in the whole procedure. Since a QR factorization needs $O(n^3)$ arithmetic operations, and since after each QR factorization the scout search takes $O(n)$ but at least $3n$ function evaluations, the total time overhead needed per function evaluation is of order $O(n^2)$. But the observed overhead is much smaller.

We compared a particular implementation VXQR1 of the VXQR scheme with the differential evolution algorithm DE of STORN & PRICE [3] and the covariance matrix adapted evolutionary strategy G_CMA-ES by AUGER & HANSEN [1, 2] on a number of problems of dimension 50, 100, 200, 500, and 1000. Our tentative conclusion (as far as the limited testing discussed here allows one) is that VXQR1 compares well with the reference solvers, though not uniformly over all problem types. VXQR1 should be used on smooth problems and approximately separable problems, DE on problems with large oscillations, and G_CMA-ES on problems that minimize the maximum of many functions.

# References

[1] A. Auger and N. Hansen, A restart CMA evolution strategy with increasing population size. pp. 1769–1776 in: The 2005 IEEE Congress on Evolutionary Computation, vol 2 (2005).

[2] N. Hansen, The CMA evolution strategy: A comparing review. pp. 75–102 in: Towards a new evolutionary computation. Advances on estimation of distribution algorithms (J.A. Lozano et al., eds.), Springer 2006.

[3] R. Storn and K. Price, Differential Evolution – a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, Journal of Global Optimization 11 (1997), 341–359.

# A Mixed Affine Reformulation Method for Global Optimization

Jordan Ninin and Frédéric Messine

*ENSEEIHT, IRIT-CNRS, University of Toulouse, France,* Jordan.Ninin@enseeiht.fr, Frederic.Messine@enseeiht.fr

**Abstract** An automatic method for constructing mixed integer linear relaxations of mixed integer non-convex optimization problems is proposed. This method is an extension of the *affine reformulation technique* [4] in order to consider mixed integer problems. The so-generated mixed integer linear program has exactly the same number of variables and of inequality constraints as the given problems. Thus, the resolution wastes less time. This technique is including in an interval branch and bound algorithm to improve the computation of lower bounds.

**Keywords:** interval arithmetic, affine arithmetic, linear relaxation, branch and bound

## 1. Introduction

Since few years, the global optimization of mixed integer non-linear programming (MINLP) problems has made some breakthroughs. It is now possible to consider very complicated large problems. One of these improvements is the growing use of relaxation techniques. The most famous one is the Reformulation-Linearization-Techniques (RLT) introduced by Sherali and Adams which provides linear formulation for non-convex equations. This is performed by introducing a new variable for every non-convex operators [1]. In this paper, the method is based on a relaxation technique using the affine arithmetic, which does not introduce new variable [4]. We will present an extension of this new technique to consider integer variables to reformulate a MINLP problem into a mixed integer linear program (MILP) which keeps the same size as the original one.

## 2. Affine Reformulation Technique (ART)

The *Affine Reformulation Technique* (ART) is a new approach, which generates automatically linear relaxations of non-linear optimization problems [4]. The originality lies in how the linear relaxation is performed. This technique is based on the *affine arithmetic*, which are not longer considered as a simple way to compute bounds but as a way to generate linear relaxations of *factorable functions*, i.e. composed of elementary operators or functions (e.g. $+, -, *, /, \exp, \log, \sqrt{}$,...).

*Affine arithmetic* was introduced in 1993 by Comba and Stolfi, developed by De Figuei-redo and Stolfi in [2] and extended by Messine in 2002 [3]. This arithmetic is based on the same principle as the *interval arithmetic* except that each quantity are represented by an affine form. It was usually used to compute lower and upper bounds of a function over an interval. The original interval are converted into an affine form; the computation is performed using the definitions of the usual operators in affine arithmetic; at the end, the final affine form of the entirely function is converted into an interval to obtain a lower and an upper bounds on the values of the function over the original interval. The main advantage is to keep linear dependency information during the computation, which makes it possible to improve the quality of the bounds.

Messine in [3] has proposed an extended affine arithmetic, named *AF2*. In *AF2*, the number of variable $\epsilon_i$ of the affine form is fixed to the size of the original domain, and all the rounded and approximated errors are added in three error terms, separating the positive, negative and unsigned errors.

The principle of ART is to identify the linear part of the affine form *AF2* as a linear relaxation. Moreover, using the error terms of the affine form *AF2*, we compute the maximal error between the relaxation and the original function. The linear relaxation depends on the considered box, but it has the advantage to generate only one inequation by inequality constraint and two inequations by equality constraint, and without adding new variables, regardless the number of non-linear term in the constraint.

## 3. Mixed integer Affine Reformulation Technique (MART)

This technique is an extension of ART to consider MINLP problem. This one permits to generate a mixed integer linear relaxation.

Considering Messine's affine form *AF2*, the number of variable $\epsilon_i$ is fixed to the size of the original domain. Thus, an affine transformation $T$ between the original set $X \times Y \subset \mathbb{R}^n \times \mathbb{Z}^m$ and $\epsilon = [-1,1]^{n+m}$ is provided. This affine transformation $T$ is the conversion of an interval into an affine form. The mixed affine reformulation technique proceeds as follows:
- The ART generates a linear relaxation of the original problem on a domain under study $X \times Y$. The so-generate linear program is considered on $[-1,1]^{n+m}$.
- The transformation $T^{-1}$ is used to return the integer variables into the original domain. Thus, the domain of the generated linear program is $[-1,1]^n \times Y$.
- Then, it remains only to consider the integer variable of the original problem as integer variables in the linear program.

In Proposition 1, we formulate the process to generate the linear relaxation. It is an extension into the mixed integer case of Proposition 3 of [4]. The transformation $T^{-1}$ are only used on the integer variables.

**Proposition 1.** *Consider* $(f_0,\dots,f_{n+m},f_\pm,f_+,f_-)$ *the reformulation of* $f$ *on* $X \times Y \subset \mathbb{R}^n \times \mathbb{Z}^m$ *using* **AF2**, *thus, if* $\forall (x,y) \in X \times Y, f(x,y) \leq 0$ *then* $\forall (z,y) \in [-1,1]^n \times Y$, $\sum_{i=1}^n f_i z_i + \sum_{i=n+1}^{n+m} \frac{2f_i}{wid(Y_i)} y_i \leq \sum_{i=n+1}^{n+m} \frac{2f_i mid(Y_i)}{wid(Y_i)} + f_\pm + f_- - f_0$ *;*
*and if* $\forall (x,y) \in X \times Y, f(x,y) = 0$ *then* $\forall (z,y) \in [-1,1]^n \times Y$, $\sum_{i=1}^n f_i z_i + \sum_{i=n+1}^{n+m} \frac{2f_i}{wid(Y_i)} y_i \leq \sum_{i=n+1}^{n+m} \frac{2f_i mid(Y_i)}{wid(Y_i)} + f_\pm + f_- - f_0$ *and* $-\sum_{i=1}^n f_i z_i - \sum_{i=n+1}^{n+m} \frac{2f_i}{wid(Y_i)} y_i \leq \sum_{i=n+1}^{n+m} \frac{-2f_i mid(Y_i)}{wid(Y_i)} + f_\pm + f_+ + f_0$.

*Proof.* Denote by $\widehat{f}(x,y)$ the affine form AF2 of $f$ on $X \times Y \subset \mathbb{R}^n \times \mathbb{Z}^m$. Here the components $f_i$ in the formulation depend also on $X \times Y$. For $i \in \{1,\dots,n\}$, the variables $\epsilon_i$ are associated to the interval $X_i \subset \mathbb{R}$ and for $i \in \{1,\dots,m\}$, $\epsilon_{n+i}$ are associated to $Y_i \subset \mathbb{Z}$.

$$\widehat{f}(x,y) = f_0 + \sum_{i=1}^{n} f_i\epsilon_i + \sum_{i=n+1}^{n+m} f_i\epsilon_i + f_\pm\epsilon_\pm + f_+\epsilon_+ + f_-\epsilon_-,$$
$$\text{with } \forall i \in \{0,\dots,n+m\}, f_i \in \mathbb{R}\ , f_\pm, f_+, f_- \in \mathbb{R}^+,$$
$$\forall i \in \{1,\dots,n+m\}, \epsilon_i = [-1,1] \text{ and } \epsilon_\pm = [-1,1], \epsilon_+ = [0,1], \epsilon_- = [-1,0].$$

By definition, the affine form AF2 is an inclusion function:

$$\forall (x,y) \in X \times Y,\ f(x,y) \in f_0 + \sum_{i=1}^{n+m} f_i\epsilon_i + f_\pm\epsilon_\pm + f_+\epsilon_+ + f_-\epsilon_-.$$

But $\forall z \in [-1,1]^n, \exists x \in X, \forall i \in \{1,\dots,n\}, z_i = T_i(x_i) = \frac{2}{wid(X_i)}(x_i - mid(X_i))$, where $wid$ represents the witdh of an interval and $mid$ its middle. Similarly, there is an affine transformation between $Y$ and $[-1,1]^m$, then, $\forall (x,y) \in X \times Y$, we have:

$$f(x,y) \in \left(\sum_{i=1}^{n} f_i T_i(x_i) + \sum_{i=n+1}^{n+m} f_i T_i(y_i) + f_0 + f_\pm[-1,1] + f_+[0,1] + f_-[-1,0]\right),$$

$$f(x,y) - \sum_{i=1}^{n} f_i T_i(x_i) - \sum_{i=n+1}^{n+m} f_i T_i(y_i) \in [f_0 - f_{n+1} - f_{n+3}, f_0 + f_{n+1} + f_{n+2}],$$

$$f(x,y) - \sum_{i=1}^{n} f_i T_i(x_i) - \sum_{i=n+1}^{n+m} \frac{2f_i}{wid(Y_i)} y_i + \sum_{i=n+1}^{n+m} \frac{2f_i mid(Y_i)}{wid(Y_i)} \in [f_0 - f_\pm - f_-, f_0 + f_\pm + f_+].$$

Proposition 1 follows directly from the application of this last equation. □

**Example 2.** *Considering the equation $\forall (x,y) \in [0,1] \times (\mathbb{Z} \cap [0,4]), x \times y = 0$. the first step consists to reformulate $[0,1]$ and $[0,4]$ into affine forms AF2. Then, the multiplication between two affine forms is computed and finally, using Proposition 1, two linear inequalities are generated:*

$$X = [0,1] \to \widehat{x} = 0.5 + 0.5\epsilon_1 \text{ and } Y = [0,4] \to \widehat{y} = 2 + 2\epsilon_2,$$
$$\widehat{x} \times \widehat{y} = 1 + \epsilon_1 + \epsilon_2 + \epsilon_\pm,$$
$$\Rightarrow \forall (z,y) \in [-1,1] \times \{0,4\}, \begin{cases} z + 0.5y \leq 1 \\ -z - 0.5y \leq 1 \end{cases}$$

*We can remark that $\forall z \in [-1,1], (z,0)$ and $\forall y \in \mathbb{Z} \cap [0,4], (-1,y)$ are solutions of the system of inequations, which implies that $\forall x \in [0,1], (x,0)$ and $\forall y \in \mathbb{Z} \cap [0,4], (0,y)$ are solutions of the original system.*

Consider the MINLP problem (P1). Using Proposition 1, we can construct a MILP problem (P2) which is a linear relaxation of (P1):

$$\begin{cases} \min\limits_{(x,y)\in X\times Y \subset \mathbb{R}^n \times \mathbb{Z}^m} & f(x,y) \\ \text{s.t.} \quad \forall k \in \{1,p\}, & g_k(x,y) \leq 0 \\ \qquad \forall l \in \{1,q\}, & h_l(x,y) = 0 \end{cases} \quad \text{(P1)} \qquad \begin{cases} \min\limits_{(z,y)\in[-1,1]^n \times Y} & c^T z + d^T y \\ \text{s.t.} & A \begin{pmatrix} z \\ y \end{pmatrix} \leq b \end{cases} \quad \text{(P2)}$$

Let denote by $(F_0, \dots, F_{n+3})$ the resulting affine form **AF2** of $F$ such as $\widehat{F}(x) = F_0 + \sum_{i=1}^{n+m} F_i \epsilon_i + F_\pm \epsilon_\pm + F_+ \epsilon_+ + F_- \epsilon_-$, then the linear program (P2) is constructed as follows:

$$c = (f_1, \dots, f_n) \qquad\qquad d = \left(\frac{2f_{n+1}}{wid(Y_1)}, \dots, \frac{2f_{n+m}}{wid(Y_n)}\right)$$

$$A = \begin{pmatrix} (g_k)_1 & \dots & (g_k)_n & \frac{2(g_k)_{n+1}}{wid(Y_1)} & \dots & \frac{2(g_k)_{n+m}}{wid(Y_n)} \\ \vdots & & \vdots & \vdots & & \vdots \\ (h_l)_1 & \dots & (h_l)_n & \frac{2(h_l)_{n+1}}{wid(Y_1)} & \dots & \frac{2(h_l)_{n+m}}{wid(Y_n)} \\ -(h_l)_1 & \dots & -(h_l)_n & \frac{-2(h_l)_{n+1}}{wid(Y_1)} & \dots & \frac{-2(h_l)_{n+m}}{wid(Y_n)} \\ \vdots & & \vdots & \vdots & & \vdots \end{pmatrix}$$

$$b = \begin{pmatrix} \sum_{i=1}^{m} \frac{2(g_k)_{n+i} mid(Y_i)}{wid(Y_i)} + (g_k)_\pm + (g_k)_- - (g_k)_0 \\ \vdots \\ \sum_{i=1}^{m} \frac{2(h_l)_{n+i} mid(Y_i)}{wid(Y_i)} + (h_l)_\pm + (h_l)_- - (h_l)_0 \\ -\sum_{i=1}^{m} \frac{2(h_l)_{n+i} mid(Y_i)}{wid(Y_i)} + (h_l)_\pm + (h_l)_+ + (h_l)_0 \\ \vdots \end{pmatrix}$$

By solving the MILP problem (P2), we can deduce that if (P2) have no feasible solution, the problem (P1) has no feasible solution too. If a feasible solution of (P2) exists, a lower bound of (P1) can be computed using the following proposition.

**Proposition 3.** *If $(z_{sol}, y_{sol})$ is a solution which minimizes the mixed integer linear program (P2) and $S$ the set of feasible solutions of the problem (P1), then*

$$\forall x \in S\ ,\ f(x) \geq c^T z_{sol} + d^T y_{sol} + f_0 - f_{\pm} - f_{-} - \sum_{i=1}^{m} \frac{2f_{n+i} mid(Y_i)}{wid(Y_i)}.$$

This new relaxation technique generates small mixed integer linear programs. Indeed, the number of variables of the relaxation is the same as the original problem and the number of constraints is at most twice. This technique is including inside an interval branch and bound algorithm. The main idea of this technique is to generate small MILP problems on each subdomain, which are quickly solved to improve lower bounds and to eliminate boxes which do not contain the global minimum. Thus, the technique is used in a context when it is better to subdivide the domain than to add more linear cutting plans to improve the linear relaxation.

## 4. Numerical Examples

To compare ART and MART techniques, we include them in an interval branch and bound algorithm. Necessarily, the algorithm using MART should visit less nodes than the ART based method. This is simply due to to the fact that ART and MART generate the same linear program. The only difference is that in MART, we center the integer variable to produce a MILP problem. But the resolution of a MILP problem could waste much time than the resolution of the LP problem which relaxes the integer variables. Indeed, most of time the resolution of a MILP problem include another branch and bound algorithm. Thus, we have now a branch and bound algorithm at each iteration of our branch and bound algorithm and not simply a resolution via a simplex or interior point code.

To illustrate, we apply the algorithm on a non-convex problem from the web library 1 COCONUT [5], named $ex7\_2\_1$, with 7 variables and 14 constraints. And we modify it by taking into account the variables $x_1, x_2, x_4$ and $x_5$ as integer variables. The tests are done on a PC-Intel-Xeon-3GHz computer with 2GB of RAM and using a 64-bit Linux system. The code are written in Fortran 90 using the SUN fortran compiler. C-PLEX 11.0 is used to solve the MILP and the LP problems. The algorithm with MART solves $ex7\_2\_1$ in **9,046** iterations with **43** seconds; The algorithm with ART solves it in **45,598** iterations with **49** seconds. Indeed, we divide that number of iterations by 5 but that takes less than 5 time more time per iterations.

This first result seems to show that for bigger and more complicated problems, MART could have a big impact in branch and bound algorithms.

## References

[1] W.P. Adams and H.D. Sherali, *A reformulation-linearization technique for solving discrete and continuous nonconvex problems*, Nonconvex Optimization and Its Applications , Vol. 31, Kluwer Academic Publishers, 1998.

[2] L. De Figueiredo and J. Stolfi, *Self-Validated Numerical Methods and Applications*, Brazilian Mathematics Colloquium monographs, IMPA/CNPq, Brazil, 1997.

[3] F. Messine, *Extentions of affine arithmetic: Application to unconstrained global optimization*, Journal of Universal Computer Science, Vol. 8, pp. 992–1015, 2002.

[4] J. Ninin, F. Messine and P. Hansen, *A Reliable Affine Relaxation Method for Global Optimization*, Technical repport, RT-APO-10-05, IRIT, 2010. ftp://ftp.irit.fr/IRIT/APO/RT-APO-10-05.pdf

[5] The COCONUT benchmarks, Library 1 from the GLOBAL Library, website :
http://www.mat.univie.ac.at/∼neum/glopt/coconut/Benchmark/Benchmark.html

# Global Search for Optimistic Solutions in Bilevel Optimization Problems

Andrey V. Orlov

*Institute for system dynamics and control theory of SB RAS, Lermontov str., 134, Irkutsk, 664033, Russia,* anor@icc.ru

**Abstract** The specific classes of nonlinear bilevel optimization problems are considered. Their optimistic statement is reduced to a nonconvex mathematical programming problem with the bilinear structure. Approximate algorithms of local and global search in the obtained problems are proposed.

**Keywords:** bilevel programming, optimistic solution, nonconvex optimization problems, local search, global search

## 1. Introduction

Problems, which arise within the framework of research economic, ecological and financial systems has, as a rule, hierarchical structure [1]–[4]. Such problems can be formulated as hierarchical games or, for example, as bilevel optimization problems [2]:

$$\left.\begin{aligned} F(x,y) \downarrow \text{“}\min_{x}\text{”},\;\; (x,y) \in X,\;\; y \in Y_*(x), \\ Y_*(x) \overset{\triangle}{=} \operatorname*{Argmin}_{y}\{\psi(x,y) \mid (x,y) \in S\}. \end{aligned}\right\} \qquad (\mathcal{BP})$$

In course of investigation of bilevel optimization problems the difficulty arises already at the stage of defining a concept of solution. The optimistic and pessimistic (guaranteed) concepts of solution are the most popular. [1]–[4]. In the first case it is supposed, that the interests of the upper level can be coordinated with actions of the lower level. In the second case the upper level is obliged to work independently, that considerably complicates the problem under scrutiny.

As the elaboration of solution methods of the bilevel oprimization problems in general statement is represented as hopeless problem at the given stage of science, it is natural to go on a way of a construction of solution methods for some classes of bilevel problems with use of their specificity.

During the three decades of intensive investigation of bilevel optimization problems there were proposed various methods for finding an optimistic solutions by different authors (see the surveys [3], [4]). Nevertheless, as far as we can conclude on the basis of available literature, there are only a few published results containing numerical solutions of even test bilevel high-dimension problems (e.g. problems with the dimension up to 200). So, development of new numerical methods for bilevel problems, while implying verification of their efficiency by numerical testing, is one of the most important problems of operations research.

In this work the new approach to elaboration of optimistic solution methods for two classes of bilevel problems is proposed. The approach is based on a possibility of equivalent representation of a bilevel optimization problem as a nonconvex optimization problem [2] with the help of Karush-Kuhn-Tucker (KKT) condidions [5]. As far as solving of the obtained nonconvex problem is concerned, we use the Global Search Theory (GST) in d.c. optimization problems developed in [6] for some classes of nonconvex optimization. The approach allows to build efficient methods for finding global solutions in d.c. optimization problems.

## 2. Problem formulation

This work is devoted to elaboration of new techniques for finding optimistic solutions of bilevel problems, where the upper level goal function is d.c. (represented by difference of two convex functions), and the lower level goal function is quadratic. So that the problem is formulated as follows:

$$\left.\begin{array}{c} F(x,y) \overset{\triangle}{=} g(x,y) - h(x,y) \downarrow \min\limits_{x,y}, \\ (x,y) \in X \overset{\triangle}{=} \{(x,y) \in I\!R^{m+n} \mid f_l(x,y) \le 0,\ l = 1,...,p\}, \\ y \in Y_*(x) \overset{\triangle}{=} \operatorname*{Argmin}\limits_{y}\{\frac{1}{2}\langle y, C_1 y\rangle + \langle xQ_1 + d_1, y\rangle \mid A_1x + B_1y \le b_1\}, \end{array}\right\} \qquad (\mathcal{P})$$

where the functions $g(\cdot)$, $h(\cdot)$, $f_l(\cdot)$, $l = 1,...,p$ are convex on $I\!R^{m+n}$, $d_1 \in I\!R^n$, $b_1 \in I\!R^q$, $A_1, B_1, C_1, Q_1$ are matrices of appropriate dimension, and $C_1 = C_1^T$ is nonnegatively defined.

Also we investigate a particular case of problem $(\mathcal{P})$ with quadratic goal function on the upper and lower levels:

$$\left.\begin{array}{c} F_1(x,y) \overset{\triangle}{=} \frac{1}{2}\langle x, Cx\rangle + \langle c, x\rangle + \frac{1}{2}\langle y, Qy\rangle + \langle d, y\rangle \downarrow \min\limits_{x,y}, \\ (x,y) \in X_1 \overset{\triangle}{=} \{(x,y) \in I\!R^{m+n} \mid Ax + By \le b\}, \\ y \in Y_*(x) \overset{\triangle}{=} \operatorname*{Argmin}\limits_{y}\{\frac{1}{2}\langle y, C_1 y\rangle + \langle xQ_1 + d_1 y\rangle \mid A_1x + B_1y \le b_1\}, \end{array}\right\} \qquad (\mathcal{P}_1)$$

where $c \in I\!R^m$; $d, d_1 \in I\!R^n$; $b \in I\!R^p$; $b_1 \in I\!R^q$; $A, B, C, Q, A_1, B_1, C_1, Q_1$ — are matrices of appropriate dimension, $C = C^T \ge 0$, $Q = Q^T \ge 0$, $C_1 = C_1^T \ge 0$.

