Proceedings of the XII global optimization workshop

MATHEMATICAL AND APPLIED GLOBAL OPTIMIZATION

MAGO 2014

Málaga, September 2014

PROCEEDINGS OF THE XII GLOBAL OPTIMIZATION WORKSHOP

MATHEMATICAL AND APPLIED GLOBAL OPTIMIZATION

MAGO 2014

Edited by

L.G. Casado
Universidad de Almería

I. García
Universidad de Málaga

E.M.T. Hendrix
Universidad de Málaga

ISBN: 978-84-16027-57-6
DEPOSITO. LEGAL: AL 695-2014

Preface

Global Optimization Workshops are organized as light overhead meetings rather spontaneously by members of the Global Optimization scientific community. Without presidents and committees in a flat organization, its aim is to stimulate discussion between senior and junior researchers on the topic of Global Optimization in a one stream setting. The tradition continues since its first meetings in Sopron (1985 and 1990) followed by Szeged (1995), Florence (GO'99, 1999), Hanmer Springs (NZ, Let's GO, 2001), Santorini (2003), San José (GO05, 2005), Mykonos (2007), Skukuza (SAGO, 2008), Toulouse (TOGO, 2010) and Natal (Br, NAGO, 2012) and now taking place in Málaga (MAGO, 2014).

The lead was taken this time by a group of researchers of the High Performance Computing -Algorithms group in southern Spain. More than 40 interested researchers sent in an extended abstract which can be used as a discussion document describing a problem and/or an algorithm to be deliberated during the meeting. In addition, Panos Pardalos was prepared to do the kick-off for the workshop with an overview of research questions that have been dealt with and topics that are still open for further research.

This proceedings book provides an overview of the questions discussed during the workshop. The idea is that researchers may continue their investigation inspired by the discussion and successful papers can be submitted to a special issue of the Journal of Global optimization dedicated to the workshop.

Eligius M.T. Hendrix (Málaga)
Inmaculada García (Málaga)
Leocadio G. Casado (Almería)

MAGO 2014 Co-chairs

Local organisers:

- Alejandro G. Alcoba
- Carmen Donoso Mateo
- Eligius M.T. Hendrix
- Guillermo Aparicio
- Inmaculada García
- Juan F.R. Herrera
- Leocadio G. Casado
- Sonia González Navarro
- Siham Tabik

Sponsors:

- Universidad de Málaga.
- Escuela Técnica Superior de Ingeniería Industrial.

Contents

Extended Abstracts

Introductory talk to the workshop MAGO 2014: Progress and Challenging Problems in Global Optimization by Panos M. Pardalos

Abstract

We are honoured to announce the talk of Panos M. Pardalos. Panos is a recognised scholar with more than 350 published journal articles, 15 books and editor of numerous journals. After his thesis and first book with J. Ben Rosen, he was one of the founding fathers of the global optimization community and its journal in 1990; the Journal of Global Optimization founded with Reiner Horst. The community is nearly as dynamic as Panos. By editing numerous journals and initiating book series, he stimulated from the beginning the appearance of papers and books on the topic of global optimization.

Outline of the talk

An overview of the state of global optimization was given by Panos in an invited plenary talk [6] at the 15th International Symposium on Mathematical Programming (University of Michigan, Aug. 15-19, 1994). Starting of from the quadratic viewpoint his talk handled relations with integer programming, semidefinite programming, fractional programming, etc. By that time an interesting aspect was the lack of available software and codes for solving global optimization problems.

In addition, it was acknowledged that global optimization had begun expanding in all directions at an astonishing rate, and that new algorithmic and theoretical techniques had begun development. The diffusion into other disciplines had proceeded at a rapid pace, and our knowledge of all aspects of the field had grown even more profound. At the same time one of the most striking trends in global optimization was the constantly increasing interdisciplinary nature of the field. This makes our discusions very dynamic; we have to adapt to new potential application fields.

In the initial stage of the discussion on global optimization [2], we tried to capture all knowledge so far in teaching books [4, 8] and handbooks [3, 5, 7]. Space was given to new fields interested in the global optimization aspects by starting book series and promoting a wide variety of application fields in the Journal of Global Optimization.