Such bilevel problems may be reduced to one or several single-level nonconvex (d.c.) problems via, for instance, the KKT-rule (see, for example, [1], [2]):

$$\left.\begin{array}{c} \Phi(x,y,v) \overset{\triangle}{=} g(x,y) - h(x,y) + \mu\langle v, b_1 - A_1x - B_1y\rangle \downarrow \min\limits_{x,y,v}, \\ (x,y,v) \in D \overset{\triangle}{=} \{(x,y,v) \mid (x,y) \in X,\ v \ge 0, \\ C_1y + d_1 + xQ_1 + vB_1 = 0,\ A_1x + B_1y \le b_1\}; \end{array}\right\} \qquad (\mathcal{DC})$$

$$\left.\begin{array}{c} \Phi_1(x,y,v) \overset{\triangle}{=} \frac{1}{2}\langle x, Cx\rangle + \langle c, x\rangle + \frac{1}{2}\langle y, Qy\rangle + \langle d, y\rangle + \\ +\mu\langle v, b_1 - A_1x - B_1y\rangle \downarrow \min\limits_{x,y,v}, \\ (x,y,v) \in D_1 \overset{\triangle}{=} \{(x,y,v) \mid Ax + By \le b, \quad v \ge 0, \\ C_1y + d_1 + xQ_1 + vB_1 = 0, \quad A_1x + B_1y \le b_1\}, \end{array}\right\} \qquad (\mathcal{DC}_1)$$

where $\mu > 0$ is a penalty parameter,

It is known, that nonconvex problems may have a large number of local solutions, which are far – even from the viewpoint of the goal function's value – from a global one [5], [6].

## 3. Local Search

Direct application of standard convex optimization methods [5] turns out to be inefficient from the view point of global search. So, there appears the need to construct new global search methods, allowing to escape from a stationary (critical) point.

For the purpose of solving the problems formulated above, we intend to construct the algorithms based on the Global Search Theory in d.c. optmization problems elaborated in [6]–[14]. Global Search Algorithms based on GST consist of two principal stages: 1) a special local search methods, which takes into account the structure of the problem under scrutiny; 2) the procedures, based on Global Optimality Conditions, which allow to improve the point provided by the Local Search Method [6]–[14].

In particular, a Local Search in problem ($\mathcal{DC}$) consists in the consecutive (approximate) solving the convex linearized problems of the form ($(x^s, y^s, v^s) \in D$)

$$\left.\begin{array}{c} g(x,y) + \dfrac{\mu}{4}(4\langle v, b_1\rangle + \|v - A_1x\|^2 + \|v - B_1y\|^2) - \\ -\langle \nabla_{xy} h(x^s, y^s)(x,y)\rangle - \dfrac{\mu}{2}(\langle v^s + A_1x^s, v\rangle + \\ +\langle (v^s + A_1x^s)A_1, x\rangle + \langle v^s + B_1y^s, v\rangle + \\ +\langle (v^s + B_1y^s)B_1, y\rangle) \downarrow \min\limits_{x,y,v}, \quad (x,y,v) \in D. \end{array}\right\} \qquad (\mathcal{PL})$$

Linearization in the problem ($\mathcal{PL}$) is performed for the basic (generic) nonconvexity of a problem ($\mathcal{DC}$) [6]–[7], and problem ($\mathcal{PL}$) can be solved by standard software packages.

To the end of a local search for Problem ($\mathcal{DC}_1$) we apply the idea of consecutive solving partial problems with respect to two groups of variables (see [8]–[14]). In order to do it, we separate the pair $(x, y)$ and the variable $v$. For a fixed value of variable $v$ problem ($\mathcal{DC}_1$) becomes a convex quadratic optimization problem, and for a fixed pair $(x, y)$ we obtain a problem of linear programming with respect to $v$ ($(x^s, y^s, v^s) \in D_1$):

$$\left.\begin{array}{c} \dfrac{1}{2}\langle x, Cx\rangle + \langle c, x\rangle + \dfrac{1}{2}\langle y, Qy\rangle + \langle d, y\rangle - \\ -\mu(\langle v^s A_1, x\rangle + \langle v^s B_1, y\rangle) \downarrow \min\limits_{x,y}, \\ Ax + By \le b, \quad A_1x + B_1y \le b_1, \\ C_1y + d_1 + xQ_1 + v^s B_1 = 0, \end{array}\right\} \qquad (\mathcal{QP})$$

$$\left.\begin{array}{c} \langle b_1 - A_1x^s - B_1y^s, v\rangle \downarrow \min\limits_{v}, \\ v \ge 0, \quad C_1y^s + d_1 + x^sQ_1 + vB_1 = 0. \end{array}\right\} \qquad (\mathcal{LP})$$

These auxiliary problems can be solved with the help of standard software packages also.

## 4. Global Search

The procedures of Global Search for problems ($\mathcal{DC}$) and ($\mathcal{DC}_1$) based on the corresponding strategy of global search for problems of d.c. minimization [6]–[14] because the goal function in problems of such kind may be represented as a difference of two convex functions. In combination with directed selection, in the process of increasing the the value of parameters $\mu > 0$, the procedures of global search forms a methods for solving problems ($\mathcal{P}$) and ($\mathcal{P}_1$).

To begin with, one needs a d.c. representation of the goal function of problem under scrutiny. For the problem ($\mathcal{DC}$) one can do it, for instance, as follows (see also ($\mathcal{PL}$)):

$$\Phi(x, y, v) = G(x, y, v) - H(x, y, v),$$

where $G(x, y, v) = g(x, y) + \mu\langle b_1, v\rangle + \dfrac{\mu}{4}(\|v - A_1x\|^2 + \|v - B_1y\|^2)$, and $H(x, y, v) = h(x, y) + \dfrac{\mu}{4}(\|v + A_1x\|^2 + \|v + B_1y\|^2)$ obviously are convex functions.

For the problem ($\mathcal{DC}_1$) the d.c. representation of the goal function can be made as follows:

$$\Phi_1(x, y, v) = G_1(x, y, v) - H_1(x, y, v),$$

where

$$\begin{aligned} G_1(x,y,v) &= F_1(x,y) + \mu\langle b_1, v\rangle + \frac{\mu}{4}(\|v - A_1x\|^2 + \|v - B_1y\|^2), \\ H_1(x,y,v) &= \frac{\mu}{4}(\|v + A_1x\|^2 + \|v + B_1y\|^2). \end{aligned}$$

The crucial moment of Global Search procedures consists in constructing an approximation of the level surface of the convex function, which generates the basic nonconvexity in the problem under consideration [6]. For the purpose of constructing such an approximation we have to take account of the information related to the problems statements.

Computational testing of the elaborated methods has shown the efficiency of the proposed approach.

## 5. Summary

In the present work, new procedures of finding optimistic solutions in two classes of nonlinear bilevel optimization problems have been elaborated. On the one hand, these procedures are based on the well known idea to replace the extremum constraint in the bilevel problem with KKT-conditions. On the other hand, for the purpose of solving the obtained nonconvex single-level problems, novel Global Search Algorithms based on the Global Search Theory from [6]–[14] for the d.c. programming problems have been applied. Besides, new local search algorithms for these nonconvex problems have been elaborated and tested.

**Acknowledgments.** The author wish to thank professor A.S. Strekalovsky for their encouragement and support.

## References

[1] Bard, J.F. (1998). *Practical Bilevel Optimization.* Dordrecht: Kluwer Academic Publishers.

[2] Dempe, S. (2002) *Foundations of Bilevel Programming.* Dordrecht: Kluwer Academic Publishers.

[3] Dempe, S. (2005) *Bilevel programming.* In "Essays and Surveys in Global Optimization" (Ed. by C. Audet, P. Hansen, G. Savard), Springer, P.165–193.

[4] Colson, B., Marcotte, P., and Savard, G. (2007) *An overview of bilevel optimization.* Annals of operations research, V.153, No.1, P.235–256.

[5] Bazaraa, M.S., Shetty, C.M. (1979) *Nonlinear programming. Theory and Algorithms.* New York: John Wiley and Sons.

[6] Strekalovsky, A.S. (2003) *Elements of nonconvex optimization.* Novosibirsk: Nauka (in russian).

[7] Strekalovsky, A.S. (2003) *On the Minimization of the Difference of Convex Functions on a Feasible Set.* Computational Mathematics and Mathematical Physics, V.43, No.3, P.380–390.

[8] Strekalovsky, A.S., Orlov, A.V. (2007) *Bimatrix games and bilinear programming.* Moscow: FizMatLit (in russian).

[9] Strekalovsky, A.S., Orlov, A.V. (2005) *Numerical search for equilibria in bimatrix games.* Computational Mathematics and Mathematical Physics, V.45, No.6, P.947–960.

[10] Strekalovsky, A.S., Orlov, A.V. (2007) *A new approach to nonconvex optimization.* Numerical Methods and Programming (http://num-meth.srcc.msu.su/english/index.html), V.8, P.160–176.

[11] Orlov, A.V. (2008) *Numerical solution of bilinear programming problems.* Computational Mathematics and Mathematical Physics, V.48, No.2, P.225–241.

[12] Strekalovsky, A.S., Orlov, A.V., Malyshev, A.V. (2010) *A local search for the quadratic-linear bilevel programming problem.* Siberian Journal of Numerical Mathematic, V.13., No.1, P.75–88 (in russian).

[13] Strekalovsky, A.S., Orlov, A.V., Malyshev, A.V. (2010) *Numerical solution of a class of bilevel programming problems.* Siberian Journal of Numerical Mathematic, V.13., No.2, P.201–212 (in russian).

[14] Strekalovsky, A.S., Orlov, A.V., Malyshev, A.V. (2010) *On computational search for optimistic solutions in bilevel problems.* Journal of Global Optimization, Special Issue celebrating the 70th birthday of Roman G. Strongin (in print, available online: DOI 10.1007s10898-009-9514-z).

# On a specific method to solve semi-obnoxious continuous facility location problems[*]

P.M. Ortigosa[1] and E.M.T. Hendrix[2]

[1] *Dept. Computer Architecture and Electronics, Almeria University,* ortigosa@ual.es

[2] *Dept. Computer Architecture, Malaga University, Spain* eligius.hendrix@wur.nl

**Abstract** Semi-obnoxious continuous location problems are mostly modeled in literature combining a convex objective representing minimum cost and a multiextremal objective representing the nondesirable part of a facility. Deterministic methods have been designed to solve such problems and generic one or bi-objective heuristic methods have been applied. This paper describes a dedicated method to solve semi-obnoxious location problems making use of its specific structure.

**Keywords:** facility location, obnoxious, pareto, heuristic, global optimization, metaheuristic

## 1. Introduction

Many models have been introduced in literature to describe the location of a facility in the plane. Objectives vary from minimum transportation cost in the Weber problem, maximising market share in competitive Huff-like models, centre problems maximising cover to finally obnoxious objectives to describe that a facility is nondesirable. Obnoxious means according to Erkut and Neuman (1989) [3], that the facility generates a disservice to the people nearby while producing an intended product or service. An intriguing aspect of objective functions describing the nondesirable effect of the facility is that it leads to multiextremal optimisation problems that are hard to solve.

Semi-obnoxious models typically combine a convex objective (e.g. Figure 1) to describe the attraction aspect of the facility with the obnoxious objective (e.g. Figure 2) and thus inherit its multimodal character leading to new challenges for optimisation methods. One can either combine both objectives in a multi-objective fashion or try to represent the efficient solutions that generate the Pareto front. The latter is comprehensible when locating only one facility for decision makers, as one has one graph representing the trade-off of the objectives and one graph representing te corresponding efficient locations on a map.

One of the research questions is how to generate the efficient locations given that we are dealing with a nonconvex objective function. An elaborate overview of literature on the topic is given by Yapicioglu et al. (2006) who approachs the problem by generic bi-objective Particle Swarm algorithms. Another way to approach the problem is to use deterministic branch-and-bound like algorithms that guarantee the quality of found locations, e.g. [6]. The argumentation for using heuristic stochastic or deterministic GO algorithms is that the objective function

[*] This work has been funded by grant TIN2008-01117 from the Spanish Ministry of Science and Innovation and P08-TIC-3518 from the Junta de Andalucía, in part financed by the European Regional Development Fund (ERDF). Eligius Hendrix is a fellow of the Spanish "Ramón y Cajal" contract program, co-financed by the European Social Fund.

is nonconvex. However, seen from the bi-objective perspective, one of the objectives is convex and the other is not. In Section 2 we describe an approach using the model of [1] as an example problem.

In Section 2, we describe a specific metaheuristic to generate efficient solutions of the semi-obnoxious continuous single facility model using the presented method. Illustrations are showed in Section 3. Finally we conclude in Section 4.

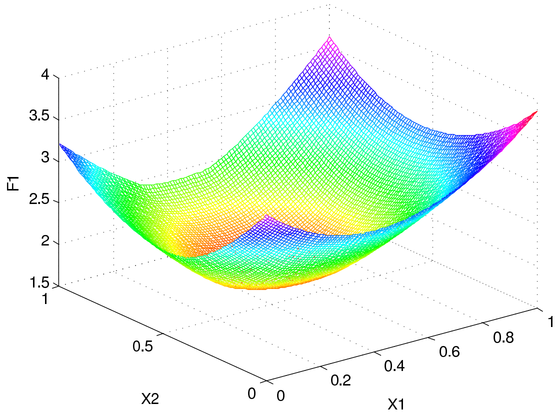


*Figure 1.* Graph minsum (Weber) function $f_1$

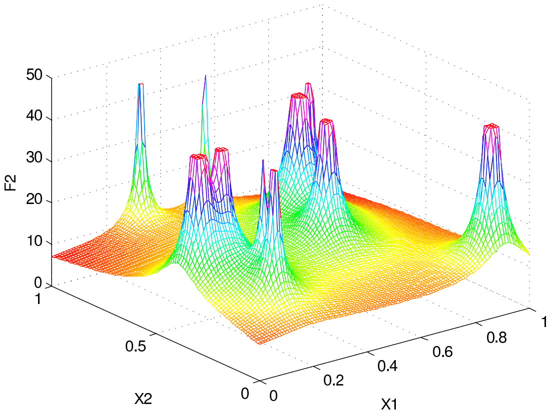


*Figure 2.* Graph obnoxious function $f_2$

## 2. A bi-objective approach

To describe the approach, we use as an example problem the semi-obnoxious model described in Brimberg and Juel [1] where the problem is notated as Bicriterion Semi-obnoxious Location Problem (BSLP). In this problem a desirable and an obnoxious objective function must be minimised. The desirable or convex objective is the classic minsum transportation cost.

$$f_1(x) = \sum_i w_i d_i(x), \tag{1}$$

where $w_i$ are weights and $d_i$ the (Euclidean or rectangular) distance from facility location $x$ to fixed (demand) point $p_i$, $i = 1, \ldots, m$. Minimising (1) is called the median problem.

The obnoxious function minimises the overall obnoxiousness when far from a demand-point, but also it reflects the local effects when close to a demand point [7].

$$f_2(x) = \sum_i v_i d_i(x)^{-b}, \tag{2}$$

where $b > 0$ takes on a specified value depending on the type of facility being considered and $v_i$ is again a weight like the population size [2].

Figures 1 and 2 give an impression of the two objective functions for 10 randomly generated fixed points and weights. One can observe that minsum objective $f_1$ is convex whereas obnoxious objective $f_2$ is multiextremal. Notice that $f_2$ function does not permit getting too close to an existing facility as $d_i(x)$ tends to zero.

A decision maker is usually interested in the efficient points over the feasible set $X$ of such a problem where $f_1$ as well as $f_2$ is minimised. An efficient (nondominated) location $x^*$ is defined in multiobjective sense such that there does not exist another location $x \in X$ with $f_1(x) < f_1(x^*)$ and $f_2(x) \leq f_2(x^*)$ or alternatively $f_2(x) < f_2(x^*)$ and $f_1(x) \leq f_1(x^*)$. One is usually interested in the set of efficient locations $X^*$ and the so called Pareto front $\{(f_1(x), f_2(x))|x \in X^*\}$ that sketches the trade-off between the two objective values.

There are several ways to approach the generation of efficient solutions. One can combine the objectives in one weighted function or alternatively restrict iteratively one objective like $f_1(x) \leq tc$ and minimise the other. Let

$$R(tc) = \{x \in X | f_1(x) \leq tc\} \tag{3}$$

denote a level set of the convex objective $f_1$. Notice that in our case $R(tc)$ is a convex set. One can follow the last approach by using Algorithm 1.

**Algorithm 1 Eff($X, f_1, f_2, \delta$)**

```
Determine $S := \arg\min_{x \in X} f_2(x)$
$tc := \min_{x \in S} f_1(x)$
while ($S \neq \emptyset$)
    $tc := tc - \delta$
    $S := \arg\min_{x \in R(tc)} f_2(x)$
    $tc := \min_{x \in S} f_1(x)$
endwhile
```

First of all, if at a certain iteration $S$ is completely interior with respect to $R(tc)$, we know that after reduction of the level $tc$ of the second objective we only have to check the boundary of $R(tc)$ on the appearance of better function values of $f_2$. Secondly, we are interested in this approach from the perspective of meta-heuristics. Given that we found the solution $x \in S$ on the boundary of $R(tx)$ one can use the information of convexity to restrict new generation of points only in the direction $d$ with $d^T \nabla f_1(x) \leq 0$. Moroever, we use developed a method to generate points uniformly over an ellipsoidal set approximating the current contour of $f_1$ by fitting a quadratic function through the current population of sample points.

In this work a metaheuristic method is implemented following the different steps described in Algorithm 1. For optimising the $f_1$ function a gradient based local optimiser is applied (Weiszfeld-like method) and for optimising $f_2$ a metaheuristic global optimization algorithm based on subpopulation has been implemented. The method will be specified further in the full paper.

## 3. Illustration

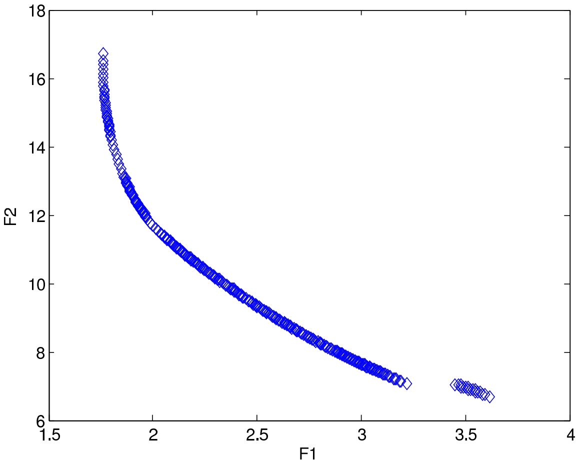


*Figure 3.* Pareto Front

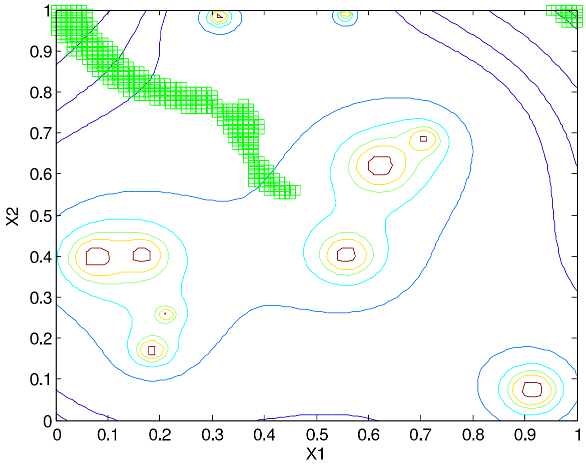


*Figure 4.* Efficient points and contours of $f_2$

To illustrate the behaviour of the algorithm, the steps of Algorithm 1 have been followed and hence a Pareto set of efficient points was generated. A case has been used where 10 demand points have been generated randomly together with the weights. Figure 3 shows the Pareto Front obtained for the two objective functions. In Figure 4, contour lines of $f_2$ have been drwan to give an impression of the optimum points. It can be see that the areas with low objective function are typically in the corners, as is usual in obnoxious objective functions, also called the "mother in law effect". The green squares in this figure represent the set of Pareto efficient points. They tend to the low values of $f_2$ as well as to the middle of the figure where typically the optimum of the minsum objective $f_1$ can be found.

## 4. Conclusions

In this work the new metaheuristic algorithm is developed and tested on four different semiobnoxious problems solved in [8] by using different particle swarm optimisers (PSO). These problems were previously defined and solved in [1], [2], [4], [5] and [7]. Comparison between the new method and PSO methods will be provided. We designed and evaluated specific methods for generating efficient solutions for the semi-obnoxious one facility problems in the plane making use of the idea that one of the objectives is convex approximating its contour by an ellipsoidal region.

## References

[1] J. Brimberg and H. Juel. A bi-criteria model for locating a semi-desirable facility in the plane. *European Journal of Operational Research*, 1:144–151, 1998.

[2] J. Brimberg and H. Juel. A minisum model with forbidden regions for locating a semi-desirable facility in the plane. *Location Science*, 6:109–120, 1998.

[3] E. Erkut and S. Newman. Analytical models for locating undesirable facilities. *European Journal of Operational Research*, 40:275–291, 1989.

[4] E. Melachrinoudis. Bicriteria location of a semi-obnoxious facility. *Computers and Industrial Engineering*, 37:581–593, 1999.

[5] E. Melachrinoudis and Z. Xanthopulos. Semi-obnoxious single facility location in euclidean space. *Computers and Operations Research*, 30:2191–2209, 2003.

[6] Dolores Romero-Morales, Emilio Carrizosa, and Eduardo Conde. Semi-obnoxious location models: A global optimization approach. *European Journal of Operational Research*, 102(2):295–301, 1997.

[7] A.J.V. Skriver and K.A. Andersen. The bicriterion semi-obnoxious location problem (BSLP) solved by an $\epsilon$-approximation. *European Journal of Operational Research*, 146:517–528, 2003.

[8] Haluk Yapicioglu, Alice E. Smith, and Gerry Dozier. Solving he semi-desirable facility location problem using bi-objective particle swarm. *European Journal of Operational Research*, 177:733–749, 2007.