We asked Panos in this talk to go back and reflect on the progress of the field, especially some of the major developments and research directions and open questions in global optimization. After two decades we have more efficient computational approaches accompanied by the availability of several global optimization solvers. However, many outstanding open questions remain and new ones arise in relation to specific applications. At the present there is a huge interest in data driven applications and optimization with massive data sets [1]. New, challenging problems arise in connection to novel algorithmic approaches (e.g. external memory algorithms) and new computing environments (e.g. cloud computing, quantum computers etc.).

References

[1] J. Abello, P.M. Pardalos, and M. G. C. Resende, editors. *Handbook of Massive Data Sets*. Kluwer Academic Publishers, Dordrecht, Holland, 2002.

[2] L. C. W. Dixon and G. P. Szegö. *Towards Global Optimisation*. North Holland, Amsterdam, 1975.

[3] R. Horst and P. M. Pardalos. *Handbook of Global Optimization*. Kluwer, Dordrecht, 1995.

[4] R. Horst, P.M. Pardalos, and N.V. Thoai, editors. *Introduction to Global Optimization*, volume 3 of *Noncovex Optimization and its Applications*. Kluwer Academic Publishers, Dordrecht, Holland, 1995.

[5] R. Horst and H. Tuy. *Global Optimization (Deterministic Approaches)*. Springer, Berlin, 1990.

[6] P. M. Pardalos. On the passage from local to global in optimization. In J. R. Birge and K. G. Murty, editors, *Mathematical Programming: State of the Art 1994*, pages 220–247. University of Michigan, Ann Arbor, 1994.

[7] P. M. Pardalos and E. H. Romeijn. *Handbook of Global Optimization Vol 2*. Kluwer, Dordrecht, 2002.

[8] P.M. Pardalos and J. B. Rosen. *Constrained Global Optimization; Algorithms and Applications*, volume 268 of *Lecture Notes in Computer Science*. Springer, Berlin, 1987.

EXTENDED ABSTRACTS

On computing order quantities for perishable inventory control with non-stationary demand[*]

A.G. Alcoba[1], E.M.T. Hendrix[1], I. García[1], K.G.J. Pauls-Worm[2], and R. Haijema[2]

[1]*Computer Architecture, Universidad de Málaga,* {agutierreza,eligius,igarciaf}@uma.es

[2]*Operations Research and Logistics, Wageningen University,* {karin.pauls,rene.haijema}@wur.nl

Abstract We study the global optimal solution for a planning problem of inventory control of perishable products and non-stationary demand.

Keywords: Inventory control, Perishable products

1. Introduction

The basis of our study is a SP model published in [3] for a practical production planning problem over a finite horizon of T periods of a perishable product with a fixed shelf life of J periods. The demand is uncertain and non-stationary such that one produces to stock. To keep waste due to out-dating low, one issues the oldest product first, i.e. FIFO issuance. Literature provides many ways to deal with perishable products, order policies and backlogging, e.g. [5, 1]. The model we investigate aims to guarantee an upper bound for the expected demand that cannot be fulfilled for every period.

The solution for such a model is a so-called order policy. Given the inventory situation $\mathbf{I}$ at the beginning of period moment t, an order policy should advice the decision maker on the order quantity Q_t. For the decision maker, simple rules are preferred. We consider a policy with a list of order periods Y with order quantities Q_t.

2. Stochastic Programming Model

The stochastic demand implies that the model has random inventory variables $\boldsymbol{I}_{jt}$ apart from the initial fixed levels I_{j0}. In the notation, $P(.)$ denotes a probability to express the chance constraints and $E(.)$ is the expected value operator for the expected costs. Moreover, we use $x^+ = max\{x, 0\}$. A formal description of the SP model from [2] is given.

Indices

t period index, $t = 1, \ldots, T$, with T the time horizon

j age index, $j = 1, \ldots, J$, with J the fixed shelf life

[*] This paper has been supported by The Spanish Ministry (projects TIN2008-01117, TIN2012-37483-C03-01) and Junta de Andalucía (P11-TIC-7176), in part financed by the European Regional Development Fund (ERDF). The study is co-funded by the TIFN (project RE002).