# Branch and Bound for nonconvex SIP

Mohand Ouanes[1,2] and Le Thi Hoai An[2]

[1] *Department of Mathematics, University of Tizi-Ouzou, Algeria,* ouanes_mohand@yahoo.fr

[2] *LITA, University of Metz, France,* lethi@univ-metz.fr

**Abstract** We propose a new method for solving nonconvex semi-infinite problems by using the techniques of global optimization. We generate two sequences of points, the first sequence of the interior points which are feasible, the second sequence of external points. The first sequence gives an upper bound and the second sequence gives a lower bound. If we decide to stop our algorithm after a finite number of iterations, we have an exact solution or an approximate solution which is feasible. We have tested our method on some test problems to show its efficiency.

**Keywords:** Branch and Bound, semi-infinite optimization, global optimization

## 1. Introduction

We consider the following problem

$$(P) \begin{cases} \min f(x) \\ g(x,s) \leq 0 \; \forall s \in S \in R \\ x \in R^n \end{cases}$$

with $f$ and $g$ of class $C^2$ and nonconvex functions. S is a compact of R. There exist many method to solve problem $(P)$: the discretization method, the lagrangian method etc...(see [1], [5], [6], [8], [11]). The difficulty for this methods is: if we decide to stop the algorithm after a finite umber of iterations the approximate solution is not feasible which is the drawback when the feasibility is too important than the optimality for certain problems. With our method at each iteration we have an approximate solution which is feasible. We solve two problems at each iteration: a relaxed discretized problem which gives a lower bound of the minimum of the objective function value of the problem $(P)$, and another problem by using the upper bound of the function g with respect to $s$ instead of g with the same objective function to obtain a upper bound, because the solution given by the second problem is feasible. The paper is organized as follows: in the section 2, we present a method how to construct a upper bound function, in the section 3, the algorithm and its convergence are presented. The numerical examples are treated in section 4.

## 2. Upper bound function

We consider a function $f$ of class $C^2$ defined on the interval $[s_0, s_1], h = s_1 - s_0$.

We now explain how to construct an upper bound of a function $f$ which is twice continuously differentiable on an interval $[a, b]$. For $m \geq 2$, let $\{w_1, w_2, ......, w_m\}$ be the pairwise functions defined as [2].

$$w_i(s) = \begin{cases} \frac{s-s_{i-1}}{s_i-s_{i-1}} & \text{if } s_{i-1} \le s \le s_i \\ \frac{s_{i+1}-s}{s_{i+1}-s_i} & \text{if } s_i \le s \le s_{i+1} \\ 0 & \text{otherwise.} \end{cases}$$

We have $\sum_{i=1}^{i=m} w_i(s) = 1, \forall s \in [a,b]$ and $w_i(s_j) = 0$ if $i \neq j$, 1, otherwise. Let $L_h f$ be the piecewise linear interpolant to $f$ at the points $s_1, s_2, \ldots, s_m : L_h f(s) = \sum_{i=1}^{i=m} f(s_i) w_i(s)$.

The next result from [2] gives a upper bound and a lower bound of $f$ on the interval $[s_0, s_1]$, $(h = s_1 - s_0)$.

**Theorem 1.** [2] *For all* $s \in [s_0, s_1]$, *we have* $|L_h f(s) - f(s)| \le \frac{1}{8} K h^2$, *i.e.,*

$$L_h f(s) - \frac{1}{8} K h^2 \le f(s) \le L_h f(s) + \frac{1}{8} K h^2.$$

In [17] the following quadratic lower bounding function of $f$ is proposed:

$$Lf(s) := L_h f(s) - \frac{1}{2} K (s - s_0)(s_1 - s) \le f(s),\ \forall s \in [s_0, s_1]$$

It has been proved (see [17]) that this lower bound is better than the affine minorization given in [2]:

$$Lf(s) \ge L_h f(s) - \frac{1}{8} K h^2.$$

In a similar way, we now introduce a concave quadratic upper bounding function of $f$ :

**Theorem 2.** *For all* $s \in [s_0, s_1]$ *we have*

$$L_h f(s) + \frac{1}{8} K h^2 \ge Uf(s) := L_h f(s) + \frac{1}{2} K (s - s_0)(s_1 - s) \ge f(s). \tag{1}$$

*Proof.* Let $E(s)$ be the function defined on $[s_0, s_1]$ by $E(s) = L_h f(s) + \frac{1}{8} K h^2 - Uf(s) = \frac{1}{8} K h^2 - \frac{1}{2} K (s - s_0)(s_1 - s) = \frac{K}{2} \left[ s^2 - (s_0 + s_1) s + s_0 s_1 + \frac{1}{4} (s_1 - s_0)^2 \right]$. $E$ is convex on $[s_0, s_1]$, and its derivative is equal to zero at $s^* = \frac{1}{2}(s_0 + s_1)$. Therefore, for any $s \in [s_0, s_1]$ we have $E(s) \ge \min\{E(s) : s \in [s_0, s_1]\} = E(s^*) = 0$. Then, the first inequality in (1) holds. Consider now the function $\phi$ defined on $S$ by $\phi(s) := Uf(s) - f(s) = L_h(s) + \frac{1}{2} K (s - s_0)(s_1 - s) - f(s)$.

It is clear that $\phi''(s) = -K - f''(s) \le 0$ for all $s \in S$. Hence $\phi$ is a concave function, and for all $s \in [s_0, s_1]$ we have $\phi(s) \ge \min\{\phi(s) : s \in [s_0, s_1]\} = \phi(s_0) = \phi(s_1) = 0$.

The second inequality in (1) is then proved. □

# 3. Algorithm and its convergence

We now describe our algorithm

## 3.1 Algorithm

**Step 0:** Let $\varepsilon > 0, S = [s_0, s_1], H$ hyperrectangle which contains the domain of $(P)$

**Step 1:** Solve the relaxed problem (of the discretized problem)

$$\left(P_0^L\right) \begin{cases} \min L_H f(x) \\ L_H g(x, s_i) \le 0\ i = 0, 1 \\ x \in H \subset R^n \end{cases}$$

to get $x_0^L$, else infeasibility. Let $LB_0^R = L_H f(x_0^L)$

**Step 2:** feasibility if possible. Compute $U_{h_s}g(x_0^L, s)$ and $max_{s\in S}U_{h_s}g(x_0^L, s)$ to obtain $s^*(x_0^L)$. If $U_{h_s}g(x_0^L, s^*(x_0^L)) \leq \epsilon$, stop, $x_0^L$ is an optimal solution else continue.

**Step 3:** Solve the problem locally (we use SQP method with warm start)

$$(P_0^U)\begin{cases} \min f(x) \\ U_{h_s}g(x, s^*(x)) \leq 0, h = s_1 - s_0 \\ x \in R^n \end{cases}$$

to get a feasible point if possible $x_0^U$ and let $UB_0 = f(x_0^U)$, else let $UB_0 = \infty$.

**Step 4:** If $UB_0 - LB_0 \leq \varepsilon$, stop, $x_0^U$ is an $\varepsilon$-optimal solution, else continue.

**Step 5:** Add $s^*(x_0^L)$ to the discretization $s_0, s_1$ and go to iteration $k$.

Iteration $k = 1, 2, 3, .......$

$k_1$ Reorder $s_{k-1}^*, s_0, s_1, ......, s_k$ by increasing order $s_0, s_1, ......, s_{k+1}$

$k_2$ Solve the two relaxed problems (of the discretized problems)

$$(P_k^L)\begin{cases} \min L_{H_{kj}}f(x) \\ L_{H_{kj}}g(x, s_i) \leq 0 \; i = 0, 1, \ldots, k+1 \\ x \; in H_{kj}, j = 1, 2 \\ x \in R^n \end{cases}$$

To get $x_{kj}^L$ and let $LB_{kj}^R = L_{H_{kj}}f(x_{kj}^L)$.

$k_3$ Feasibility if possible. If $U_{h_s}g(x_{kj}^L, s^*(x_{kj}^L)) \leq \varepsilon$, delete $H_{kj}$, and continue.

$k_4$ Solve the problem locally (we use SQP method with the warm start)

$$(P_k^U)\begin{cases} \min f(x) \\ max_{s\in[s_{j-1}, s_j]}U_{h_s}g(x, s) \leq 0, j = 1, ...., k+1 \\ x \in R^n \end{cases}$$

to get a feasible point if possible $x_k^U$ and let $UB_k = min\{UB_{k-1}, f(x_k^U)\}$, else let $UB_k = \infty$.

$k_5$ Let $LB_k = min\{LB_{kj}^R, LB^{rest}\}$, if $UB_k - LB_k \leq \varepsilon$, stop, $x_k$ such that $UB_k = f(x_k)$ is an $\varepsilon$-optimal solution, else add $s^*(x_k^L)$ to the actual discretization, delete all $H_{ij}$ such that $LB_{ij} \geq UB_k$, select $H_{ij}$ with the lowest lower bound and let it $H_{k+1}$ and go to $k_1$.

**Remarks**

1. $U_{h_s}g(x, s) = g(x, s_0)w_0(s) + g(x, s_1)w_1(s) + g(x, s_0)w_0(s) + \frac{K}{2}(s - s_0)(s_1 - s)$ is an upper bound function of $g(x, .)$ on the interval $S = [s_0, s_1]$.
2. $max_{s\in[s_0,s_1]}U_{h_s}g(x, s)$ is computed by $s^*(x) = \frac{s_0+s_1}{2} + \frac{g(x,s_1-g(x,s_0}{Kh}$.
3. If

$$(P_0^U)\begin{cases} \min f(x) \\ U_{h_s}g(x, s^*(x)) \leq 0, h = s_1 - s_0 \\ x \in R^n \end{cases}$$

has a solution then it is clear that it is feasible for the problem $(P)$ because we have $g(x, s) \leq U_{h_s}g(x, s^*(x))$.

4. $L_{H_{kj}}f(x)$ is lower bound function (which is convex) computed by using the vertex of the main diagonal(for details and convergence see [18]) which is the generalisation of the method presented in section 2 adapted to the lower bound.
5. $LB^{rest}$ is the set of lower bounds of hyperrectangles which aren't deleted or subdivided.

### 3.2 Convergence

As the problems which give $UB_k, LB_k$ have the same objective function, it suffices to show the following theorems.

**Theorem 3.** $lim_{h\to 0}(g(x,s) - U_{h_s}g(x,s)) = 0, h = s_1 - s_0.$

*Proof.* $0 \leq (U_{h_s}g(x,s) - g(x,s)) \leq (U_{h_s}g(x,s) - L_{h_s}g(x,s)) \leq K(s-s_0)(s_1-s) \leq Kh^2$ then $0 \leq lim_{h\to 0}((U_{h_s}g(x,s) - g(x,s)) \leq lim_{h\to 0}Kh^2 = 0$ and the theorem is proved. □

**Theorem 4.** *$LB_k \nearrow f(x^*)$ and $UB_k \searrow f(x^*)$ when $k \to \infty$.*

*Proof.* We have the domain of problem $(P_k^U$ which tends to the domain of problem $(P)$ by the interior,and by construction $UB_k$ is not an increasing sequence. The domain of problem $(P_k^L$ which tends to the domain of problem $(P)$ by the exterior then $lim_{k\to\infty}(UB_k - LB_k) = 0$. $LB_k$ is obtained by solving the discretized problems then it is an increasing sequence. By $lim_{k\to\infty}(UB_k - LB_k) = 0$, we conclude that $LB_k \uparrow f(x^*$ and $UB_k \downarrow f(x^*$ when $k \to \infty$. □

**Remarks**

1. To get the upper bound, we have to solve the problem $\left(P_k^U\right)$ by any local method, we use SQP method with the warm start in our method. To get the lower bound, we have to solve globally the discretized problem at each iteration which is a difficult problem. We propose in our method to solve a relaxed problem by using the main diagonal of $H$ which is a convex problem.
2. The lower bound fonction $L_{H_{kj}}g(x,s_i) \to g(x,s_i)$, when the lenght of the main diagonal of the hyperrectangle $H_{kj} \to 0$.
3. We use the exhaustive $w - subdivision$ which means that $H_{kj}$ tends to one point when the number of iterations $\to \infty$ then $LB_k - LB_k^R \to 0$, with $LB_k$ the exact lower bound of the discretized problem at iteration$k$. We have showed $(UB_k - LB_k) \to 0$ when $k \to \infty$ which means that: $(UB_k - LB_k^R) \to 0$ when $k \to \infty$. Then $UB_k \to f(x^*)$ and $LB_k^R \to f(x^*)$, with $x^*$ the optimal solution of the (SIP) problem.

## 4. Numerical examples

Problem 1 [1]

$$\begin{cases} \min \frac{1}{3}x_1^2 + x_2^2 + \frac{1}{2}x_1 - x_2 \\ x_1^2 + 2x_1x_2s - \sin(s) \leq 0, \forall s \in [0,2] \\ x_1, x_2 \in R \end{cases}$$

We solve the relaxed discretized problem by using the endpoints of the interval $\{0,2\}$. the solution found is $(0, 0.5)$ which is feasible then it is optimal.

Problem 2 [1]

$$\begin{cases} \min \frac{1}{3}x_1^2 + x_2^2 + \frac{1}{2}x_1 \\ (1 - x_1^2s^2)^2 - x_1s^2 - x_2^2 + x_2 \leq 0, \forall s \in [0,1] \\ x_1, x_2 \in R \end{cases}$$

We solve the relaxed discretized problem by using the endpoints of the interval $\{0,1\}$. the solution found is $(-0.75, -0.61808)$ which is feasible then it is optimal.

Problem 3 [8]

$$\begin{cases} \min 2.25\exp(x_1 + x_2) \\ s - \exp(x_1 + x_2 \leq 0, \forall s \in [0,1] \\ x_1, x_2 \in R \end{cases}$$

We solve the relaxed discretized problem by using the endpoints of the interval $\{0,1\}$. the solution found is $(-0.405, -0.405)$. We use the upper bound function, as $g(x,s)$ is linear with respect to $s$ then its maximum is reached at the endpoints of the interval, in this case it is 1. We obtain the optimal solution which is the same as in [8].

Problem 4 [8]

$$\begin{cases} \min 1.21\exp(x_1 + x_2) \\ s - \exp(x_1 + x_2 \leq 0, \forall s \in [0,1] \\ x_1, x_2 \in R \end{cases}$$

We solve the relaxed discretized problem by using the endpoints of the interval $\{0, 1\}$. the solution found is $(-0.0953, -0.0953)$. We use the upper bound function, as $g(x, s)$ is linear with respect to $s$ then its maximum is reached at the endpoints of the interval, in this case it is 1. We obtain the optimal solution which is the same as in [8].

## 5. Conclusion

We have proposed a new method to solve non convex semi-infinite problems by using two sequences of points, one of the interior points and another of external points which give upper and lower bound respectively. The techniques of the global optimization are used in our method. Our method gives an exact solution or an approximate solution which is feasible. Numerical examples found in the litterature are solved efficiently.

## References

[1] B. Bhattacharjee,W.H.Green,Jr and P.Barton (2005), *Interval Methods for Semi-infinite Programs*, Computational Optimization and applications, 30:1-13

[2] De Boor, C.: A practical Guide to Splines Applied Mathematical Sciences. Springer Verlag (1978)

[3] Bradley and M.Bell (1989), *Global convergence of semi-infinite optimization method*, Applied maths and optimization, Springer Verlag N.Y

[4] P.G. Ciarlet (1980), *Elements finis*, Dunod,

[5] Conn and Gould (1979), *An exact penalty function for semi-infinite problems*, University of Bonn, W. Germany

[6] Coope and Watson (1985), *A projected lagragian algorithm for semi-infinite progamming*, Mathematical Programming, North Holland

[7] Gribbik (1979), *A central cutting plane algorithm for semi-infinite programming problems*, Univ. Pitsburg, U.S.A

[8] Guo-Xin Liu (2007), *A homotopy interior point method for semi-infinite programming problems*, Journal of Global optimization, 37:637-646

[9] Gustavson and Glashoff (1983), *Linear optimization and approximation*, Royal institut of technology, Stockholm, Swenden

[10] Gustavson and Kortaneck (1984), *Linear optimization and applications*, Royal institut of technology, Stockholm, Swenden

[11] Hettich and Kortaneck (1993), *Semi-infinite programming : Theory, methods an applications*, SIAM revue, 35.3:380-429

[12] H. Hu (1996), *A globally convergente method for semi-infinite linear programming*, Journal of Global optimization, 8.2:189-199

[13] H. Hu (1998), *A one phase algorithm for linear semi-infinte programming*, Mathematical Programming, North Holland

[14] O.L. Kstyukova (2000), *An algorithm for constructing solutions for a family of linear semi-infinite problems* ,Variationnal calculus, optimal control and applications, 1-34

[15] Le Thi Hoai An and Pham Dinh Tao (1998), *A branch and bound method via D.C. optimization algorithm and ellipsoiadal technique for box constrained nonconvex quadratic problems*, Journal of Global optimization, 13.2:171-206

[16] Le Thi Hoai An and M. Ouanes (2004), *A tighter lower bound by convex quadratic function for univariate global optimization* ,Edited by Le Thi Hoai An and Pham Dinh Tao in Hermes Science Publishing :223-231

[17] Le Thi Hoai An and Ouanes (2006), *Convex quadratic underestimation and Branch and Bound for univariate global optimization with one nonconvex constraint*, RAIRO Recherche Opérationnelle, 40:285-302

[18] Le Thi Hoai An and Ouanes (2007), *Methode de la diagonale dominante pour un probleme d'optimisation globale en dimension n sur un hyperrectangle*, Cosi'07, UST Oran 11-13 juin 2007, Algerie.

[19] M.K. Luandjula, M. Belhouas and M. Ouanes (1994), *Semi-infinite programming: A collocation approach*, International center for theoritical physics, Triest, Italy

[20] M.K. Luandjula and M. Ouanes (2001), *A cutting plane method for semi-infinite optimization* , African journal of science and technology, 1-10

[21] G.Still (2001), *Discretization in semi-infinite programming: The rate of the convergence*, Optimization, 53-69

# Robust Support Vector Machines with Polyhedral Uncertainty of the Input Data

Panos M. Pardalos and Neng Fan

*Center for Applied Optimization, Industrial and Systems Engineering Department, University of Florida,*
{pardalos,andynfan}@ufl.edu

**Keywords:** Support Vector Machines, robust optimization, polyhedral uncertainty, nonlinear programming, classification.

In this paper, we use robust optimization models to formulate the support vector machines (SVMs) with polyhedral uncertainties of the input data points. The formulations in our models are nonlinear and we use Lagrange multipliers and other reformulation methods to solve these problems. In addition, we have proposed the models for transductive SVMs with input uncertainties. They are formulated as mixed integer nonlinear programs. We use decomposition methods to solve these problems. The numerical experiments show that our proposed methods are useful for data with noise in classification.

# Global Optimization Applications in Biomedicine

Panos M. Pardalos and Neng Fan

*Center for Applied Optimization ISE, MBE Departments, McKnight Brain Institute and University of Florida Genetics Institute, University of Florida* {pardalos,andynfan}@ufl.edu

**Keywords:** global optimization, biomedicine, data mining, cancer research.

In recent years optimization has been widely used in many problems in biomedicine. These problems are inherently complex and very difficult to solve. In this talk we are going to focus on global optimization techniques (multi-quadratic 0-1 integer programming) in computational neuroscience and biclustering (nonlinear fractional 0-1 integer programming) based data mining approaches in cancer research. In addition, several other applications will be briefly discussed.

Proceedings of TOGO 2010, pp. 123 – 126.

# The Linear Bilevel Problems via Nonconvex Constraint Problems

Elena G. Petrova and Tatyana V. Gruzdeva

*Institute for System Dynamics and Control Theory SB of RAS, Russia,* nekolyap@mail.ru

**Abstract** A new approach for finding optimistic solutions in linear bilevel problems is considered. First, we investigate the interrelations between these problems and problems with constraint represented as a difference of two convex functions (d.c. constraint). Further on the basis of the global search strategy for d.c. constraint problems a special global method for linear bilevel problems is developed.

**Keywords:** bilevel programming, optimistic solution, nonconvex optimization, d.c. function, problems with nonconvex constraint, local search, global search

## 1. Introduction

This paper is devoted to solving linear bilevel problems [3]. Bilevel programming problems arise when a two-level hierarchical system is modelled. In this system there is a leader who takes the decision first, and a follower whom decision depends on the leader's one. This kind of problems has a wide field of applications [1]: network design, transportation, game theory, engineering and economics. The bilevel problems are generally difficult to solve due to the non-convex nature of the search space and nondifferential of the objective function of the upper level resulting from the complex interaction of the upper and lower problems. Furthermore, it may have local optima different from a global one even w.r.t. the values of objective function.

As well known, elaboration of methods for solving nonconvex problems of large dimensions is one of the problems of contemporary importance [5]. Our paper proposes a new way of solving the linear bilevel problems. As will be shown in the first part of the article, a bilevel problem can be reduced to optimization problem with d.c. constraint. Therefore we can use the global search strategy for d.c. constraint problems [7] in this case.

## 2. Problem statement

Consider the linear bilevel program in the form

$$\left.\begin{array}{c} f(x,y) \overset{\triangle}{=} \langle c,x\rangle + \langle d,y\rangle \downarrow \min\limits_{x,y}, \\ \text{s.t. } \; Ax + By \le b, \\ \text{y is a solution of} \\ \left.\begin{array}{c} \langle d^1, y\rangle \downarrow \min\limits_{y} \\ \text{s.t. } \; A_1x + B_1y \le b^1, \end{array}\right\} (\mathcal{P}_L(x)) \end{array}\right\} \qquad (\mathcal{BP})$$

where $A,\ B,\ A_1,\ B_1$ are given matrices of dimensions $(p \times m)$, $(p \times n)$, $(q \times m)$, $(q \times n)$ respectively and $c \in \mathbb{R}^m$, $d, d^1 \in \mathbb{R}^n$, $b \in \mathbb{R}^p$, $b^1 \in \mathbb{R}^q$ are given vectors.

Let us define the following sets:

$$X \overset{\triangle}{=} \{(x,y) \in I\!R^{m+n} \mid Ax + By \leq b\},\ Y(x) \overset{\triangle}{=} \{y \in I\!R^n \mid A_1x + B_1y \leq b^1\},$$

$$Y_*(x) \overset{\triangle}{=} \operatorname{Arg}\min_y \left\{\langle d^1, y\rangle \mid A_1x + B_1y \leq b^1\}\right\}.$$

Sets $X$ and $Y(x)$ are called upper and lower level constraints. $Y_*(x)$ is an extremal constraint of problem $(\mathcal{BP})$. This formulation corresponds to the optimistic formulation and implies that whenever the optimal solution set $Y_*(x)$ does not reduce itself to a singleton for some $x$, the leader selects a solution among these indifferent reactions set that suits him best. Let sets $X$ and $Y_*(x)$ be compacts for any fixed $x \in X$. Consider lower level problem $(\mathcal{P}_L(x))$. It is a linear programming problem. Suppose $x \in X$ is the parameter; then dual problem is

$$\left.\begin{array}{r} \langle A_1x - b^1, v\rangle \uparrow \max\limits_v \\ \text{s.t. } v \in V \overset{\triangle}{=} \{v \in I\!R^q \mid vB_1 = -d^1,\ v \geq 0\}. \end{array}\right\} (\mathcal{D}_L(x))$$

Suppose, the functions $f(x,y)$ and $\langle d^1, y\rangle$ are bounded above on sets $X$ and $Y(x)$ respectively. In this case

$$\inf_x \inf_y \{\langle d^1, y\rangle \mid y \in Y(x),\ x \in Pr(X)\} > -\infty. \tag{1}$$

In accordance with the dual theory of linear programming, the following inequality

$$\langle d^1, y\rangle - \langle A_1x - b^1, v\rangle \geq 0 \tag{2}$$

is truth for all feasible triples $(x,\ y,\ v)$. Apart from that, by assumption (1) problems $(\mathcal{P}_L(x))$ and $(\mathcal{D}_L(x))$ have solutions. This means that there exist vectors $\bar{y} \in I\!R^n$ and $\bar{v} \in I\!R^q$ such that the triple $(x,y,v) = (x,\bar{y},\bar{v})$ satisfies the following system

$$\left.\begin{array}{r} \langle d^1, y\rangle = \langle A_1x - b^1, v\rangle, \\ A_1x + B_1y \leq b^1,\ \ vB_1 = -d^1,\ \ v \geq 0. \end{array}\right\} \tag{3}$$

If we replace $Y_*(x)$ by system (3) in problem $(\mathcal{BP})$, we obtain the following mathematical optimization problem:

$$\left.\begin{array}{r} \langle c, x\rangle + \langle d, y\rangle \downarrow \min\limits_{x,y,v}, \\ \text{s.t. } Ax + By \leq b, \\ A_1x + B_1y \leq b^1, \\ vB_1 = -d^1,\ \ v \geq 0, \\ \langle d^1, y\rangle = \langle A_1x - b^1, v\rangle. \end{array}\right\} \tag{$\mathcal{P}$}$$

We see that problem $(\mathcal{P})$ is nonconvex since it has the bilinear constraint-equality $\langle d^1, y\rangle = \langle A_1x - b^1, v\rangle$. The following theorem shows interrelations between $(\mathcal{P})$ and $(\mathcal{BP})$.

**Theorem 1.** *[1, 3] The tuple $(x^*, y^*)$ is the solution of bilevel problem $(\mathcal{BP})$ if and only if there exists vector $v^*$ such that $(x^*, y^*, v^*)$ is a solution of $(\mathcal{P})$.*

In other words, it means that we can solve problem $(\mathcal{P})$ with nonconvex constraint instead of bilevel problem $(\mathcal{BP})$.

One of the popular approaches for solving $(\mathcal{P})$ is penalty methods [8]. But here we propose to solve this problem by means of global search theory for d.c. constrained problems. Taking into account (2), we can represent bilinear constraint-equality as follows:

$$F(x,y,v) \overset{\triangle}{=} \langle A_1x - b^1, v\rangle - \langle d^1, y\rangle \geq 0, \tag{4}$$

since (4) is fulfilled for feasible $(x,y,v)$ only if $\langle d^1, y\rangle = \langle A_1x - b^1, v\rangle$. Besides $F(x,y,v)$ can be represented as a difference of two convex functions: $F(x,y,v) = g(x,v) - h(x,y,v)$, where

$g(x,v) = \frac{1}{4}\|A_1x+v\|^2$, $h(x,y,v) = \frac{1}{4}\|A_1x-v\|^2 + \langle d^1, y\rangle + \langle b^1, v\rangle$. So, $(\mathcal{P})$ is a d.c. constraint problem. We can formulate this problem in the following form:

$$f(z) \downarrow \min, \ z \in S, \ \ F(z) \leq 0, \qquad (\mathcal{DCC}),$$

where $z \overset{\triangle}{=} (x,y,v)$, $S \overset{\triangle}{=} \{Ax+By \leq b, \ \ A_1x + B_1y \leq b^1, \ \ \ vB_1 = -d^1, \ \ v \geq 0\}$. In the next sections of the article we propose local and global algorithms for solving $(\mathcal{DCC})$.

## 3. Local and global search

The special local search method to solve d.c. constraint problems, we propose [4], consists of two procedures. The first procedure constructs a point $(\overline{x}, \overline{y}, \overline{v})$ with the property $\Phi(\overline{x}, \overline{y}, \overline{v}) = 0$. The second procedure consists in solving the linearized problem

$$\begin{array}{l} h(x,y,v) - \langle \nabla g(\bar{x}, \overline{v}), (x,v)\rangle \downarrow \min\limits_{x,y,v}, \\ \text{s.t.} \ \ (x,y,v) \in S, \ \ f(x,y) \leq \rho. \end{array} \tag{5}$$

However, as the following assertion shows, no needs to apply the latter procedure for the given problem.

**Proposition 2.** *Any feasible point $\bar{z} = (\bar{x}, \bar{y}, \bar{v})$ in the problem $(\mathcal{P})$ is a solution of linearized problem* $(5)$ *with* $\rho := \overline{\rho} = f(\overline{x}, \overline{y})$.

In this case, special local search method for d.c. constrained problems becomes a procedure of seeking a feasible point.

Basic steps of global search algorithm for d.c. constraint problems [7] are local search, construction of a level surface approximation and solving of linearized problems. The level surface approximation of function $g(x,v) = \frac{1}{4}\|A_1x+v\|^2$ is constructed with using of basis vectors $e^i \in I\!R^{m+q}$.

Linearized problem is following:

$$\left.\begin{array}{c} J(x,y,v) \downarrow \min\limits_{x,y,v}, \\ \text{s.t.} \ \ Ax + By \leq b, \ \ A_1x + B_1y \leq b^1, \\ vB_1 = -d^1, \ \ v \geq 0, \ \ \langle c, x\rangle + \langle d, y\rangle \leq \rho, \end{array}\right\} \qquad (\mathcal{PL}(\bar{x}, \bar{v}, \rho))$$

where $J(x,y,v) \overset{\triangle}{=} \frac{1}{4}\|A_1x-v\|^2 + \langle d^1, y\rangle + \langle b^1, v\rangle - \langle \nabla g(\bar{x}, \bar{v}), (x,v)\rangle$.

Problem $(\mathcal{PL}(\bar{x}, \bar{v}, \rho))$ is convex quadratic programming problem. Therefore we can use application packages to solve $(\mathcal{P})$, for example, Xpress-MP: http://www.dashoptimization.com.

Taking into account specific structure of problem $(\mathcal{P})$, global search algorithm for d.c. constraint problems undergoes a modification. Let $\beta_- \overset{\triangle}{=} \inf(g, S)$, $\beta_+ \overset{\triangle}{=} \sup(g, S)$, $\Delta\beta$ be given.

**Global search algorithm**

**Step 0.** Put $k := 0$. Find a feasible point $z^0 = (x^0, y^0, v^0)$.
**Step 1.** Put $\beta_k := g(z^k)$, $\rho_k := f(z^k)$. Construct approximation

$$\mathcal{A}_k = \{w^i = \begin{pmatrix} \bar{x}^i \\ \bar{v}^i \end{pmatrix} = \lambda_i e^i \mid g(\bar{x}^i, \bar{v}^i) = \beta_k, \ i = 1, \ldots, m+q\}.$$

**Step 2.** For $i = 1, \ldots, n$ find solution $u^{ik} = (x^{ik}, y^{ik}, v^{ik})$ of linearized problem $(\mathcal{PL}(\bar{x}^i, \bar{v}^i, \rho_k))$.
**Step 3.** With the help of components $v^{ik}$ for $i = 1, \ldots n$ to find the new feasible points $p^{ik}$ in problem $(\mathcal{P})$.

**Step 4.** Find the point $p^k : F(p^k) = \min_{1 \le i \le n} F(p^{ik})$
**Step 5.** If $f(p^k) < \rho_k$ then $z^{k+1} := p^k, \ k := k+1$ and go to Step 1.
**Step 6.** If $f(p^k) \ge \rho_k$ then $z^{k+1} := z^k, \ \beta_{k+1} := \beta_k + \Delta\beta$ and go to Step 1.
**Step 7.** If $f(p^k) \ge \rho_k, \ \forall \beta \in [\beta_-, \beta_+]$ then **STOP:** $z := z^k$ is a solution of problem $(\mathcal{P})$.

## 4. Computational experiments

In this section we report on some computational experiments with the global search algorithm. The test problems were obtained with a method developed in [2]. The global algorithm was implemented in C++ and was run on a Pentium IV, 3.00GHz. Some results are presented in the table 1, in which we use the abbreviations: $m+n$ — size of problem; $constr$ — number of constraints; $Seria$ — number of problems in seria; $ls_{max}$ — maximal number of local minima in problems of seria; $Solv$ — number of solved problems in seria; $St$ — number of iterations; $Loc$ — number of critical points; $T_{av}$ — Average time (in sec.).

*Table 1.* Computational results for series of problems.

| $m+n$ | $constr$ | $Seria$ | $ls_{max}$ | $Solv$ | $St$ | $Loc$ | $T_{av}$ |
|---|---|---|---|---|---|---|---|
| 150 | 375 | 10 | $2^{75}$ | 10 | 57 | 6177 | 0:03:26.68 |
| 200 | 500 | 10 | $2^{100}$ | 10 | 51 | 6861 | 0:07:59.72 |
| 300 | 750 | 10 | $2^{150}$ | 10 | 103 | 16418 | 0:46:24.81 |
| 400 | 1000 | 10 | $2^{200}$ | 10 | 37 | 8607 | 0:47:42.74 |
| 500 | 1250 | 10 | $2^{250}$ | 10 | 20 | 5010 | 0:53:24.26 |
| 600 | 1500 | 10 | $2^{300}$ | 9 | 21 | 13018 | 1:56:33.12 |
| 800 | 2000 | 10 | $2^{400}$ | 10 | 24 | 17628 | 11:00:04.94 |
| 1000 | 2500 | 10 | $2^{500}$ | 10 | 24 | 20029 | 23:04:45.37 |

## 5. Conclusion

As a result, the global search algorithm was able to solve almost all the problems considered. These results show that the algorithm had a good behavior and can be used for solving linear bilevel problems.

**Acknowledgments.** The authors wish to thank Prof. A.S. Strekalovsky for his encouragement and support.

## References

[1] J.F. Bard Practical Bilevel Optimization. Dordrecht: Kluwer Academic Publishers, 1998.

[2] P. Calamai, L. Vicente Generating linear and linear-quadratic bilevel programming problems. *SIAM J. Scientific and Stat. Computing*, volume 14, pages 770–782, 1993.

[3] S. Dempe *Foundations of Bilevel Programming*, Dordrecht: Kluwer Academic Publishers, 2002.

[4] T.V. Gruzdeva, A.S. Strekalovsky Local Search in Problems with Nonconvex Constraints. *Comput. Math. Math. Phys.*, volume 47, pages 381–396. MAIK Nauka/Interperiodica, 2007.

[5] R. Horst, H. Tuy Global Optimization. Deterministic Approaches. Berlin: Springer-Verlag, 1993.

[6] F. Jia, F. Yang, S.-Y. Wang Sensitivity analusis in bilevel linear programming. *Systems Science and Mathematical Sciences*, volume 11, pages 359–366. 1998.

[7] A.S. Strekalovsky Minimizing Sequences in Problems with d.c. Constraints. *Comput. Math. Math. Phys.*, volume 45, pages 418–429. MAIK Nauka/Interperiodica, 2005.

[8] D. White, G. Anandalingam A penalty function approach for solving bi-level linear programs. *Journal of Global Optimization*, volume 3, pages 397–419. 1993.

# Air traffic conflict resolution via B-splines and genetic algorithms

Clement Peyronne[1,2,5], Daniel Delahaye[2], Laurent Lapasset[1], and Marcel Mongeau[4,5]

[1] *Capgemini, 15, Av. Docteur Maurice Grynfogel, 31000 Toulouse, France,* {clement.peyronne, llapasse}@capgemini.com

[2] *ENAC, Toulouse, France,* delahaye@recherche.enac.fr

[4] *CNRS; Institut de Mathematiques de Toulouse UMR 5219; F-31062 Toulouse, France* marcel.mongeau@math.univ-toulouse.fr

[5] *Universite de Toulouse; UPS, INSA, UT1, UTM; Institut de Mathematiques de Toulouse; F-31062 Toulouse, France.* marcel.mongeau@math.univ-toulouse.fr

**Keywords:** B-splines, Genetic algorithm, Conflict resolution

## 1. Introduction

From the beginning, the most critical point of Air Traffic Control was to ensure safety separation distance between airplanes. To achieve this goal, a safety standard separation has been defined : 5 Nm (Nautical miles) horizontally and 1000 feet vertically (*separation box*). Air traffic controllers are responsible for ensuring the respect of this separation rules.

However, air traffic has been constantly increasing and has already used all available ressources to increase airspace capacity. In the future, Air Traffic Management (ATM) will have to deal with a doubling of the air traffic while ensuring at least equivalent standards of safety [5]. The SESAR european project aims to find solutions to this problematic by automating the current system or by providing a decision support to the air traffic controllers in order to decrease their workload. Many works have been done on full automation, the most promising ones being Genetic Algorithm (GA) [1] and navigation function [2] methods. However, considering the technological advances on the airplane Flight Managment System (FMS) we will explore in this paper the possibility of a full automation generating continuous trajectories that new FMS can follow. In the first section, we detail the problem modeling (trajectory and optimization), in the second section we present our results and perspectives.

## 2. Problem modeling

### 2.1 Trajectory model

We choose a continuous model for representing trajectories. Considering the new FMS abilities, In order to obtain FMS fliable trajectories, we use B-splines, the approximation tool which provides interesting properties for our concerns. The primary B-spline objective was to find a curve interpolating a set of points of $R^2$ called *control points*. This objective was later extended to approximation, thereby avoiding the undesirable oscillation inherent to interpolation. In our study, we shall focus on this use of splines to approximate a set of control points. The *control polygon*, the linear curve linking the control points, completely defines the curve [3]. Indeed, the B-spline is to stay in the convex envelope of the control points.

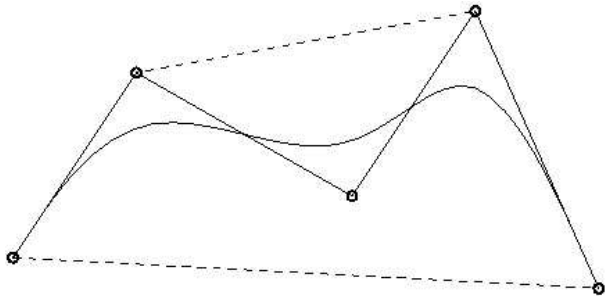


*Figure 1.* Control points, control polygon and the resulting B-spline

Basically, B-splines are parametrized curves generalizing the Bezier curve concept. It is an efficient approximation tool which is constructed from polynomial pieces joined at certain parameter's values called *knots*, stored in a *knot vector*. In a very simplified way, if we consider a set of control points $(X_i, Y_i) = P_i \in R^2 (i = 0 : n)$, and a parameter $u$, we can define the B-spline as follows:

$$C(u) = (\sigma_x(u), \sigma_y(u)), u \in [a, b]$$

where $\sigma_x(u)$ and $\sigma_y(u)$ are the B-splines approximations of the couples $(i, X_i)_{i=0:n}$ and $(i, Y_i)_{i=0:n}$ for $u \in [a, b]$.

We choose to rely on B-splines modeling trajectories because it is a very efficient tool for curve approximation in terms of both approximation quality and computational time. Moreover, B-splines feature interesting properties such as $\mathcal{C}^2$-continuity (crucial for modeling smooth aircraft trajectories, robustness and flexibility and the use flexibility (if one control point is displaced, only a small part of the curve will be affected).

## 2.2 Optimization method : Genetic Algorithms (GA)

When several aircraft are involved in a conflict, the conflict resolution problem has been shown to be NP-hard [1]. Moreover, the optimization variables being the B-splines control-point location, we shall see that our objective function (1) is not differentible with respect to these variables. Consequently, we must rely on black box (direct) optimization to address our problem. In this paper, we choose to use stochastic global optimization method : genetic algorithms.

To guide the control points location, we use classical genetic algorithms [7]. First, a population of individuals is generated in the state space. For each individual, we calculate the objective function (the *fitness*). Then, we select the best individuals according to their fitness and we randomly apply genetic operators (mutation, crossover). From this operation, a new population is created and we apply the same process again. Evolution between two generations is shown in Figure 2.

**Fitness evaluation : Conflict detection.**

In order to evaluate each individual fitness, we decode it into $N$ trajectory curves (one per aircraft) and we then evaluate two quantities. First, how many conflicts the situation creates and secondly what is the total extra distance engendered with respect to the direct routes. To calculate these quantities, we discretize the airspace into square cells of size half the standard separation. Our conflict detection is performed in two steps:

- First, for each airplane, we store the grid's cells through which the airplanes fly, the airplane number (its label), the entry and exit times in and out each of the stored cell

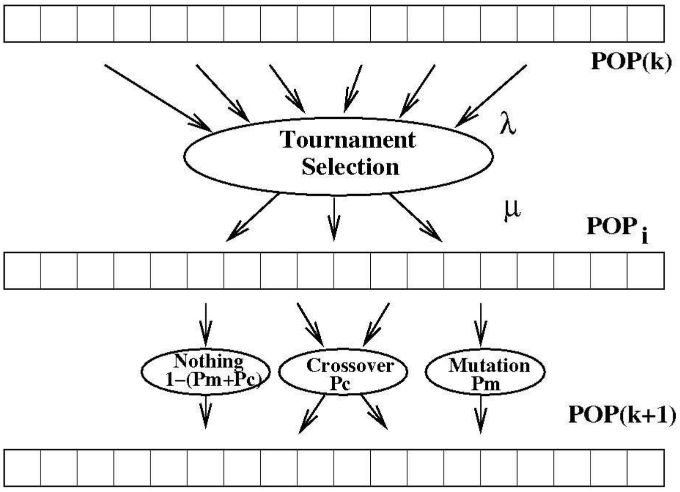


*Figure 2.* GA principle

- Then, we go through each stored cell and we check whether any other airplane goes through any of the eight neighbouring cells for other airplanes. If there are such airplanes, we check the time to see whether there is a conflict between these two airplanes. If so, we calculate the conflict duration.

Our conflict detection procedure send back the chromosome fitness to the evolutionary algorithm. Here is the formula we use to calculate the fitness:

$$f(X) = -(CN + (\frac{NR}{DR} - N) \tag{1}$$

Where $CN$ is the number of conflict, $NR$ the length of the new route calcultated by the algorithm, $DR$ the length of the direct route.and $N$ the number of plane. High fitness correponds to good individuals. Indeed, the lower are the number of conflict and the route lengthening, the better is the chromosome.

## 3. Computational results

In this section, we present results we obtained on a roundabout test proble. This problem consists in making 16 planes equidistributed on a circle of $100Nm(= 185200m)$ radius fly to the diametrically opposed point at a common speed (each point on the circle has an outgoing and an incoming trajectory).

For this configuration, our method obtains a conflict-free situation displayed in Figure 3. When the fitness is in $]-1,0]$ the situation is conflict-free (see (1)).

Although one can easily solve intuitively this academic problem due to its symetry, our implementation does not exploit any symetry here. This result, which we obtain automatically shows that our methodology is promising as the obtained conflict resolution is consistent with experts' advice.

## 4. Conclusion

We have shown in this paper that the combination of B-splines and genetic algorithms can be a promising methodology for automatic conflict resolution in air traffic control. More results are to come on different, various and more realistic situations. We also have in mind several developments to improve our approach like using the sharing (deals with equirepartition of

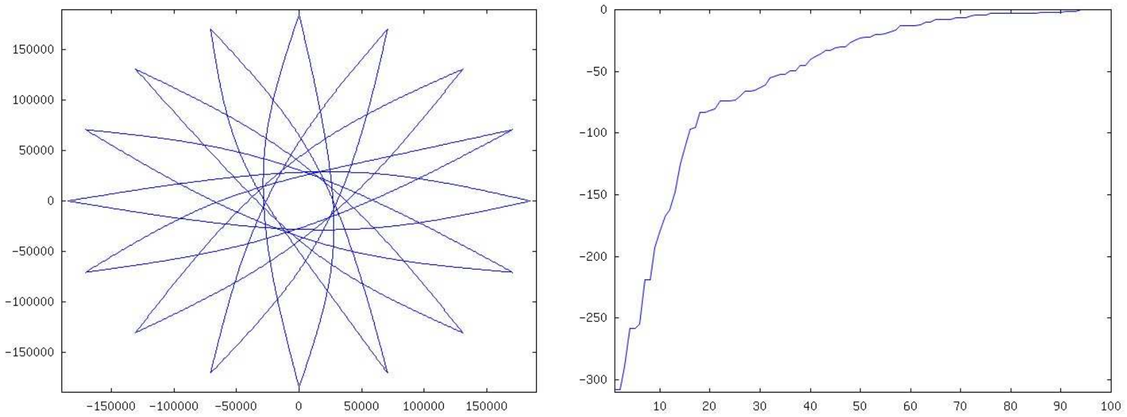


*Figure 3.* The roundabout configuration after resolution and the fitness evolution with respect to the generation number

the population on the different maximums) in our GA or implementing a *self-adaptative* GA (every parameter in the chromosome such as bandwidth, control points' number, etc will be considered within the chromosome encoding, as proper optimization variables. Furthermore, we plan to exploit our B-spline model of trajectory to address the conflict resolution problem with deterministic derivative-free optimization methods [4]. Indeed, despite the local aspect of these methods, they can also be adapted to global optimization.

# References

[1] N. Durand. *Algorithmes genetiques et autres outils d'optimisation appliques à la gestion du trafic aerien*. HDR thesis, Ecole Doctorale Informatique de Toulouse, 2004.

[2] K. Zeghal. *Vers une theorie de la coordination d'actions, application à la navigation aérienne*. PhD thesis, Universite Paris VI, 1994.