Data
$\boldsymbol{d}_t$ Normally distributed demand with expectation $\mu_t > 0$ and variance $(cv \times \mu_t)^2$ where cv is a given coefficient of variation
k fixed ordering cost, $k > 0$
c unit procurement cost, $c > 0$
h unit inventory cost, $h > 0$
w unit disposal cost, is negative when having a salvage value, $w > -c$
β service level, $0 < \beta < 1$

Variables

$Q_t \geq 0$ ordered and delivered quantity at the beginning of period t
$Y_t \in \{0, 1\}$ setup of order
$\boldsymbol{I}_{jt}$ Inventory of age j at end of period t, initial inventory fixed $I_{j0} = 0$, $\boldsymbol{I}_{jt} \geq 0$ for $j = 1, \ldots, J$

The total expected costs over the finite horizon is to be minimized.

$$f(Q) = \sum_{t=1}^{T} \left(C(Q_t) + E\left(h \sum_{j=1}^{J-1} \boldsymbol{I}_{jt} + w\boldsymbol{I}_{Jt} \right) \right), \tag{1}$$

where procurement cost is given by the function

$$C(x) = k + cx, \text{ if } x > 0, \text{ and } C(0) = 0. \tag{2}$$

The FIFO dynamics of inventory of items of different age j starts by defining waste

$$\boldsymbol{I}_{Jt} = (\boldsymbol{I}_{J-1,t-1} - d_t)^+, \; t = 1, \ldots, T, \tag{3}$$

followed by the inventory of other ages that still can be used in the next period:

$$\boldsymbol{I}_{jt} = \left(\boldsymbol{I}_{j-1,t-1} - (\boldsymbol{d}_t - \sum_{i=j}^{J-1} \boldsymbol{I}_{i,t-1})^+ \right)^+, \; t = 1, \ldots, T, \; j = 2, \ldots, J-1. \tag{4}$$

and finally the incoming and freshest products, with $j = 1$:

$$\boldsymbol{I}_{1t} = \left(Q_t - (\boldsymbol{d}_t - \sum_{j=1}^{J-1} \boldsymbol{I}_{j,t-1})^+ \right)^+, \; t = 1, \ldots, T. \tag{5}$$

Lost sales for period t is defined by

$$\boldsymbol{X}_t = \left(\boldsymbol{d}_t - \sum_{j=1}^{J-1} \boldsymbol{I}_{j,t-1} - Q_t \right)^+. \tag{6}$$

The service level constraint for every period is

$$E(X_t) \leq (1 - \beta)\mu_t, \; t = 1, \ldots, T \tag{7}$$

Notice that the incoming products are the freshest product, $j = 1$. We consider a simple order policy, where the decision maker is provided a list of order periods Y_t and order quantities Q_t where $Y_t = 0$ implies $Q_t = 0$. This can be considered an MINLP problem to derive what are the optimal values of the (continuous) order quantities Q_t and the corresponding optimal (integer) order timing Y.

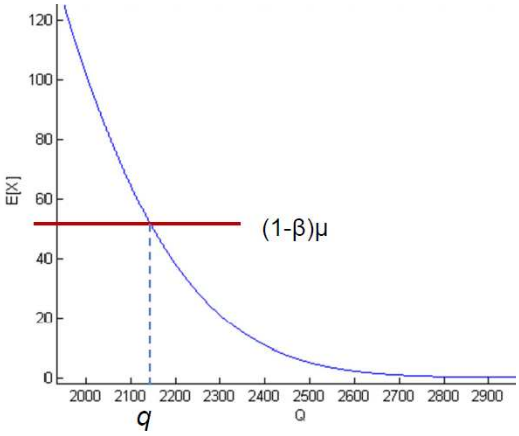


Figure 1: One period loss function for $\boldsymbol{d} \sim N(1950, 0.25 \cdot 1950)$ and corresponding basic order quantity q.

3. Replenishment cycles and basic order quantities

We study several theoretical properties of the order quantities Q and the list of order periods Y. We first focus on the concept of replenishment cycles and determine in which cases a so-called basic order quantity defines the optimal order quantity in Section 3.2.