[3] C. Rabut. *Courbes, outils pour la CAO, approximation à une variable*. INSA Toulouse lecture. 2008.

[4] A.R. Conn, K. Scheinberg, L.N. Vicente. *Introduction to derivative-free optimization*. MPS-SIAM Series on Optimization, 2009.

[5] J-M. Alliot, D. Colin de Verdière (CENA) *L'ATM: 20 ans d'effort et perspectives*. Journée d'etudes : vers l'automatisation du vol et de sa gestion, Toulouse – Onera – 26 novembre 2003.

[6] D. Gianazza *Algorithme evolutionnaire et A∗ pour la separation en 3D des flux de trafic aerien* Journal Européen des systemes automatises. 2005.

[7] T. Baeck, D.B. Fogel, Z. Michalewicz. *Handbook of the evolutionnary computation*. Mathematics 1997.

# A two-level evolutionary algorithm for solving the leader-follower problem on the plane with variable demand*

J. L. Redondo[1], J. Fernández[2], A.G. Arrondo[1], I. García[3] and P.M. Ortigosa[1]

[1] *Dpt. of Computer Architecture and Electronics, University of Almería, Spain.*

[2] *Dpt. of Statistics and Operations Research, University of Murcia, Spain.*

[3] *Dpt. of Computer Architecture, University of Málaga, Spain.*

**Abstract** A chain (the leader) wants to set up a single new facility in a planar market where similar facilities of a competitor (the follower), and possibly of its own chain, are already present. The follower will react by locating another single facility after the leader locates its own facility. Both the location and the quality (design) of the new leader's facility are to be found. The aim is to maximize the profit obtained by the leader following the follower's entry. The demand is supposed to be concentrated at $n$ demand points, which split their buying power among the facilities proportionally to the attraction they feel for them. The attraction for a facility depends on both the location and the quality of the facility. In most competitive location literature it is assumed that the demand is deterministic, i.e., fixed regardless the conditions of the market. However, demand can vary depending on prices, distances to the facilities, etc. Taking variable demand into consideration increases the complexity of the problem and, therefore, the computational effort needed to solve it, but it makes some models more realistic. Two heuristic methods are proposed to cope with this hard-to-solve global optimization problem, namely, a grid search procedure and a two-level evolutionary algorithm.

**Keywords:** Nonlinear bi-level programming problem, centroid (or Stackelberg) problem, continuous location, competition, evolutionary algorithm

## 1. Introduction

Competitive location deals with the problem of locating facilities to provide a service (or goods) to the customers (or consumers) of a given geographical area where other competing facilities offering the same service are already present (or will enter the market in the near future). The scenario considered in this paper is that of a *duopoly*. A chain, the *leader*, wants to locate a new single facility in a given area of the plane, where there already exist $m$ facilities offering the same goods or product. The first $k$ of those $m$ facilities belong to the chain ($0 \leq k < m$) and the other $m - k$ to a competitor chain, the *follower*. The leader knows that the follower, as a reaction, will subsequently position a new facility too. The demand is supposed to be concentrated at $n$ demand points, whose locations $p_i$ and buying power are known. The location $f_j$ and quality of the existing facilities is also known. The aim is to maximize the profit obtained by the leader following the follower's entry. These types of bilevel programming problems were introduced by Hakimi, who introduced the terms *medianoid* for the follower problem, and *centroid* for the leader problem [2].

*This work has been funded by grants from the Spanish Ministry of Science and Innovation (TIN2008-01117, ECO2008-00667/ECON) and Junta de Andalucía (P06-TIC-01426, P08-TIC-3518), in part financed by the European Regional Development Fund (ERDF).

Demand points split their demand probabilistically over all facilities in the market in proportion to their attraction to each facility, determined by the different perceived qualities of the facilities and the distances to them. In this work it is considered that demand is elastic, that is, it varies depending on some factors. For instance, as already stated in [1], consumer expenditures on the products or services offered by the facilities may increase for a variety of reasons related to location of the new facility: opening new outlets may increase the overall utility of the product; the marketing expenditures resulting from the new facilities may increase the overall 'marketing presence' of the product, leading to increased consumer demand; or some consumers who did not patronize any of the facilities (perhaps because none were close enough to their location) may now be induced to do so. On the other hand, the quality of the facilities may also affect consumer expenditures since a better service usually leads to more sales. Taking variable demand into consideration increases even more the complexity of the problem and, therefore, the computational effort needed to solve it, but it makes the model more realistic.

The medianoid (follower) problem with variable demand (see [4] for an analytical description of the model) has been solved in [4] using the evolutionary algorithm UEGO (see [3]) and a steepest descent-type method (namely, a Weiszfeld-like algorithm). The resulting algorithm will be called UEGO_med throughout this paper. In this work two procedures for solving the centroid (leader) problem are studied, namely, a grid search procedure and a evolutionary algorithm. The results obtained by the evolutionary algorithm (which outperforms the grid search procedure) are promising.

## 2. Solving the centroid problem

Next two heuristics devised to cope with the leader problem are presented.

### 2.1 GS: a grid search procedure

The first method is a simple Grid Search procedure (GS). A grid of points that cover the leader's 3-dimensional searching region is generated. For each point of the grid we first check its feasibility. If it is feasible, then we evaluate the objective function. Notice that in order to do it, we first have to solve the corresponding medianoid problem to obtain an optimal solution for the follower. To this aim, the algorithm UEGO_med is used (see [4]). When all the feasible points of the grid have been evaluated, a second finer grid is constructed in the vicinity of the point of the first grid having the best objective value. In our first grid, the length of the step between two adjacent points was 0.1 units in each coordinate, and in the second grid, 0.02 units.

### 2.2 TLUEGO: A two-level evolutionary global optimization algorithm

TLUEGO is a evolutionary algorithm designed for solving the leader problem with variable demand. Although it is based on UEGO algorithm (see [3, 4] for a general description of UEGO), several modifications and new ideas have been introduced to deal with the problem at hand. In the following those modifications are briefly described.

**Species definition:** A species will be represented by a vector in the form $(nf_1, nf_2, r)$, where $nf_1$ refers to the leader point, $nf_2$ to the follower point, and $r$ to a radius, which specifies an attraction area of the species. There exists a relationship between $nf_1$ and $nf_2$: $nf_2$ is the solution of the medianoid problem when taking the original $m$ existing facilities and $nf_1$ into account.

**Species creation:** A species-list is maintained by the algorithm. Initially, a feasible point $nf_1$ is randomly chosen, and from it, the first species is constructed. Later on, in a given iteration, we do what follows. For every species in the species-list, we randomly generate feasible points for the leader's facility within the radius of attraction of the species. Then, for each pair of those points, we compute the midpoint of the segment connecting them. If the (approximate) objective value (its fitness value) for the leader problem at the midpoint is greater than at the extreme points, then the midpoint becomes a candidate point to generate a new species. Otherwise, if the value of the objective function at both extreme points is greater than at the midpoint then both extreme points become candidate points. The approximate objective values for the leader problem are computed inheriting the follower's facility of the species from which they were generated. Thus, for every species in the species-list we have a sublist of candidate points to generate new species.

Unfortunately, this process may generate a large number of candidate points. In order to reduce it, the candidate points are fused as described in the next paragraph. After that, for each candidate point in this reduced list we compute its corresponding follower's facility (applying UEGO_med) and then evaluate the correct objective value for the leader's facility. The new species (with the corresponding radius according to the iteration) are inserted in the species-list.

**Fuse process:** If at iteration $i$ the leader's facilities of two species from the species-list are closer to each other than a given radius $r_i$, then the two species are fused. The new leader's facility will be the one with the best objective value, and the follower's facility will be the corresponding one. Its radius will be the largest of the original species.

**Optimization process:** For every species in the list a local optimization process is applied. Notice that a local optimizer usually assumes that the configuration of the problem during the optimization process does not change. However, for the centroid problem this is not true, since every time that the leader's facility changes, so does the follower's facility. Thus, the value of the objective function of the leader's problem may change if the new configuration is taken into account. This means that the new follower's facility should be computed every time that the leader's facility changes. However, obtaining the exact new follower's facility using UEGO_med would make the process very time-consuming. That is why we have designed a new local optimization procedure. In such method the leader species is optimized using an hybrid heuristic method called SASS (*Single Agent Stochastic Search* [5]). In each iteration of SASS, the corresponding follower is computed using a steepest descent-type method (a Weiszfeld-like algorithm). After $15$ iterations, the follower is computed using UEGO_med and the optimized leader is evaluated considering this new follower as new facility. The optimization procedure is repeated again considering as original species the optimized one. If the objective function value of the final optimized species is better than the initial one, then the species will be replaced.

## 3. Computational studies

To study the performance of the algorithms, we have generated different problems, varying the number $m$ of existing facilities and the number $k$ of those facilities belonging to the leader's chain. In the preliminary studies presented here the number $n$ of demand points has been fixed to $15$ or $25$. The actual settings $(n, m, k)$ employed are detailed in Tables 1 and 2.

Since TLUEGO is a heuristic, each run may provide a different solution. To study the robustness of the algorithm, it has been run five times for each problem and average values have been computed. In the column labelled 'Av(Time)', the average time in the five runs (in seconds) is given, in 'BestSol' column the best solution $(x_1, x_2, \alpha)$ found in the five runs ($(x_1, x_2)$ is the location of the new facility, and $\alpha$ its quality), in 'MaxDist' column the maximum Euclidean distance between any pair of solutions provided by the algorithm (this gives an idea of how far the solutions provided by the algorithm in different runs can be), in the next three columns the minimum, the average and the maximum objective value in the five runs are given, and in 'Dev' column the standard deviation. GS has been run once, and the values obtained (CPU time, solution and objective value) are given in Table 2.

*Table 1.* Average results for all the problems solved by the algorithm TLUEGO.

| $(n, m, k)$ | Av(Time) | BestSol | MaxDist | Min | Av | Max | Dev |
|---|---|---|---|---|---|---|---|
| (15,2,1) | 1446 | (7.860,7.841,0.50) | 0.000 | 38.732 | 38.732 | 38.732 | 0.000 |
| (15,5,2) | 1204 | (5.713,2.342,0.50) | 0.078 | 39.880 | 39.947 | 39.972 | 0.035 |
| (15,10,4) | 4737 | (2.273,0.487,0.50) | 0.000 | 38.323 | 38.323 | 38.323 | 0.000 |
| (25,2,1) | 3375 | (7.066,7.225,4.95) | 0.237 | 60.905 | 60.995 | 61.053 | 0.051 |
| (25,5,2) | 3847 | (6.152,2.006,2.44) | 0.306 | 46.689 | 46.715 | 46.740 | 0.020 |
| (25,10,4) | 3290 | (0.486,4.980,3.19) | 0.264 | 62.247 | 62.378 | 62.477 | 0.081 |

As we can see, TLUEGO provides better results than GS. And although GS also gives good results, it is rather time-consuming. Moreover, there is no guarantee that GS can find a good approximation of the global optimum. If the objective function value increases dramatically in a small neighbourhood around the global optimum and the grid is not dense enough, the second finer grid can focus around a local optimum. Something similar can happen when a local optimum exists whose objective value is close to the global optimum value and the grid is not dense enough. The risk of failure is even higher in the presence of constraints, as happens in our centroid problem, since it may occur that the global optimum is surrounded (in part) by infeasible areas, and the grid may not have a feasible point near to the global optimum. In fact, we have used GS only as a safeguard to check the goodness of TLUEGO, and also because it allows us to investigate the difficulty of the problem at hand and to draw the graphs of the objective function projected in both the location and the quality spaces.

On the other hand, TLUEGO is rather robust, in the sense that in all the runs it obtains the same solution (except for the precision employed in the stopping criterion). This can be seen looking at the low values of MaxDist and Dev. Since UEGO was designed to scape from local minima and look over the whole feasible set, we can conclude that TLUEGO is a suitable algorithm for solving the centroid problem introduced in this paper.

## 4. Conclusions

In this study we have dealt with the $(1|1)$-centroid (Stackelberg or Simpson) problem with variable demand. Two heuristics have been introduced for handling the problem, namely, a grid search procedure and a evolutionary algorithms. The computational studies have shown that the evolutionary algorithm TLUEGO provides better results than GS, and using less computational effort.

*Table 2.* Results for all the problems solved by the algorithm Grid Search

| $(n, m, k)$ | Time | Sol | Obj |
|---|---|---|---|
| (15,2,1) | 21595 | (7.860,7.840,0.50) | 38.730 |
| (15,5,2) | 20958 | (5.720,2.340,0.50) | 39.952 |
| (15,10,4) | 29985 | (2.260,0.480,0.50) | 38.304 |
| (25,2,1) | 40931 | (7.080,7.240,4.75) | 59.912 |
| (25,5,2) | 40762 | (6.140,2.000,2.25) | 46.388 |
| (25,10,4) | 54347 | (0.480,4.980,2.75) | 61.836 |

## References

[1] O. Berman and D. Krass. Locating multiple competitive facilities: Spatial interaction models with variable expenditures. *Annals of Operations Research*, 111(1):197–225, 2002.

[2] S.L. Hakimi. On locating new facilities in a competitive environment. *European Journal of Operational Research*, 12(1):29–35, 1983.

[3] J.L.Redondo. *Solving competitive location problems via memetic algorithms. High performance computing approaches*. PhD thesis, Universidad de Almería, 2008. Available at `http://www.um.es/geloca/gio/tesisJuani.pdf`

[4] J.L. Redondo, J. Fernández, A.G. Arrondo, I. García, and P.M. Ortigosa. Deterministic or variable demand? Does it matter when locating a facility?. Submitted.

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

# On a CP approach to solve a MINLP inventory model*

Roberto Rossi,[1] S. Armagan Tarim,[2] Brahim Hnich,[3] Steven Prestwich[4] and Eligius M. T. Hendrix[5]

[1] *Logistics, Decision and Information Sciences, Wageningen UR, the Netherlands,* roberto.rossi@wur.nl

[2] *Department of Management, Hacettepe University, Turkey,* armtar@yahoo.coml

[3] *Faculty of Computer Science, Izmir University of Economics, Izmir, Turkey,* brahim.hnich@ieu.edu.tr

[4] *Cork Constraint Computation Centre, University College, Cork, Ireland,* s.prestwich@4c.ucc.ie

[5] *Computer Architecture, Málaga University, Spain,* eligius.hendrix@wur.nl

**Abstract** One of the most important policies adopted in inventory control is the replenishment cycle policy. Such a policy provides an effective means of dampening planning instability and coping with demand uncertainty. We describe a constraint programming approach for computing optimal replenishment cycle policy parameters under non-stationary stochastic demand, ordering, holding and shortage costs. Our solution approach exploits the convexity of the cost-function to dynamically compute during search the cost associated with a given decision variable assignment. By using our model we gauge the quality of an existing approximate mixed integer linear programming approach that exploits a piecewise linear approximation for the complex cost function. Furthermore, our computational experience shows that our approach can solve realistic instances in a fraction of a second.

**Keywords:** constraint programming, replenishment cycle policy, non-stationary demand, shortage cost, MINLP

## 1. Introduction

Much of the inventory control literature concerns the computation of optimal replenishment policies under demand uncertainty. One of the most important policies adopted is the ($R$,$S$) policy (also known as the *replenishment cycle* policy). A detailed discussion on the characteristics of ($R$,$S$) can be found in de Kok [3]. In this policy a replenishment is placed every $R$ periods to raise the inventory position to the order-up-to-level $S$. An important extension for existing stochastic production/inventory control problems consists of incorporating a non-stationary demand process. Under this assumption the ($R$,$S$) policy takes the non-stationary form ($R_n$,$S_n$) where $R_n$ denotes the length of the $n^{th}$ replenishment cycle and $S_n$ the corresponding order-up-to-level. To compute the *near* optimal ($R_n$,$S_n$) policy values, [5] apply a mixed integer programming (MIP) formulation using a piecewise linear approximation to a complex cost function with fixed procurement cost each time a replenishment order is placed. So far no complete approach exists for computing ($R_n$,$S_n$) policy parameters under a shortage cost scheme. In fact, [5] show that the cost structure is complex in this case and differs significantly from the one under a service level constraint. In this work, we give an *exact*

*This work has been funded by grants from the Spanish Ministry of Science and Innovation (TIN2008-01117). Eligius Hendrix is a fellow of the Spanish "Ramón y Cajal" contract program, co-financed by the European Social Fund. B. Hnich and A. Tarim are supported by the Scientific and Technological Research Council of Turkey under Grant No. SOBAG-108K027. A. Tarim is supported by Hacettepe University-BAB.

constraint programming [1] (CP) formulation of the ($R_n$,$S_n$) inventory control problem under shortage cost scheme. Our approach embeds dedicated cost-based filtering methods [4] to improve performances of the search. Our contribution is two-fold: we can now efficiently obtain provably optimal solutions for the MINLP ($R_n$,$S_n$) inventory control problem under shortage costs and we can gauge the accuracy of the piecewise linear approximation proposed by [5]. Computational results shows the effectiveness of our approach.

## 2. From a stochastic to a deterministic equivalent model

We consider the single stocking location, single product inventory problem over a finite planning horizon of $N$ periods. The demand $d_t$ in period $t$ is assumed to be a normally distributed random variable with known probability density function (PDF) $g_t(d_t)$. Demand is assumed to occur instantaneously at the beginning of each period. The mean rate of demand may vary from period to period. Demands in different time periods are assumed to be independent. Demands occurring when the system is out of stock are assumed to be back-ordered and satisfied as soon as the next replenishment order arrives.

In the general multi-period production/inventory problem with stochastic demands we are interested in finding the timing of the stock reviews and the size of non-negative replenishment orders, $X_t$ in period $t$, which minimize the expected total cost over a finite planning horizon of $N$ periods:

$$\min\ E\{TC\} =$$

$$\int_{d_1}\int_{d_2}\cdots\int_{d_N}\sum_{t=1}^{N}\left(a\delta_t + vX_t + hI_t^+ + sI_t^-\right) g_1(d_1)\ldots g_N(d_N)\mathrm{d}(d_1)\ldots\mathrm{d}(d_N) \tag{1}$$

subject to, for $t = 1\ldots N$,

$$X_t > 0 \Rightarrow \delta_t = 1 \tag{2}$$

$$I_t = \sum_{i=1}^{t}(X_i - d_i) \tag{3}$$

$$I_t^+ = \max(0, I_t) \tag{4}$$

$$I_t^- = -\min(0, I_t) \tag{5}$$

$$X_t, I_t^+, I_t^- \in \mathbb{R}^+ \cup \{0\}, \quad I_t \in \mathbb{R}, \quad \delta_t \in \{0,1\} \tag{6}$$

where

$d_t$ : the demand in period $t$, a normal random variable with PDF $g_t(d_t)$,
$a$ : the fixed ordering cost,
$v$ : the proportional direct item cost,
$h$ : the proportional stock holding cost,
$s$ : the proportional shortage cost,
$\delta_t$ : a {0,1} variable that takes the value of 1 if a replenishment occurs in period $t$ and 0 otherwise,
$I_t$ : the inventory level at the end of period $t$, $-\infty < I_t < +\infty$, $I_0 = 0$
$I_t^+$ : the excess inventory at the end of period $t$ carried over to the next period,
$I_t^-$ : the shortages at the end of period $t$, or magnitude of negative inventory,
$X_t$ : the replenishment order placed and received in period $t$, $X_t \geq 0$.

Let $R(i,j)$ denote a *replenishment cycle* that schedules an inventory review at period $i$ to cover subsequent demand up to period $j$ with cost as a function of the opening inventory level $S$:

$$\sum_{k=i}^{j}\left(hz_{\alpha(i,k)}\sigma_{i,k} + (h+s)\sigma_{i,k}[\phi(z_{\alpha(i,k)}) - (1-\alpha(i,k))z_{\alpha(i,k)}]\right), \tag{7}$$

where $G_{i,k}(\cdot)$ and $\sigma_{i,k}$ denote, respectively, the cumulative distribution function and the standard deviation of $d_i + \ldots + d_k$; $\alpha(i,k) = G_{i,k}(S)$; and $z_{\alpha(i,k)} = \Phi^{-1}(\alpha(i,k))$. $\Phi(\cdot)$ and $\phi(\cdot)$ denote, respectively, the standard normal CDF and PDF. Therefore we have $j - i + 1$ cost components: the holding and shortage costs at the end of period $i, i+1, \ldots, j$. Since we can prove that the cost function to be convex, for each possible replenishment cycle we can efficiently compute the optimal $S^*$ that minimizes it by using gradient based methods for convex optimization.

## 3. CP approach to solve the deterministic equivalent model

A *deterministic equivalent* (see [2]) CP formulation of the model is constructed. The expected total cost for $R(i,j)$ with opening inventory level $S_i$, is iteratively computed by a special-purpose constraint `objConstraint`$(\cdot)$ which uses a slightly extended version of Eq. (7). Intuitively, within this constraint the expected total cost for a certain replenishment plan will be computed as the sum of all the expected total costs for replenishment cycles in the solution, plus the respective ordering costs.

A *deterministic equivalent* CP formulation is then

$$\begin{aligned} \min \quad & E\{TC\} = C & (8) \\ \text{subject to} \quad & \texttt{objConstraint}\left(C, \tilde{I}_1, \ldots, \tilde{I}_N, \delta_1, \ldots, \delta_N, d_1, \ldots, d_N, a, h, s\right) & (9) \\ \text{and for } t = 1, \ldots, N \quad & \tilde{I}_t + \tilde{d}_t - \tilde{I}_{t-1} \geq 0 & (10) \\ & \tilde{I}_t + \tilde{d}_t - \tilde{I}_{t-1} > 0 \Rightarrow \delta_t = 1 & (11) \\ & \tilde{I}_t \in \mathbb{R}, \quad \delta_t \in \{0,1\}. & (12) \end{aligned}$$

Decision variable $\tilde{I}_t$ represents the expected closing inventory level at the end of period $t$ and $\tilde{d}_t$ represents the expected value of demand in a given period $t$.

The objective function (8) gives expected total cost over the planning horizon. The function `objConstraint`$(\cdot)$ dynamically computes buffer stocks and assigns to $C$ the expected total cost related to a given assignment for replenishment decisions, depending on the demand distribution in each period. In order to propagate `objConstraint`$(\cdot)$, during the search we wait for a partial assignment involving some or all $\delta_t$ variables. We look for an assignment where there exists some $i$ s.t. $\delta_i = 1$, some $j > i$ s.t. $\delta_{j+1} = 1$ and for every $k$, $i < k \leq j$, $\delta_k = 0$. This uniquely identifies a replenishment cycle $R(i,j)$.