3.1 Feasible replenishment cycles

Literature on inventory control e.g. [5] applies the concept of a replenishment cycle, i.e. the length of the period R for which the order of size Q is meant. For stationary demand, the replenishment cycle is fixed, but for non-stationary demand the optimal replenishment cycle R_t may depend on the period.

Definition 1. *Given list of order periods $Y \in \{0,1\}^T$ and $N = \sum_{t=1}^{T} Y_t$. The order timing vector $A(Y) \in \mathbb{N}^N$ gives the order moments $A_i < A_{i+1}$ such that $Y_{A_i} = 1$.*

Definition 2. *Given list of order periods $Y \in \{0,1\}^T$ and $N = \sum_{t=1}^{T} Y_t$ Replenishment cycle $R_i(Y) = A_{i+1} - A_i,\ i = 1, \dots, N-1$ and $R_N = T - R_n + 1$.*

Notice that for the perishable case with a shelf life J, to fulfil the service level constraint, practically the replenishment cycle can not be larger than the shelf life J; so $R_i \leq J$.

Lemma 3. *Let Y be an order timing vector of the SP model, i.e. $Y_t = 0 \Rightarrow Q_t = 0$. Y provides an infeasible solution of the SP model, if it contains more than $J-1$ consecutive zeros.*

This means that a feasible order timing vector Y does not contain a consecutive series with more than $J-1$ zeros.

3.2 Basic order quantities

Consider a replenishment cycle of one period $R = 1$, zero inventory and the order quantity q that minimizes the cost function such that the service level constraint (7) is fulfilled. The expected lost sales $L(q)$ is

$$L(q) = E(\boldsymbol{d} - q)^{+} = \int_{q}^{\infty} (x-q) f(x) dx \tag{8}$$

where f is the density function of $\boldsymbol{d}$. L is known as the loss function.

The cost function is monotonously increasing in the order quantity Q, so in order to minimize it we need to find q such that $L(q) = (1 - \beta)\mu$ as illustrated in Figure 1. Since demand is normally distributed, the solution has to be calculated numerically. Here there are several ways to proceed. One can use the derivative of loss function $L'(q) = \int\limits_{-\infty}^{q} f(x)dx - 1 = F(q) - 1$ to approximate q using *Newton Raphson*. For the described model, the determination of q, only has to be done once.

Lemma 4. *Let $\boldsymbol{d} \sim N(\mu_1, cv \times \mu)$ and φ be the pdf and Φ the cdf of the standard normal distribution. The solution of $L_d(q) = (1-\beta)\mu$ fulfils $q = \mu(1+cv \times \hat{q})$ where $\hat{q}$ solves $\varphi(\hat{q}) - (1 - \Phi(\hat{q}))\,\hat{q} = \frac{1-\beta}{cv}$.*

Proof. Using the results in [4] for $\boldsymbol{d} \sim N(\mu, cv \times \mu)$, the loss function can be expressed as

$$L_d(q) = cv \times \mu \left(\varphi\left(\frac{q-\mu}{cv \times \mu}\right) - \left(1 - \Phi\left(\frac{q-\mu}{cv \times \mu}\right)\right)\frac{q-\mu}{cv \times \mu}\right). \tag{9}$$

The equation $L(q) = (1 - \beta)\mu$ substituting $q = \mu(1 + cv \times \hat{q})$ implies

$$\varphi\left(\frac{q-\mu}{cv \times \mu}\right) - \left(1 - \Phi\left(\frac{q-\mu}{cv \times \mu}\right)\right)\frac{q-\mu}{cv \times \mu} = \varphi(\hat{q}) - (1 - \Phi(\hat{q}))\hat{q} = \frac{1-\beta}{cv}. \tag{10}$$

□

The basic order quantity $\overline{Q}_{1t} = \mu_t(1 + cv \times \hat{q})$ provides an upper bound on the order quantity Q_t if $R_t = 1$, because inventory may be available. The basic order quantities for longer replenishment cycles are far more complicated; $R_t = 2$ implies

$$E\left(\boldsymbol{d}_{t+1} - (\overline{Q}_{2t} - \boldsymbol{d}_t)^+)\right)^+ = (1 - \beta)\mu_{t+1}$$

and $R_t = 3$ implies

$$E\left((\boldsymbol{d}_{t+2} - ((\boldsymbol{d}_{t+1} - (\overline{Q}_{3t} - \boldsymbol{d}_t)^+)^+)\right)^+ = (1 - \beta)\mu_{t+2},$$

where we also have to take the constraint $\overline{Q}_{1t} \leq \overline{Q}_{2t} \leq \overline{Q}_{3t}$ into account. These basic order quantities can only be found by simulation.