There may be more replenishment cycles associated to a partial assignment. If we consider each $R(i,j)$ identified by the current assignment, it is easy to minimize the convex cost function already discussed, and to find the optimal expected closing inventory level $\tilde{I}_j$ for this particular replenishment cycle independently of the others.

By independently computing the optimal optimal expected closing inventory level for every replenishment cycle identified, two possible situations may arise: (i) the closing inventory levels obtained satisfies every inventory conservation constraint (Eq. (10)); or (ii) for some couple of subsequent replenishment cycles this constraint is violated. In other words, we observe an expected negative order quantity. If the latter situation arises, we can adopt a fast convex optimization procedure to compute a feasible buffer stock configuration with minimum cost.

The algorithm for computing optimal buffer stock configurations in presence of negative order quantity scenarios simply exploits the linear dependency between the opening inventory level of the second cycle and the expected closing inventory level of the first cycle. Due to this dependency the overall cost is still convex; we can apply convex optimization to find the optimal buffer stock configuration. Note that this reasoning still holds in a recursive process. Therefore, we can optimize buffer stock for two subsequent replenishment cycles, then we can treat these as a new single replenishment cycle, since their buffer stocks are linearly dependent, and repeat the process in order to consider the next replenishment cycle if a negative order quantity scenario arises. A lower bound for the expected total cost associated to the current partial assignment for $\delta_t$, $t = 1, \dots, N$ variables is now given by the sum of all the cost components $C(S_i, i, j)$, for each replenishment cycle $R(i, j)$ identified by the assignment.

## 4. Numerical experience

The described approach was first compared to the one descibed in and the set of instances presented in [5]. For these instances, a piecewise linear approximation with seven segments usually provides a solution with a cost reasonably close to optimal (<1% difference); while using two segments produces a cost difference up to about 7%. The next experiment concerns a single problem with period demands generated from seasonal data with no trend: $\tilde{d}_t = 50[1 + \sin(\pi t/6)]$. In addition to the "no trend" case (P1) we also consider three others: (P2) positive trend case, $\tilde{d}_t = 50[1+\sin(\pi t/6)]+t$; (P3) negative trend case, $\tilde{d}_t = 50[1+\sin(\pi t/6)]+(52-t)$; (P4) life-cycle trend case, $\tilde{d}_t = 50[1 + \sin(\pi t/6)] + \min(t, 52 - t)$.
In each test we assume an initial null inventory level and a normally distributed demand for every period with a coefficient of variation $\sigma_t/\tilde{d}_t$ for each $t \in \{1, \dots, N\}$, where $N$ is the length of the considered planning horizon. We performed tests using four different ordering cost values $a \in \{50, 100, 150, 200\}$ and two different $\sigma_t/\tilde{d}_t \in \{1/3, 1/6\}$. The planning horizon length takes even values in the range $[20, 38]$. The holding cost used in these tests is $h = 1$ per unit per period. Our tests also consider two different shortage cost values $s = 15$ and $s = 25$. Direct item cost is $v = 2$ per unit produced.

Our CP approach generally requires only a fraction of a second to produce the optimal solution. Only in 6 instances the optimal solution was not produced within the given time limit of 5 seconds. Nevertheless, it should be also noted that the worst case running time for our approach over the whole test bed was $6,77$ minutes. Therefore even in the few cases in which an optimal solution is not found in a less than 5 seconds, our CP model provides a reasonable running time.

## 5. Conclusions

We developed a constraint programming approach able to compute optimal replenishment cycle policy parameters under non-stationary stochastic demand, ordering, holding and shortage costs. In our model we exploited the convexity of the cost-function during the search to dynamically compute the cost associated with a given decision variable assignment. By using our approach we assessed the quality of an existing approximate mixed integer linear programming approach that exploits a piecewise linear approximation for the complex cost function. Our computational experience show the effectiveness of our approach.

## References

[1] K. Apt. *Principles of Constraint Programming.* Cambridge University Press, Cambridge, UK, 2003.

[2] J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming.* Springer Verlag, New York, 1997.

[3] A. G. de Kok. Basics of inventory management: part 2 The (R,S)-model. Research memorandum, FEW 521, 1991. Department of Economics, Tilburg University, Tilburg, The Netherlands.

[4] F. Focacci, A. Lodi, and M. Milano. Cost-based domain filtering. In *Proceedings of the 5th International Conference on the Principles and Practice of Constraint Programming*, pages 189–203. Springer Verlag, 1999. Lecture Notes in Computer Science No. 1713.

[5] S. A. Tarim and B. G. Kingsman. Modelling and Computing ($R^n$,$S^n$) Policies for Inventory Systems with Non-Stationary Stochastic Demand. *European Journal of Operational Research*, 174:581–599, 2006.

# Applications of the Global Optimization Methods for the Solution of Truss Structures with Interval Parameters

Iwona Skalna[1] and Andrzej Pownuk[2]

[1] *Department of Applied Computer Science, AGH University of Science and Technology, ul. Gramatyka 10, Cracow, Poland,* skalna@agh.edu.pl

[2] *Department of Mathematical Sciences, The University of Texas at El Paso, 500 West University Avenue, El Paso, TX 79968-0514, USA,* andrzej@pownuk.com

**Abstract** The problem of solving parametric linear systems whose input data are functions of interval parameters is considered. The goal is to compute a tight enclosure for the solution set of such systems. Several techniques are employed to reach this goal. Sensitivity analysis is compared with evolutionary optimization method and interval global optimization. Evolutionary optimization is used both to approximate the hull from below and to obtain the starting point for global optimization. Several acceleration techniques are used to speed up the convergence of the global optimization. Additionally, the parallel computations are involved. Some illustrative examples are solved by the discussed methods. It is shown that interval global optimization can be successfully used for solving the problems under consideration. All optimization methods described in this paper are parallelizable and can be implemented by using MPI Library.

**Keywords:** Interval Global Optimization, Sensitivity Analysis, Parametric Linear Systems, Parallel Programming

## 1. Interval equations

Consider the interval equation

$$F(u, \boldsymbol{p}) = 0 \ , \tag{1}$$

where $\boldsymbol{p} = [\underline{p}_1, \overline{p}_1] \times ... \times [\underline{p}_m, \overline{p}_m]$, $u = (u_1, ..., u_n)$, and $F = (F_1, ..., F_n)$. Function $F$ can be very complicated. One can consider a system of algebraic, differential, integral equations, in general, any type of equations including relational ones.

The solution set of system (1) is defined as

$$u(\boldsymbol{p}) = \{u : F(u, p) = 0, p \in \boldsymbol{p}\} \tag{2}$$

Generally, the solution set $u(\boldsymbol{p})$ has a very complicated shape (it is not necessarily convex). Therefore, the problem of solving system of equations (1) is usually formulated as a problem of finding an interval vector (outer solution) that contains the solution set. The tightest interval solution is called a hull solution [5] or an *optimal solution* [13]. The problem of computing hull solution can be defined as a family of $2n$ global optimization problems:

$$\underline{u}_i = \begin{cases} \min u_i \\ F(u, p) = 0 \\ p \in \boldsymbol{p} \end{cases} , \quad \overline{u}_i = \begin{cases} \max u_i \\ F(u, p) = 0 \\ p \in \boldsymbol{p} \end{cases} , \quad i = 1, \ldots, n, \tag{3}$$

and the following theorem holds.

**Theorem 1.** *Let $F(u, \boldsymbol{p}) = 0$ and let $\underline{u}_i$ and $\overline{u}_i$ denote, respectively, the solution of the i-th minimization and maximization problem (3). Then*

$$\boldsymbol{u} = \square u(\boldsymbol{p}) = \square\{u : F(u, p) = 0, p \in \boldsymbol{p}\} = [\underline{u}_1, \overline{u}_1] \times ... \times [\underline{u}_m, \overline{u}_m]. \tag{4}$$

## 2. Monotonicity and uniform monotonicity

Function $u = u(p_1, p_2, ..., p_m)$ is monotonically increasing with the respect to the variable $p_i$ if

$$p_{i0} \geqslant p_{i1} \Rightarrow u(..., p_{i0}, ...) \geqslant u(..., p_{i1}, ...) \tag{5}$$

Function $u = u(p_1, p_2, ..., p_m)$ is monotonically decreasing with the respect to the variable $p_i$ if

$$p_{i0} \geqslant p_{i1} \Rightarrow u(..., p_{i0}, ...) \leqslant u(..., p_{i1}, ...) \tag{6}$$

If the function is monotonically increasing or monotonically decreasing then the function is monotone.

**Theorem 2.** *If the function is monotone with respect to all variables $p_1, ..., p_m$ then extreme values of the function $u = u(p_1, ..., p_m)$ are attained at vertices of the box* $\mathbf{p}$.

In the case of truss structures it is possible to prove that the implicit function $u = u(p_1, ..., p_m)$, which is defined by the equation (2) is monotone [6]. Because of that the following theorem is true.

**Theorem 3.** *In the case of truss structures, extreme values of the displacements $u$ are attained at the vertices of the box* $\mathbf{p}$*, where* $\mathbf{p}$ *contains only area of cross-section, Young modulus, and point loads [6].*

In order to get extreme values of the displacements $u = u(p_1, ..., p_m)$ it is possible to apply endpoint combination method [5]. That is the practical conclusion of the theorem 3. Unfortunately endpoint combination method is very time-consuming, because of that it is not possible to use that method in practice.

**Definition 4.** *If a function $u = u(p) = u(p_1, ..., p_m)$ is monotone with respect to all variables $p_1, ..., p_m$ for all $p \in \boldsymbol{p}$, then $u$ is* ***uniformly monotone****.*

**Theorem 5.** *If the function is uniformly monotone, then extreme values can be calculated by using one iteration of the gradient method (sensitivity analysis [9]).*

According to the numerical experiments [12], displacements of some truss structures are uniformly monotone and some are not.

For monotone functions $u_i = u_i(p)$ the maximum and the minimum can be found by using the following procedure

$$If\ \ \frac{\partial u_i}{\partial p_j} \geqslant 0\ \ then\ \ p_j^{min,i} = \underline{p}_j,\ p_j^{max,i} = \overline{p}_j, \tag{7}$$

$$If\ \ \frac{\partial u_i}{\partial p_j} < 0\ \ then\ \ p_j^{min,i} = \overline{p}_j,\ p_j^{max,i} = \underline{p}_j, \tag{8}$$

$$\underline{u}_i = u_i(p^{min,i}),\ \ \overline{u}_i = u_i(p^{max,i}). \tag{9}$$

Derivatives $\frac{\partial u_i}{\partial p_j}$ and different interval solution can be calculated in parallel.

## 3. Gradient free method

One of the key aspects of gradient method, described in the previous sections, is calculation of derivatives. Unfortunately calculation of derivatives may be a very complex task in the complex computational methods. However, it is possible to simplify the process of calculations derivatives by using finite difference method [7].

$$\frac{\partial u_j}{\partial p_i} \approx \frac{u_j(..., p_i + \Delta p_i, ...) - u_j(..., p_i, ...)}{\Delta p_i} \tag{10}$$

It is also possible to apply higher order and multi-point finite difference schemas. Numerical approximation of the derivatives $\frac{\partial u_i}{\partial p_j}$ and different interval solution can be calculated in parallel.

## 4. Evolutionary algorithm description

Evolutionary optimization is one of the alternatives to the methods described above. Evolutionary computational techniques can deals with complex problems where the monotonicity assumption is not valid. To solve optimization problems (3), the following evolutionary strategy is applied. Individuals in the population are $k$-dimensional real vectors, where $k$ is a number of interval parameters. Elements of the initial population are generated at random based on the uniform distribution. The $10\%$ of the best individuals pass to the next generation, and the rest of population is generated using the non-uniform mutation and arithmetic crossover. The parents are selected using tournament selection of size 2.

Parallel implementations of evolutionary algorithms come in two group. Coarse grained parallel genetic algorithms assume a population on each of the computer nodes and migration of individuals among the nodes. Fine grained parallel genetic algorithms assume an individual on each processor node which acts with neighbouring individuals for selection and reproduction. Other variants, like genetic algorithms for online optimization problems, introduce time-dependence or noise in the fitness function.

## 5. Global optimization method

The strategy described in Section 4 produces very good inner approximation of the actual hull solution. In order to get reliable solution, an interval global optimization method can be applied [15]. The main deficiency of the global optimization is its high computational cost. To cope with this problem various inclusion functions are employed, several acceleration techniques such as monotonicity test, cut-off test based on the result of evolutionary optimization are used, and the technique which deals with parallel computations is applied. Many acceleration techniques can be applied in parallel.

## 6. Summary

In this paper different optimization methods are applied for solution of system of equations with the interval parameters. The objective is to find the best optimization algorithm that can be applied for each specific problem.

Gradient methods are very fast and they give the exact results if the problem is uniformly monotone. Unfortunately, very often the problems are not uniformly monotone and in that case gradient methods give a very good inner estimation of the optimal solution.

Evolutionary algorithms are stochastic optimization methods which can be applied in situations where the gradient method (based on monotonicity assumption) gives inaccurate results. The evolutionary optimization result also approximates the optimal solution from below. However, usually this approximation is more accurate for non-monotone problems. Moreover, the evolutionary optimization result can be used to perform an efficient cut-off test for global optimization.

In order to get reliable solution special global optimization method is proposed [15]. Suggested acceleration techniques significantly reduce the computational time of global optimization.

According to numerical results, the gradient (or gradient free) method give very accurate solution [12] for many problems of structural mechanics. However, there is also a large class of problems for which monotonicity assumption is not acceptable. In that cases different optimization methods such as evolutionary optimization or interval global optimization can be applied.

## References

[1] Floudas Ch.A. Deterministic Global Optimization. Theory, Methods and Applications Kluwer Academic Publisher, London, 2000.

[2] Hansen E.R. Global Optimization Using Interval Analysis Marcel Dekker, New York, 1992.

[3] Horst R. and Pardalos P.M. (eds.) Handbook of Global Optimization Kluwer, Dordrecht, 1995.

[4] Moore R.E. Interval Analysis Prentice-Hall, Englewood Cliffs, 1966.

[5] Neumaier A. Interval Methods for Systems of Equations Cambridge University Press, 1990.

[6] Neumaier A. and Pownuk A. Linear Systems with Large Uncertainties with Applications to Truss Structures. Reliable Computing, 13(2):149–172, 2007.

[7] Powell M.J.D. An efficient method for finding the minimum of a function of several variables without calculating derivatives. Computer Journal, 7:152-162, 1964.

[8] Pownuk A. Optimization of mechanical structures using interval analysis Computer-Assisted Mechanics and Engineering Sciences, 7(4):551–558, 2000.

[9] Pownuk A. Numerical solutions of fuzzy partial differential equation and its application in computational mechanics. Fuzzy Partial Differential Equations and Relational Equations: Reservoir Characterization and Modeling (M. Nikravesh, L. Zadeh and V. Korotkikh, eds.), Studies in Fuzziness and Soft Computing, Physica-Verlag, 2004.

[10] Pownuk A. General Interval FEM Program Based on Sensitivity Analysis, NSF workshop on Reliable Engineering Computing, February 20-22, 2008, Savannah, Georgia, USA, pp.397-428

[11] Pownuk A. and Martinez J.T. and De Morales M.H. and Pinon R. Application of the sensitivity analysis to the solution of some differential equations with the interval parameters - list of examples The University of Texas at El Paso, Department of Mathematical Sciences Research Reports Series Texas Research Report No. 2009-02, El Paso, Texas, USA http://andrzej.pownuk.com/publications/2009-Pownuk-Research-Report-2009-2.pdf

[12] Pownuk A. Monotonicity of the solution of the interval equations of structural mechanics - list of examples The University of Texas at El Paso, Department of Mathematical Sciences Research Reports Series Texas Research Report No. 2009-01, El Paso, Texas, USA, http://andrzej.pownuk.com/publications/2009-Pownuk-Research-Report-2009-1.pdf, 2009.

[13] Shary S.P. On optimal solution of interval linear equations SIAM Journal on Numerical Analysis 32(2):610–630, 1995.

[14] Skalna I. A Method for Outer Interval Solution of Systems of Linear Equations Depending Linearly on Interval Parameters Reliable Computing, 12(2):107–120, 2005

[15] Skalna I. A Global Optimization Method for Solving Parametric Linear Systems Whose Input Data are Rational Functions of Interval Parameters Lecture Notes in Computer Science, 2009.

[16] Zienkiewicz O.C. and Taylor R.L. and Zhu J.Z. The Finite Element Method its Basis and Fundamentals Butterworth-Heinemann, 2005.

Proceedings of TOGO 2010, pp. 145 – 148.

# Global Convergence and the Powell Singular Function

Trond Steihaug and Sara Suleiman

*Department of Informatics, University of Bergen, P. Box 7803, 5020 Bergen, Norway,*
{Trond.Steihaug, Sara.Suleiman}@ii.uib.no

**Abstract** The Powell singular function was introduced 1962 by M.J.D. Powell as an unconstrained optimization problem. The function is also used as nonlinear least squares problem and system of nonlinear equations. The function is a classic test function included in collections like MINPACK, Hoch and Schittkowski and CUTE as well as an example problem in text books. In the global optimization literature the function is stated as a difficult test case. The function is convex and the Hessian has a double singularity at the solution. In this paper we consider Newton's method and methods in the Halley class. While Newton's method is locally convergent with second order rate of convergence all methods in the Halley Class have third order rate of convergence. In this talk we will discuss the relationship between these method and consider the Powell Singular Function. We show that these methods have global convergence. However, they all have a linear rate of convergence. We will illustrate these properties with numerical experiments.

**Keywords:** System of nonlinear equations, global optimization, Powell singular, high order method.

## 1. Introduction

Consider the system of nonlinear equations $F(x) = 0$ where the function $F : \mathbb{R}^n \to \mathbb{R}^n$ and $F = (f_1, \ldots, f_n)^T$. In the following we assume that the function is at least two times continuously differentiable. The Jacobian matrix or the first derivative $F'(x) \in \mathbb{R}^{n\times n}$ is an $n\times n$ matrix where component $(i, j)$ is $\frac{\partial f_i(x)}{\partial x_j}$, and the second derivative $F''(x) \in \mathbb{R}^{n\times n\times n}$ is an $n \times n \times n$ tensor where component $(i, j, k)$ is $\frac{\partial^2 f_i(x)}{\partial x_k \partial x_j}$. The classical iterative method to solve $F(x) = 0$ is the Newton method. For a given iterate $x^k$ the correction is given by

$$s^k_{(1)} = -\big(F'(x^k)\big)^{-1} F(x^k), \tag{1}$$

provided that the Jacobian matrix at $x^k$ is nonsingular and the new iterate is $x^{k+1} = x^k + s^k_{(1)},\ k \geq 0$. If $x^*$ is a solution and $F'(x^*)$ is nonsingular it is well known that if the starting point $x^0$ is sufficiently close to the solution, then Newton's method converges with a quadratic rate of convergence. The Halley class of methods [6] have local and third order rate of convergence. For a given value of $\alpha$ and a given iterate $x^k$, the new iterate of a method in the Halley class is given by

$$x^{k+1} = x^k + s^k_{(1)} + s^k_{(2)}, \qquad k \geq 0 \tag{2}$$

where $s^k_{(1)}$ is the Newton step (1) and $s^k_{(2)}$ is given by

$$s^k_{(2)} = -\frac{1}{2}\left(F'(x^k) + \alpha F''(x^k) s^k_{(1)}\right)^{-1} F''(x^k) s^k_{(1)} s^k_{(1)}. \tag{3}$$

In 1962 M.J.D. Powell [10] introduced the function to test an unconstrained optimization algorithm. The function goes under the names Powell Quartic function or Powell Singular function. Consider the nonlinear system of equations $F(x) = 0$ where $F : \mathbb{R}^4 \to \mathbb{R}^4$ is defined by

$$F(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \\ f_3(x) \\ f_4(x) \end{pmatrix} \equiv \begin{pmatrix} x_1 + 10x_2 \\ \sqrt{5}(x_3 - x_4) \\ (x_2 - 2x_3)^2 \\ \sqrt{10}(x_1 - x_4)^2 \end{pmatrix}, \tag{4}$$

and the objective function for unconstrained optimization

$$\begin{aligned} f(x) &= F(x)^T F(x) = f_1(x)^2 + f_2(x)^2 + f_3(x)^2 + f_4(x)^2 \\ &= (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4. \end{aligned} \tag{5}$$

The objective function is convex and the unique unconstrained minimizer and the unique solution to the nonlinear system of equations is $x^* = (0, 0, 0, 0)^T$. The Jacobian matrix of $F$ at the solution is singular and the Hessian matrix of $f$ at the solution is also singular. In this extended abstract we only consider the nonlinear system of equations and we use the formulation first discussed in [2]. However, the theorems can easily be modified for the unconstrained optimization case.

The objective function is a standard reference test for derivative free optimization methods [9, 8], for heuristic methods introduced in [13] as well as global optimization algorithms [15]. The test function is also a standard reference in textbooks [1, 13, 5, 12, 4, 14, 11]. The scalar function can be extended in many ways [3] for unconstrained problems with $n > 4$. The function (4) gives rise a difficult nonlinear system of equations [7].

To simplify the notation we will remove the iteration index and use that $x$ is the current iterate and $x^+$ is the next iterate.

## 2. The Newton Method

The first derivative of Powell Singular Function (4) is given by

$$F'(x) = \begin{bmatrix} 1 & 10 & 0 & 0 \\ 0 & 0 & \sqrt{5} & -\sqrt{5} \\ 0 & 2(x_2 - 2x_3) & -4(x_2 - 2x_3) & 0 \\ 2\sqrt{10}(x_1 - x_4) & 0 & 0 & -2\sqrt{10}(x_1 - x_4) \end{bmatrix}. \tag{6}$$

The determinant of Jacobian matrix $F'(x)$ is

$$\det(F'(x)) = 84\sqrt{50}(x_1 - x_4)(x_2 - 2x_3).$$

We immediately have that $F'(x)$ is nonsingular if and only if $f_3(x) \neq 0$ and $f_4(x) \neq 0$ using the definition (4). Assuming that $F'(x)$ is nonsingular, it can be shown that in the Newton step (1) we have $(F'(x))^{-1} F(x) = (I - N)x$ and the new iterate $x^+ = x + s_{(1)}$ is given by

$$x^+ = Nx, \tag{7}$$

where the matrix $N$ is

$$N = \frac{1}{42} \begin{bmatrix} 20 & -10 & 20 & -20 \\ -2 & 1 & -2 & 2 \\ -1 & -10 & 20 & 1 \\ -1 & -10 & 20 & 1 \end{bmatrix}. \tag{8}$$

The matrix $N$ has full set of eigenvectors and the eigenvalues are $\frac{1}{2}$ and $0$. The value of $F$ at the new iterate is

$$F\left(x^{+}\right) = \frac{1}{4}\begin{pmatrix} 0 \\ 0 \\ (x_2 - 2x_3)^2 \\ \sqrt{10}(x_1 - x_4)^2 \end{pmatrix}. \tag{9}$$

Thus the components $f_1(x^{+})$ and $f_2(x^{+})$ vanish for all $x$ and $f_i(x^{+}) = \frac{1}{4}f_i(x),\ i = 3, 4$. We can conclude that $\det(F'(x^{+})) \neq 0$ when $\det(F'(x)) \neq 0$. The following result for Newton's method can be derived from the above observations.

**Theorem 1.** *Let $x^0$ be any starting point so that $F'(x^0)$ is nonsingular then Newton's method is well defined for all $k \geq 0$ and given by (7). Further for any starting point the sequence of iterates $\{x_k\}$ defined in (7) is globally convergent and converges linearly to the solution $x^* = 0$ with quotient factor $\frac{1}{2}$.*

## 3. The Halley Class

For a method in the Halley class the new iterate $x^{+} = x + s_{(1)} + s_{(2)}$ where $s_{(1)}$ is the Newton step (1) and $s_{(2)}$ is defined by equation (3). It follows that

$$F'(x) - \alpha F''(x)\big(F'(x)\big)^{-1}F(x) =$$

$$(2-\alpha)\begin{bmatrix} \frac{1}{(2-\alpha)} & \frac{10}{(2-\alpha)} & 0 & 0 \\ 0 & 0 & \frac{\sqrt{5}}{(2-\alpha)} & \frac{-\sqrt{5}}{(2-\alpha)} \\ 0 & (x_2 - 2x_3) & -2(x_2 - 2x_3) & 0 \\ \sqrt{10}(x_1 - x_4) & 0 & 0 & -\sqrt{10}(x_1 - x_4) \end{bmatrix}, \tag{10}$$

where $\alpha$ is a given scalar. The determinant can be shown to be $105\sqrt{2}(\alpha-2)^2(x_1-x_4)(x_2-2x_3)$ so if $F'(x)$ is nonsingular, then $F'(x) - \alpha F''(x)\big(F'(x)\big)^{-1}F(x)$ is also nonsingular for $\alpha \neq 2$. It can be shown that $s_{(2)}$ can be written in terms of the matrix $N$ given by (8)

$$s_{(2)} = \frac{1}{2(\alpha - 2)}Nx. \tag{11}$$

Since $s_{(1)}$ is Newton step $x + s_{(1)} = Nx$ and $x^{+} = Nx + s_{(2)}$ substituting for (11) the next iterate $x^{+}$ is

$$x^{+} = \frac{2\alpha - 3}{2(\alpha - 2)}Nx. \tag{12}$$

For specific values of $\alpha$ in the Halley class we get some well known methods. Chebyshev's method, Halley's method and Super–Halley's method are obtained when $\alpha = 0, \frac{1}{2}$ and $1$ respectively. We can thus summarize the observations in the following convergence theorem.

**Theorem 2.** *Let $x^0$ be any starting point so that $F'(x^0)$ is nonsingular then methods in the Halley class are well defined for $\alpha < 11/6$ or $\alpha > 5/2$ and given by (12). Further for any starting point the sequence of iterates $\{x_k\}$ defined in (12) is globally convergent and converges linearly to the solution $x^* = 0$ with quotient factor $\left|\frac{2\alpha-3}{4(\alpha-2)}\right|$.*

## 4. Numerical Experiment

Consider PSF (4) solved by Newton, Chebyshev, Halley, and Super–Halley methods, and implemented using Matlab. Using the standard starting point $x^0 = (3, -1, 0, 1)^T$, with the stop-

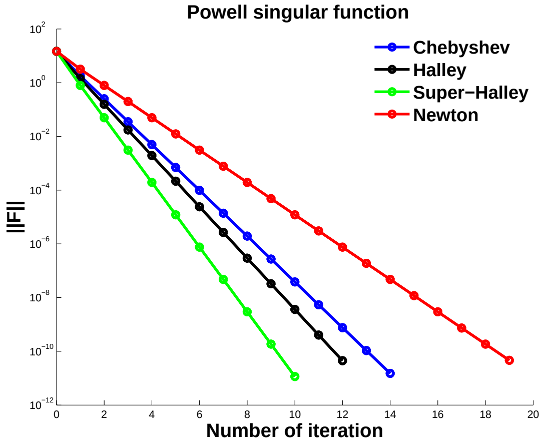


*Figure 1.* Comparing the number of iterations of Chebyshev's method, Halley's method, Super–Halley's method, and Newton's method to solve PSF.

ping criteria $||F(x)|| \leq 10^{-10}$. The Figure 1 shows that all methods are linearly convergent and Super–Halley is the best method.

## References

[1] R. P. Brent. *Algorithms for Minimization Without Derivatives*. Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1973.

[2] R. P. Brent. Some Efficient Algorithms for Solving Systems of Nonlinear Equations. *SIAM Journal on Numerical Analysis*, 10(2):327–344, 1973.

[3] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. Testing a Class of Methods for Solving Minimization Problems with Simple Bounds on the Variables. *Mathematics of Computation*, 50(182):399–430, 1988.

[4] W. Cheney D. Kincaid. *Numerical Analysis: Mathematics of Scientific Computing*. Brooks/Cole, 3rd edition, 2002.

[5] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. New Jersey, Prentice-Hall, 1983.

[6] G. Gundersen and T. Steihaug. On Large Scale Unconstrained Optimization Problems and Higher Order Methods. *Optimization Methods and Software*, 25:337–358, 2010.

[7] R. B. Kearfott. Some Tests of Generalized Bisection. *ACM Transactions on Mathematical Software*, 13(3):197–220, 1987.

[8] J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *Computer Journal*, 7:308–313, 1965.

[9] R. O'Neill. Algorithm AS 47: Function Minimization Using a Simplex Procedure. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 20(3):338–345, 1971.

[10] M. J. D. Powell. An Iterative Method for Finding Stationary Values of a Function of Several Variables. *Computer Journal*, 5:147–151, 1962.

[11] A. Qing. *Differential Evolution: Fundamentals and Applications in Electrical Engineering*. John Wiley & Sons (Asia) Pte, Singapore, 2009.

[12] S. S. Rao. *Engineering Optimization: Theory and Practice*. John Wiley & Sons, Inc., 3rd edition, 1996.

[13] H.-P. Schwefel. *Numerical Optimization of Computer Models*. John Wiley & Sons, Inc., New York, NY, USA, 1981.

[14] J.A. Snyman. *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient–Based Algorithms*. Springer Verlag, 2005.

[15] G. W. Walster, E. R. Hansen, and S. Sengupta. Test Results for a Global Optimization Algorithm. In *Proceedings of the SIAM Conference on Numerical Optimization, Boulder, Colorado June 12–14,1984*, pages 272–287. SIAM, 1985.

# Global search in nonconvex optimization problems

Alexander S. Strekalovsky

*Institute for System Dynamics and Control Theory SB of RAS, Lermontov Str., 134, Irkutsk, 664033, Russia* strekal@icc.ru

**Abstract** We consider a general optimization problem with functions, represented as a difference of two convex functions. In order to attack that problem we first concentrate on two partial cases: d.c. minimization and d.c. constraint problems — and explain local search methods and global search procedure for these two problems. Finally we give an overview of application problems investigated by the developed approach.

**Keywords:** nonconvex optimization, d.c. function, local search, global search

## 1. Introduction

Many optimization problems arising from different application areas turn out to be really nonconvex [1]-[4], in which, as known, most of local solutions are different from a global one even with respect to the value of the objective function.

Moreover, often the number of local solutions increases exponentially w.r.t. the dimension of the space, where the problem is stated.

On the other hand, the contemporary situation in Nonconvex Optimization may be viewed, as dominated by B&B and its ideological sattelites approach [2, 3]. At the same time, the classical methods of convex optimization [1, 4] have been thrown aside because of its inefficiency [2, 3, 7]. As known, the conspicuous limitation of convex optimization methods applied to nonconvex problems is their ability of being trapped at a local solution, or more often at a critical point depending on a starting point [1, 2, 4, 7].

On the other hand, applying B&B approach often we fall into so-called computational infinity, when the procedure is finite, even we are able to prove the finite convergence of the method, but it is impossible to compute a global solution in a rather reasonable time.

Taking into account the situation, we proposed another way for solving d.c. optimization problems, the principal step of which can be described on one of the basic d.c. problems: d.c. minimization (DC)

$$(\mathcal{P}): \qquad f(x) = g(x) - h(x) \downarrow \min_x, \;\; x \in D, \tag{1}$$

where $g$ and $h$ are convex functions and $D$ is a convex set, say, from $I\!R^n$.

The solution methods for d.c. problems are based on three principles.

I. Linearization w.r.t. basic nonconvexities of the problems, and, as a consequence, the consideration of the (partially) linearized problems:

$$(\mathcal{P}L(y)): \qquad g(x) - \langle h'(y), x\rangle \downarrow \min_x, \;\; x \in D,$$

where $h'(y)$ is a subgradient of the convex function $h(\cdot)$ at a point $y \in I\!R^n$.

II. Application of most advanced convex optimization methods [4] for solving the linearized problems $(\mathcal{P}L)$.

III. Application new mathematical (optimization) tools, as Global Optimality Conditions (GOC) and Global Search Strategy, based on GOC.

## 2. Local Search Methods (LCM)

In contrast to well-known B&B, cuts and similar methods throwing away classical methods of convex optimization, we insist on a certain, but nondirect, application of these algorithms in Global Optimization. For example, for solving d.c. minimization problem $(\mathcal{P})$–(1) the following (partially) linearized problem is a basic element (a "brick") of Global Search Method

$$(\mathcal{P}L_s): \qquad f(x) = g(x) - \langle h'(x^s), x\rangle \downarrow \min_x, \; x \in D, \tag{2}$$

where $x^s$ is a current iterate and $h'(x^s) \in \partial h(x^s)$ is a subgradient of $h$ at $x^s$. It means that the choice of a solving method for $(\mathcal{P}L_s)$ has a considerable impact on Global Search. Local Search procedure for $(\mathcal{P})$ may consists in consecutive solving the $(\mathcal{P}L_s)$: knowing $x^s \in D$, we find $x^{s+1} \in D$ as an approximate solution of $(\mathcal{P}L_s)$–(2). Unexpectedly, the process tends to a solution $x_*$ of the linearized problem

$$(\mathcal{P}L_*): \qquad f(x) = g(x) - \langle h'(x_*), x\rangle \downarrow \min_x, \; x \in D,$$

called henceforth a (algorithmicly) critical, $x_* \in Sol(\mathcal{P}L_*)$.

In addition to $(\mathcal{P})$ we also consider the d.c. constraint problem (DCC)

$$(\mathcal{DCC}): \qquad \left.\begin{array}{c} \phi(x) \downarrow \min, \;\; x \in S, \;\; S \subset I\!R^n, \\ F(x) = g(x) - h(x) \le 0, \end{array}\right\} \tag{3}$$

for which we proposed specials local search methods (LSM)[11] providing a critical points.

The first one consists in consecutive solution of the convex problem (if $\phi$ is convex), as follows,

$$(\mathcal{DCL}_s): \qquad \left.\begin{array}{c} \phi(x) \downarrow \min, \;\; x \in S, \\ g(x) - \langle h'(x^s), x - x^s\rangle - h(x^s) \le 0, \end{array}\right\} \tag{4}$$

and the other considers the convex problem

$$(\mathcal{QL}_s): \qquad \left.\begin{array}{c} g(x) - \langle h'(x^s), x\rangle \downarrow \min, \\ x \in S, \;\; \phi(x) \le \rho_s. \end{array}\right\} \tag{5}$$

The both methods converge to critical points.

## 3. Global Search procedures

The general procedure of Global Search consists of two stages:

a) Local Search;

b) Procedures of escaping from a critical point based upon GOC.

The meaning of this combination consists in the algorithmic (constructive) property of GOC providing a better feasible point when GOC are broken down. Actually, for $(\mathcal{P})$ GOC are, as follows, [7]

$$\left.\begin{array}{l} z \in Sol(\mathcal{P}) \Rightarrow \forall (y, \beta) \in I\!R^n \times I\!R: \\ h(y) = \beta - \xi, \;\; \xi := g(z) - h(z) \overset{\triangle}{=} f(z) \\ g(x) - \beta \ge \langle h'(y), x - y\rangle \;\; \forall x \in D. \end{array}\right\} \tag{6}$$

If for some $(\hat{y}, \hat{\beta})$ in (6) and $\hat{x} \in D$ one has $g(\hat{x}) < \beta + \langle \nabla h(\hat{y}), \hat{x} - \hat{y}\rangle$, then due to convexity of $h(\cdot)$ we immediately obtain

$$f(\hat{x}) = g(\hat{x}) - h(\hat{x}) < h(\hat{y}) + \xi - h(\hat{y}) = f(z)$$

so that $f(\hat{x}) < f(z)$, and $\hat{x}$ is better than $z$.

By varying the parameter $(y, \beta) \in I\!R^n \times I\!R$ in $(6)$, and by solving the corresponding linearized problems (cf. $(6)$)

$$g(x) - \langle h'(y), x \rangle \downarrow \min_x, \ x \in D, \tag{7}$$

($y$ is not obligatory feasible!) we get a family of starting points $x(y, \beta)$ for LSM. Besides, no needs to consider all $(y, \beta)$, it is sufficient to violate the inequality in $(6)$ for one pair $(\hat{y}, \hat{\beta})$ only.

Basing on these all results we developed a general approach for solving globally the general d.c. optimization problem, as follows,

$$\left.\begin{array}{r} F_0(x) \downarrow \min, \ \ x \in S \subset I\!R^n, \\ F_i(x) \leq 0, \ \ i = 1, \ldots, m, \end{array}\right\} \tag{8}$$

where $F_i(x) = g_i(x) - h_i(x)$ and $g_i, \ h_i(x)$ are convex functions on $I\!R^n, \ \ i = 1, \ldots, m.$

The large field of computational experiments confirmed the effectiveness of the approach for high dimensional problems even in the case of program implementation performed by students and post-graduation students [9]–[14].

## 4. Applications

4.1. **Bimatrix games (BMG).** A new method for finding the Nash equilibrium in BMG has been developed [8]. This one is based on reducing BMG to a bilinear maximization problem and a following application of Global Search Strategy. The testing on widely generated BMG of high dimension (up to $1000 \times 1000$) showed the comparable effectiveness of the method.

4.2. Bilevel problems can be viewed as extremal problems having a special constraint in the form of another optimization problem (follower problem). Besides, the constraints of the leader are depending on the variables of the follower. A special complex of programmes for solving these problems was developed and successfully tested on a large number stochastically generated examples of different complexity and dimension (up to $150 \times 150$) [15].

4.3. Linear complementarity problem was solved by variational approach stating it as d.c. minimization problem of dimension up to 400 [14].

4.4. Problems of financial and medical diagnostic can be formulated in the form of nonlinear (polyhedral) separability [6]. The generalization of Global Search Theory for nonsmooth case allows to develop a programming complex for solving such problems of rather high dimension with demonstrated its effectiveness during computational simulations.

4.5. Well-known problems of **Discrete programming** [9] and nonconvex **Optimal control** problems [12, 13] have also been considered.

## 5. Conclusion

We developed a new approach for solving general d.c. optimization problems, in particular, d.c. minimization and d.c. constraint's problem. This approach is based on Global Optimality Conditions and consists of two stages:

1) Special Local Search Method (LSM);

2) Procedures of escaping from critical points provided by LSM.

An attractive peculiarity of the approach consists in non-direct application of Classical Convex Optimization Methods (Interior Point Methods, Newtonean and Trust-Regions Procedures, SQP etc.) into LSM and Global Search Procedures. Numerical results of high dimensions witnessed (certified) rather competitive effectiveness of the approach.

# References

[1] Vasil'ev F.P. (2002) *Optimization Methods.* Factorial, Moscow.

[2] Horst R., Tuy H. (1993) *Global Optimization. Deterministic Approaches.* Berlin, Springer-Verlag.

[3] *Frontiers in Global Optimization.* (2004) ed. by C.A. Floudas, P.M. Pardalos. New York, Kluwer Academic Publishers.

[4] Nocedal J., Wright S.J. (2006) *Numerical Optimization.* Berlin, New York.

[5] Hiriart-Urruty J.B., Lemarshal C. (1993) *Convex Analysis and Minimization Algorithms.* Berlin, Springer Verlag.

[6] Mangasarian, O.L., Street, W.N., Wolberg, W.H. (1995) *Breast Cancer Diagnosis and Prognosis via linear Programming.* Operations Research, N 43, pages 570–577.

[7] Strekalovsky A.S. (2003) *Elements of Nonconvex Optimization.* Nauka, Novosibirsk.

[8] Strekalovsky A.S., Orlov A.V. (2007) *Bimatrix Games and Bilinear Programming.* Moscow, Fizmatlit.

[9] Strekalovsky A.S., Kuznetsova A.A. (2001) *On solving the Maximum Clique Problem.* Journal of Global Optimization, Vol. 21, pages 265–288.

[10] Strekalovsky A.S., Orlov A.V. (2007) *A new approach to nonconvex optimization.* Numerical Methods and Programming. Vol.8, pages 11–27.

[11] Strekalovsky A.S., Gruzdeva T.V. (2007) *Local Search in Problems with Nonconvex Constraints.* Computational Mathematics and Mathematical Phisics. Vol. 47, pages 381–396.

[12] Strekalovsky A.S., Yanulevich M.V. (2008) *Global Search in the Optimal Control Problem with a Terminal Objective Functional Represented as the Difference of Two Convex Functions.* Computational Mathematics and Mathematical Physics. Vol. 48, pages 1119–1132.

[13] Strekalovsky A.S. (2007) *Optimal Control Problems with Terminal Functionals Represented as the Difference of Two Convex Functions.* Computational Mathematics and Mathematical Physics. Vol. 47, pages 1788–1801.

[14] Mazurkevich E.O., Petrova E.G., Strekalovsky A.S. (2009) *On Numerical Solving Linear Complimentarity Problem.* Computational Mathematics and Mathematical Phisics. Vol. 49, pages 1385–1398.

[15] Strekalovsky A.S., Orlov A.V., Malyshev A.V. (2010) *On computational search for optimistic solution in bilevel problems.* Journal of Global Optimization (available online, DOI: 10.1007/s10898-009-9514-z).

[16] Strekalovsky A.S., Orlov A.V., Malyshev A.V. (2010) *A local search for the quadratic-linear bilevel programming problem.* Siberian Journal of Numerical Mathematics. Vol 13, pages 75–88.

[17] Strekalovsky A.S., Orlov A.V., Malyshev A.V. (2010) *Numerical solution of a class of bilevel programming problems.* Siberian Journal of Numerical Mathematics. Vol 13, pages 201–212.

# Routing of hazardous materials: models and optimization

Nora Touati Moungla and Vincent Jost

*École polytechnique, Laboratoire d'informatique (LIX), 91128 Palaiseau Cedex, France,* {touati,vjost}@lix.polytechnique.fr

**Abstract** The routing of vehicles represents an important component of many distribution and transportation systems and has been intensively studied in the operations research literature. In this paper, particular consideration is given to routing models for the transportation of hazardous materials. This problem has received a large interest in recent years, this results from the increase in public awareness of the dangers of hazardous materials and the enormous amount of hazardous materials being transported. We describe here some major differences between routing of hazardous materials and the classical vehicle routing problems. We review some general models and optimization techniques and propose several direction for future research.

**Keywords:** routing, transportation of hazardous materials, multi-objective optimization

## 1. Introduction

Hazardous Materials Management (HMM) has received a large interest in recent years, this results from the increase in public awareness of the dangers of hazardous materials and the enormous amounts of hazardous materials being transported. HMM is concerned with four main subjects: risk analysis [3], routing and scheduling [8], facility location [6, 8] and treatment and disposal of waste [10]. We focus here on Routing of Hazardous Materials (RHM). The main target of this problem is to select routes from a given origin 's' to a given destination 't' such that the risk for the surrounding population and the environment is minimum, without producing excessive economic costs. The risk associated with the hazardous materials makes these problems more complicated by its assessment, the collection of the related data and the resolution of the associated formulations.

In analyzing the routing of hazardous materials problem, it is important to include multiple objectives, this results from the presence of multiple interested parties or stakeholders. In this case, it is not possible to identify a single "best" route, generally "Pareto optimal" routes represent the available tradeoffs explicitly. Another important aspect of the transportation of hazardous materials is uncertainty, this is a result of the difficulty of risk measurement and the lack of data. We present in this paper some characteristics of routing of hazardous materials problem and describe some models and the most used resolution methods for this problem.

## 2. Routing for hazardous materials

The difference between RHM and other routing problems is mainly the risk. We present in this section some methodologies in RHM problem consisting essentially of modeling and resolution frameworks.

## 3. The risk

Because of the risk, a network related to this problem is generally different from other transportation networks. In RHM, the arcs do not necessarily connect junctions, in the case where a road between two junctions goes through a set of different density population regions, this arc is divided into a set of arcs in such a way that these new arcs have the same incident probability and the same consequences.

Although the fact that the major target of the RHM problem is the minimization of the risk, there is no universally accepted definition of risk [5]. It is pointed in [8] that the evaluation of risk in transportation of hazardous materials generally consists of the evaluation of the probability of an undesirable event, the exposure level of the population and the environment, and the degree of the consequences (e.g., deaths, injured people, damages). In practice, these probabilities are difficult to obtain due to the lack of data and generally, the analysis is reduced to consider the risk as the expected damage or the population exposure.

As the risk is a part of the objective function, it is quantified with a path evaluation function [5]. Some risk models (the conditional risk model for example, see [12]) lead to non-linear binary integer formulations which can not optimized using a simple labeling algorithm (the associated models violate a path selection optimality principle). Generally, these models are based on approximations which lead to tractable formulations. Three axioms are introduced in [5] for prescribing properties for path evaluation functions. These axioms allow to check the monotonicity of the links attribute and to check if the path selection model verifies Bellman's optimality principle.