4. Conclusions

An MINLP model has been presented to determine order quantities for a perishable product inventory control problem. So far, basic order quantities can be determined that provide a feasible policy of the model. The next question is how given this starting policy to find the optimal order quantities and order timing for the problem.

References

[1] R. Hedjar, M. Bounkhel, and L. Tadj. Predictive control of periodic-review production inventory systems with deteriorating items. *Top*, 12(1):193–208, 2004.

[2] K. G. J. Pauls-Worm, E. M. T. Hendrix, R. Haijema, and J. G. A. J. van der Vorst. Inventory control for a perishable product with non-stationary demand and service level constraints. *Working paper Optimization Online, www.optimization-online.org/DB_FILE/2013/08/4010.pdf*, 2013.

[3] K. G. J. Pauls-Worm, E. M. T. Hendrix, R. Haijema, and J. G. A. J. van der Vorst. Inventory control for a perishable product. *International Journal of Production Economics*, 2014.

[4] R. Rossi, S. A. Tarim, S. Prestwich, and B. Hnich. Piecewise linear approximations of the standard normal first order loss function. Technical report, arXiv:1307.1708, 2013.

[5] E. A. Silver, D. F. Pyke, and R. Peterson. *Inventory Management and Production Planning and Scheduling*. Wiley, 1998.

On Benchmarking Stochastic Global Optimization Algorithms

Algirdas Lančinskas[1], Eligius M.T. Hendrix[2], and Julius Žilinskas[1]

[1] *Institute of Mathematics and Informatics, Vilnius University, Vilnius, Lithuania,* algirdas.lancinskas@mii.vu.lt

[2] *Department of Computer Architecture, Universidad de Málaga and*
Operations Research and Logistics Group, Wageningen University, eligius.hendrix@wur.nl

Abstract A multitude of heuristic stochastic optimization algorithms with a plethora of fantasy names have been published to obtain good solutions of the box-constrained global optimization problems often with a limit on the used function evaluations. In the larger question of which algorithms behave well on which type of instances, our focus is here on the benchmarking of the behavior of algorithms by applying experiments on test instances. We argue that a good minimum performance benchmark is due to pure random search; i.e. algorithms should do better. We introduce the concept of the cumulative distribution function of the record value as a measure with the benchmark of pure random search and the idea of algorithms being dominated by others and illustrate this with available frequently used algorithms.

Keywords: Stochastic Global Optimization, Benchmark, Black-Box, Meta-Heuristic

1. Introduction

We consider the box-constrained global optimization problem

$$f^* = \min_{\mathbf{x} \in X} f(\mathbf{x}), \tag{1}$$

where $f(\mathbf{x})$ is a continuous function, $X \subset \mathbb{R}^n$ is a box constrained feasible region, and n is the number of the problem variables. The idea of the black-box optimization is that function evaluations imply running an external routine that may take minutes or hours to provide the evaluated objective function value. Many times the question is to obtain a good, but not necessarily optimal solution within a day, several days, or a week. The question translates to obtaining a good solution with a limited number N (budget) of function evaluations.

For generating good solutions for such a problem, many stochastic heuristic algorithms have been described in literature; e.g. [4]. Although concepts of simulated annealing and population algorithms already existed for a long time, many algorithms have been developed under the terminology of evolutionary algorithms or meta-heuristics after the appearance of the work [5] on genetic algorithms. Mathematical statistical analysis of the speed of convergence is hindered by complicated algorithm descriptions. Therefore, researchers rely on numerical tests with a set of test problems that have evolved in books and on the Internet after the first set described in [3].

The ultimate question is which types of algorithms perform well on which type of instances; what defines the characteristics of the case to be solved such that one algorithm is more successful than the other? This question requires to investigate for which instances a specific algorithm does not perform well compared to simple benchmarks. We are aware, that in most or all published numerical results of algorithms, the focus on worst case behavior is lacking.