## 4. Models and optimization

RHM problem is multi-objective in nature, nevertheless, some papers deal with single-objective problems. These models often fail to handle the conflict between transportation risk and cost. A number of multi-objective models have been proposed in the literature. With multiple objectives, all objectives usually cannot be optimized simultaneously. Generally, a set of alternative (Pareto-optimal) solutions are computed. As the number of Pareto-optimal solutions can be exponential as a function of the network size, one might wish to compute a subset of these solutions "approximating well" the set of all Pareto-optimal solutions.

Resolution methods of hazardous materials routing can be classified in two categories. The "local routing" which consists in selecting routes between only one origin-destination pair and transporting a single commodity at a time. The "global routing" where different hazardous materials have to be shipped simultaneously among different origin-destination pairs.

### 4.1 Local routing of hazardous materials

The one origin-destination routing consists of selecting a route between a given origin-destination nodes for a given hazardous materials, transport mode and vehicle type.

Weighting methods are widely used due to their simplicity and computational efficiency. They are based on optimizing a weighted linear combination of the objectives. This can be done using the classical shortest path algorithm. The drawback of these methods is that they are able to identify only a subset of Pareto-optimal solutions. Since some solutions of interest might be ignored, a method which can identify them is desirable. However, in some cases when the decision maker is able to express additional a priori knowledge and preferences on the objectives, the problem can be reduced to a single objective optimization problem.

The goal programming formulations offers considerable flexibility to the decision maker, the purpose is to minimize the deviation between the achievement of objectives (goals) and their aspiration level (the acceptable achievement level for the objective). This method is able to compute Pareto-optimal solutions that can not be obtained with the weighting method.

### 4.2 Global routing of hazardous materials

A non-negligible work in the literature focuses on the selection of a single commodity route between only one origin-destination pair. In practice, a more adapted model is global routing. When many vehicles have to be routed between the same origin-destination nodes, these vehicles are routed on the same path, so, the risk associated to regions surrounding this path could be hight. In this case, we sometimes wish to distribute the risk in an equitable way over the population.

**Risk equity.** Different techniques was proposed to handle equity on the transportation network. In [7], the authors guarantee equity by constraining the differences of the risks associated to every pair regions, to be less than or equal to a given threshold. The computation of routes with a fairly distributed risk consists in generating dissimilar origin-destination paths. The dissimilar path generation problem has been dealt in the literature in many ways, we cite the Iterative Penalty Method, the Gateway Shortest-Paths method, the Minimax Method and the p-dispersion Method.

**Multi-commodity flow models.** The transportation of hazardous materials can be naturally modeled by a multi-commodity flow model [2]. Given an origin-destination pair and the amount of commodities to be transported between such an origin and a destination, the multi-commodity minimum cost flow model finds the optimal distribution of such quantities minimizing the total transportation cost.

## 5. Uncertainty in routing of hazardous material

It is important to classify the nature of uncertainty in transportation of hazardous materials problems. In particular, three types of uncertainty concern the amount of population present near a route, traffic congestion and weather conditions. The effect of the release of hazardous materials and the travel time can be modeled by means of random variables whose distributions may vary over time [14] over a stochastic and time-dependent network [1]. Optimal routing on time-dependent networks can be classified into three categories [4]:

- *A priori optimization*: Optimal routes are definitively computed before the travel begins, the random arc travel time is reduced to its expected value and a standard shortest path problem is applied.
- *Adaptive route selection*: The optimal route depends on intermediate informations concerning past travel time, road congestion and weather conditions. The adaptive road specifies the road link to be chosen at each intermediate node, as a function of the arrival time at the node [11]. The multi-objective version of the adaptive route selection was proposed in [9].
- *Adaptive route selection with real-time updates*: In this case, recent technologies such as automatic vehicle location and mobile communication systems permits to guide the route of vehicles based on real-time informations. Estimation of future values of some network attributes such as travel times, incident probabilities and population in the impact area are updated using the real-time informations.

## 6. Conclusion

Transportation of hazardous materials is a complex and seemingly intractable problem, principally because of the inherent trade-offs between social, environmental, ecological, and economic factors. An model for the routing of hazardous materials problem not only needs to be accurate but also technically and computationally tractable. There is no common conceptual model for the RHM problem. Works in this field take generally into account different considerations (economic, environmental and social) and significant simplifications are necessary to obtain tractable models. Several challenging directions for future research can be stemmed from this problem:

- A common resolution approach for multi-objective shortest path problems consists of computing Pareto optimal solutions. As the number of Pareto optimal solutions can grow exponentially with the size of the network, one can propose to the decision maker a subset of Pareto optimal solutions representing a good approximation of the Pareto optimal solutions set or compute preferred solutions by exploiting some preferences of the decision maker.
- An important issue of the RHM is the treatment of the stochastic phenomena, indeed, some or all attributes of this problem are stochastic.
- Considerable advances are needed to appropriately treat the stochastic phenomena when some transformation of measures (cost and risk) are nonlinear.
- New technological advances in communication systems and Global Positioning System (GPS) are challenging researchers to develop routing models and robust optimization procedures that are able to respond quickly to changes in the data.
- Most studies on RHM in the literature deal with road transportation mode [4]. Although rail transport is a safer transportation mode (automatic control system, cross less populated zones), and more capacitated, it has received less attention.

## References

[1] T. Chang and L. Nozick and M. Turnquist. Multi-objective path finding in stochastic dynamic networks, with application to routing hazardous materials shipments. *Transportation Science*, 39(3):383–399, 2005.

[2] M. Caramia and P. Dell'Olmo. Multi-objective Management in Freight Logistics: Increasing Capacity, Service Level and Safety With Optimization Algorithms. *Springer London Ltd*, 2008.

[3] E. Erkut and A. Ingolfsson. Transport risk models for hazardous materials: revisited. *Operations Research Letters*, 33:81–89, 2005.

[4] E. Erkut. and S. Tjandra and V. Verter. Hazardous Materials Transportation. *Chapter 9 in Transportation, C. Barnhart and G. Laporte (editors), Handbooks in Operations Research and Management Science*, Elsevier 14:539–621, 2007.

[5] E. Erkut and V. Verter. Modeling of transport risk for hazardous materials. *Operations Research*, 46(5):625–642, 1998.

[6] E. Erkut and V. Verter. Facility Location: A Survey of Applications and Methods. *Hazardous materials logistics, in: Z. Drezner (Ed.), Springer Series in Operations Research*, Springer, Berlin, (Chapter 20), 1995.

[7] R. Gopalan and R. Batta and M.H. Karwan. The equity constrained shortest path problem. *Computers and Operations Research*, 17:297–307, 1990.

[8] G.F. List and P.B. Mirchandani and M.A. Turnquist and K.G. Zografos. Modeling and analysis for hazardous materials transportation: Risk analysis, routing/scheduling and facility location. *Transportation Science* 2:100–114, 1991.

[9] E.D. Miller-Hooks. Adaptive least-expected time paths in stochastic, time-varying transportation and data networks. *Networks*, 37(1):35–52, 2001.

[10] A.K. Nemaa and S.K. Gupta. Optimization of regional hazardous waste management systems: an improved formulation. *Waste Managemen*t 19:441–451, 1999.

[11] S. Nguyen and S. Pallottino. Hyperpaths and shortest hyperpaths. *In: Combinatorial Optimization. Lecture Notes in Mathematics*, 1403:258–271, Springer-Verlag, Berlin, 1986.

[12] R.A. Sivakumar and R. Batta and M.H. Karwan. A multiple route conditional risk model for transporting hazardous materials. *INFOR* 33:20–33, 1995.

[13] P. Toth and D. Vigo. The vehicle routing problem. *Philadelphia: Society for Industrial and Applied Mathematics*, 2002.

[14] M. Turnquist. Multiple Objectives, Uncertainty and Routing Decisions for Hazardous Materials Shipments. *Proceedings of the 5th International Conference on Computing in Civil and Building Engineering ASCE*, New York 357–364, 1993.

# On Local search for nonconvex optimal control problem

Maxim V. Yanulevich

*Institute for System Dynamics and Control Theory of SB of RAS, Lermontov str. 134, Irkutsk, 664033, Russia,* max@irk.ru

**Abstract** A nonconvex optimal control problem in which a nonconvexity is generated by an integro-terminal objective functional is considered. A new local search method, which allows to obtain a control process verifying, in particular, Pontryagin principle, is proposed. Some peculiar properties of convergence of the algorithm are investigated. Furthermore, some preliminary computational simulations have been conducted, the results of which certify a rather competitive efficiency of the algorithm.

**Keywords:** nonconvex optimal control problems, Pontryagin's maximum principle, locally and globally optimal processes

## 1. Introduction

The present paper investigates only one type of nonconvexities generated by the Bolza objective functional. On the other hand, the objective is also rather simple: to construct a special local search method allowing to obtain a stationary (in the sense of Pontryagin maximum principle (PMP) [1–6]) point.

Note, this goal — to create a special local search method for each kind of nonconvex problems — have not been implemented and fixed until present for the OC problem, while — as far as finite dimensional problems are concerned, — some results have been obtained for different kinds of nonconvex problems (d.c. minimization, convex maximization, d.c. constraint problems etc.) [7–10]. In our opinion, only after creating special Local Search Algorithms (LSA) for different types of nonconvex OC problems, one can begin with constructing Global Search Procedures (based, say, on Global Optimality Conditions) allowing to escape from a stationary (PMP) process with improving the value of objective functional on various nonconvex OC problems [9].

## 2. Problem formulation

Consider the following control system

$$\left.\begin{array}{c} \dot{x}(t) = A(t)x(t) + b(u(t),t) \quad \overset{\circ}{\forall} t \in T = [t_0, t_1], \\ x(t_0) = x_0 \in I\!R^n, \end{array}\right\} \tag{1}$$

where the matrix $A(t)$ has components $t \mapsto a_{ij}(t)$, $i, j = 1, 2, \ldots, n$, from $L_\infty(T)$ and the mapping $(u,t) \mapsto b(u,t)\colon I\!R^{r+1} \to I\!R^n$ is continuous w.r.t. every variable $u \in I\!R^r$ and $t \in I\!R$. Further, the control $u(\cdot)$ verifies the standard assumptions of OC, as follows,

$$u(\cdot) \in \mathcal{U} = \{\, u(\cdot) \in L^r_\infty(T) \mid u(t) \in U \quad \overset{\circ}{\forall} t \in T \,\} \tag{2}$$

where the set $U$ is compact in $I\!R^r$. Here the sign $\overset{\circ}{\forall}$ means "for almost all".

Under these assumptions for each $u(\cdot) \in \mathcal{U}$ there exists an unique solution $x(t) = x(t,u)$, $t \in T$, of the initial Cauchy problem (1), $x(\cdot) \in AC^n(T)$ (i.e. absolutely continuous) [2, 3].

The goal of this control consists in minimization of the following Bolza functional

$$(\mathcal{P}):\ J(u) := \varphi(x(t_1)) + \int\limits_T \big[F(x(t),t) + f(u(t),t)\big]\,dt \downarrow \min_u, \quad u \in \mathcal{U}, \tag{3}$$

subject to (1)–(2), i.e. $x(t) = x(t,u)$, $t \in T$, $u \in \mathcal{U}$ in (3). Here the function $x \mapsto \varphi(x)\colon I\!R^n \to I\!R$ is differentiable on a rather large open convex set $\Omega \subset I\!R^n$ and the function $(u,t) \mapsto f(u,t) : I\!R^{r+1} \to I\!R$ is a continuous w.r.t. each variable. Besides, let's function $\varphi(\cdot)$ be d.c. function (see [3, 7]), i.e. $\varphi(\cdot)$ is represented as the difference of two convex functions:

$$\varphi(x) = g_1(x) - h_1(x) \quad \forall x \in \Omega, \tag{4}$$

where $g_1(\cdot)$ and $h_1(\cdot)$ are convex functions on $\Omega$.

Function $(x,t) \mapsto F(x,t)$ is also continuous w.r.t. each variable and besides is differentiable d.c. function w.r.t. the first variable $x$ on $\Omega \subset I\!R^n$ such as the following d.c. decomposition holds:

$$F(x,t) = g(x,t) - h(x,t) \quad \forall x \in \Omega, \quad t \in [t_0,t_1], \tag{5}$$

where $x \mapsto g(x,t)$ and $x \mapsto g(x,t)$ is convex functions w.r.t. variable $x$.

Under such assumptions Problem $(\mathcal{P})$–(1)–(3) turns out to be nonconvex so that it may possess a number of locally optimal and stationary (verifying PMP) processes $(x_*(\cdot), u_*(\cdot))$, which are different from a global one $(z(\cdot), w(\cdot))$, $z(t) = x(t,w)$, $t \in T$, $w(\cdot) \in \mathcal{U}$, even w.r.t. the values of the objective functional. The examples of such problems can be found in [8, 9].

The following sections of the paper present some regular procedures allowing to obtain a stationary process for the Problem $(\mathcal{P})$–(1)–(3).

## 3. Linearized problem

Under the assumptions of the previous section consider the maximization problem as follows

$$\begin{aligned}(\mathcal{PL}(y))\colon \qquad & I_y(u) := g_1(x(t_1,u))) - \langle \nabla h_1(y(t_1)), x(t_1,u)\rangle + \\ + \int\limits_T & \big[g(x(t,u),t) - \langle \nabla h(y(t),t), x(t,u)\rangle + f(u(t),t)\big]\,dt \downarrow \min_u, \quad u \in \mathcal{U}\end{aligned} \tag{6}$$

subject to (1)–(2), where $y(t) \in I\!R^n$, $t \in [t_0,t_1]$, is a given continuous function. It is well-known, in convex Problem $(\mathcal{PL}(y))$–(6) Pontrygin's Maximum Principle turns out to be the necessary and sufficient condition for the process $(x_*(t), u_*(t))$ being (globally) optimal w.r.t. $(\mathcal{PL}(y))$. More precisely, if $(x_*(t), u_*(t))$ is solution to $(\mathcal{PL}(y))$–(6), then the maximum condition

$$H_L(x_*(t), u_*(t), \psi(t), t) = \max_{v \in U} H_L(x_*(t), v, \psi(t), t) \quad \overset{\circ}{\forall} t \in T, \tag{7}$$

holds with the Pontryagin function for the problem $(\mathcal{PL}(y))$–(6)

$$H_L(x,u,\psi,t) = \langle \psi, A(t)x + b(u,t)\rangle - g(x,t) + \langle \nabla h(y(t),t), x\rangle - f(u,t). \tag{8}$$

Here the function $\psi(t) = \psi_y(t)$, $t \in [t_0,t_1]$ is a unique absolutely continuous solution of the adjoint system

$$\begin{gathered}\dot\psi(t) = -A(t)^\top \psi(t) + \nabla g(x_*(t),t) - \nabla h(y(t),t), \quad t \in T, \\ \psi(t_1) = \nabla h_1(y(t_1)) - g_1(x_*(t_1)).\end{gathered} \tag{9}$$

On account of the special form (8) of the Pontryagin function, the maximum condition (7) may be written in the following form:

$$\langle \psi(t), b(u_*(t),t)\rangle + f(u_*(t),t) = \max_{v \in U}[\langle \psi(t), b(v,t)\rangle + f(v,t)] \quad \overset{\circ}{\forall} t \in T. \tag{7'}$$

Note that for the problem $(\mathcal{PL}(y))$ the adjoint state $\psi_y(t) = \psi(t, y(\cdot), x_*(\cdot))$ corresponding to Pontryagin's extremal $(x_*(\cdot), u_*(\cdot))$ is given implicitly. In order to verify that the process $(x_*(\cdot), u_*(\cdot))$ satisfy PMP, it's necessary find adjoint trajectory $\psi(\cdot) = \psi(\cdot, y, x_*(\cdot))$ depending on phase state $x_*(\cdot)$ (see [3–6]).

## 4. Local Search Method

To the end of solving Problem $(\mathcal{P})$–(1)–(3) the following way has shown itself rather efficient for finite-dimensional problems. In the case of Optimal Control, this procedure can be described as follows.

Once a feasible control $u^s(\cdot) \in \mathcal{U}$ is given, the next iteration $u^{s+1}(\cdot) \in \mathcal{U}$ is chosen as an (approximate) solution of the linearized problem

$$\begin{aligned}(\mathcal{PL}_s)\colon \qquad & I_s(u) := g_1(x(t_1, u)) - \langle \nabla h_1(x^s(t_1)), x(t_1, u)\rangle + \\ + \int\limits_T \big[g(x(t,u),t) - \langle \nabla h(x^s(t),t), x(t,u)\rangle &+ f(u(t),t)\big]\, dt \downarrow \min_u, \quad u \in \mathcal{U},\end{aligned} \tag{10}$$

where $x(t,u)$ and $x^s(t) = x(t, u^s)$, $t \in [t_0, t_1]$, are solutions of the system (1) of ODEs with $u(\cdot)$ and $u^s(\cdot)$, respectively.

The issue of convergence of the sequence $\{x^s(\cdot), u^s(\cdot)\}$ generated by the above procedure emerges immediately.

On the other hand, it is clear from the previous section that solution $(x^{s+1}(\cdot), u^{s+1}(\cdot))$ of Problem $(\mathcal{PL}_s)$–(10) satisfy the following maximum condition

$$\langle \psi^s(t), b(u^{s+1}(t),t)\rangle + f(u^{s+1}(t),t) = \max_{v \in U} \big[\langle \psi^s(t), b(v,t)\rangle + f(v,t)\big] \quad \overset{\circ}{\forall} t \in T, \tag{11}$$

where adjoint trajectory satisfy the system

$$\begin{gathered}\dot{\psi}^s(t) = -A(t)^\top \psi^s(t) + \nabla g(x^{s+1}(t),t) - \nabla h(x^s(t),t), \quad t \in T, \\ \psi^s(t_1) = \nabla h_1(x^s(t_1)) - g_1(x^{s+1}(t_1)),\end{gathered} \tag{12}$$

and phase state $x^{s+1}(t)$ provides for the control $u^{s+1}(\cdot) \in \mathcal{U}$.

After that, the state $x^{s+1}(t)$ is computed as the solution of the control system (1) corresponding to the control $u^{s+1}(\cdot) \in \mathcal{U}$.

This idea leads us to a more realistic algorithm, the principal steps of which have been discussed above.

Let there be given a sequence of numbers $\{\delta_s\}$ such that

$$\delta_s > 0, \quad s = 0, 1, 2..., \quad \sum_{s=0}^{\infty} \delta_s < +\infty, \tag{13}$$

and a current process $(x^s(\cdot), u^s(\cdot))$, $u^s(\cdot) \in \mathcal{U}$, $x^s(t) = x(t, u^s)$, $t \in [t_0, t_1]$.

Having the state $x^s(\cdot) \in AC^n(T)$ one can solve the corresponding adjoint system (12). After that we construct the control $u^{s+1}(\cdot) \in \mathcal{U}$ by approximate solving almost everywhere over $T$ the finite dimensional problem provided by rule (11), so that the following inequality holds:

$$\begin{gathered}\langle \psi^s(t), b(u^{s+1}(t),t)\rangle + f(u^{s+1}(t),t) + \frac{\delta_s}{t_1 - t_0} \geq \\ \geq \sup_{v \in U} \big[\langle \psi^s(t), b(v,t)\rangle + f(v,t)\big] \quad \overset{\circ}{\forall} t \in T.\end{gathered} \tag{14}$$

The following theorem proves the convergence of sequence $\{x^s(\cdot), u^s(\cdot)\}$ generated by the above procedure.

**Theorem 1.** *The sequence of controlled processes* $\{x^s(\cdot), u^s(\cdot)\}$ *generated by rules* (12), (13) *and* (14) *fulfils PMP in the sense of condition*

$$\lim_{s\to\infty} \sup_{v\in U}[\langle\psi^s(t), b(v,t) - b(u^s(t),t)\rangle + f(v,t) - f(u^s(t),t)] = 0 \qquad \overset{\circ}{\forall} t \in T. \tag{15}$$

*Furthermore, numerical sequences* $J(u^s)$ *and* $I_s(u^s)$ *converge.*

## 5. Summary

In the present paper, a nonconvex OC problem has been considered, in which the nonconvexity has been generated by maximizing the objective functional with d.c. terminal part, and with the integral part having a d.c. integrand (w.r.t. the state).

On the whole, the problem turns out to be nonconvex in the sense that there may be local solutions, which are rather far from a globally optimal process even w.r.t. the values of the goal functional.

Further, for this problem we have proposed and substantiated the local search method based, on one hand, on the classical linearization idea and, on the other, on the method of solving linearized problems.

In addition, the convergence of the developed algorithm has been investigated. Finally, the first computational testing of the developed algorithm has shown itself rather efficient and demonstrated the possibility of applying the algorithm to the global search procedure, which we intend to describe in our forthcoming papers.

## Acknowledgments

The author wish to thank scientific adviser Prof. Alexander Strekalovsky for your encouragement and support.

## References

[1] Pontryagin L.S., Boltyanskii V.G, Gamkrelidze R.V., Mishchenko E.F. (1976) The Mathematical Theory of Optimal Processes. New York: Interscience.

[2] Lee E.B., Markus L. (1967) Foundations of Optimal Control Theory. New York: Wiley.

[3] Vasil'ev F.P. (2002) Optimization Methods. Moscow: Factorial Press (in Russian).

[4] Gabasov R., Kirillova F.M. (1979) Optimization of linear systems. New York: Plenum Press.

[5] Vasiliev O.V. (1996) Optimization methods. Atlanta: Word Federation Publishing Company.

[6] Srochko V.A. (2000) Iterative Solution of Optimal Control Problems. Moscow: Fizmatlit (in Russian).

[7] Strekalovsky A.S. (2003) Elements of Nonconvex Optimization. Novosibirsk: Nauka (in Russian).

[8] Strekalovsky A.S., Yanulevich M.V. (2008) Global search in the optimal control problem with a therminal objective functional represented as a difference of two convex functions. Computational Mathematics and Mathematical Physiscs, 48(7): 1119–1132.

[9] Strekalovsky A.S. (2007) Optimal control problems with terminal functionals represented as a difference of two convex functions. Computational Mathematics and Mathematical Physiscs, 47(11): 1788–1801.

[10] Strekalovsky A.S., Sharankhaeva E.V. (2005) Global Search in a Nonconvex Optimal Control Problem. Computational Mathematics and Mathematical Physiscs. 45(10): 1719–1734.

# Author Index

# Topic Index