The research question that keeps us busy in this research is how to evaluate the quality of an algorithm for an individual test case. We argue that the performance of Pure Random

Search (PRS) can be taken as a benchmark and focus on the statistical performance in order to measure how much better (or worse) other algorithm performs.

2. Cumulative density of the best point found

In general, a stochastic optimization algorithm generates a series of points x_k that approximate an (or the, or all) minimum point(s). According to the generic description of [8]:

$$\mathbf{x}_{k+1} = Alg(\mathbf{x}_k, \mathbf{x}_{k-1}, \ldots, \mathbf{x}_1, \xi), \tag{2}$$

where ξ is a random variable, and k is the iteration counter. Description (2) represents the idea that a next point $\mathbf{x}_{k+1}$ is generated based on the information in all former points $\mathbf{x}_k, \mathbf{x}_{k-1}, \ldots, \mathbf{x}_1$ and a random effect ξ based on generated pseudo-random numbers. The final result of running an algorithm with N function evaluations on a test function is the random record function value $\mathbf{Y}_N = \min_{k=1,\ldots,N} f(\mathbf{x}_k)$. The quality of an algorithm A with N trials is defined by the cumulative distribution function of the record $\mathbf{Y}_N$ which we will denote by $CDFR_N^{[\mathrm{A}]}(y) = P\{\mathbf{Y}_N \leq y\}$. This concept is three dimensional when we consider the probability, the level y and the budget on function evaluations N and therefore hard to capture in an analysis.

What is done often is to focus simply on the expected value $E(\mathbf{Y}_N)$ as function of the budget N measured as a numerical average. It may be clear that this ignores the variation; for some run (repetition), an algorithm may fail and for another not.

In order to understand the concept, let us first consider the starting point of stochastic global optimization algorithms of sampling one trial point $\mathbf{x}$ uniformly drawn over the feasible region. Consider $\mu(y) = P\{f(\mathbf{x}) \leq y\}$ being the cumulative distribution function of random variable $\mathbf{y} = f(\mathbf{x})$, where $\mathbf{x}$ is uniform over X. So, basically $CDFR_1^{[\mathrm{PRS}]}(y)$ is provided by the function $\mu(y)$ with domain $[f^*, \max_X f(\mathbf{x})]$. For PRS, the probability that a level y is reached after generating N trial points is given by $1-(1-\mu(y))^N$, which in fact defines $CDFR_N^{[\mathrm{PRS}]}(y)$ of pure random search which we will call $P_N(y)$.

$P_N(y)$ provides a benchmark for all stochastic algorithms. For each function value y one should at least reach the probability $P_N(y)$, i.e. what is the difference between $CDFR_N^{[\mathrm{A}]}(y)$ and $P_N(y) = 1-(1-\mu(y))^N$ after having generated N points? One can compare algorithms systematically for instances by comparing their $CDFR_N^{[\mathrm{A}]}(y)$ function.

Another extreme benchmark algorithm in stochastic optimization is Multistart (MS) [1]. It requires a local (nonlinear) optimization routine $\mathrm{LS}(\mathbf{x}_0, N_{\mathrm{LS}})$ as a procedure which given a starting point $\mathbf{x}_0$ and the limit on the number of function evaluations (N_{LS}) returns a point in the domain that approximates a local minimum point. In contrast to PRS, numerical results therefore depend on the LS routine applied. For reproducibility, we will apply a standard MATLAB routine *fmincon*. It is useful to mention that the $CDFR_N^{[\mathrm{MS}]}$ of MS has a typical step shape where the objective values of the local minima reveal a certain probability mass.

Our idea is to have a measure for comparison of two algorithms A and B, to see whether one is performing better on a certain problem instance. It may be clear that algorithm A is doing better than B on an instance for effort N if $\forall y,\ CDFR_N^{[\mathrm{A}]}(y) > CDFR_N^{[\mathrm{B}]}(y)$.

In numerical results, often the focus is on average behavior to determine whether

$$E\left(\mathbf{Y}_N^{[\mathrm{A}]}\right) > E\left(\mathbf{Y}_N^{[\mathrm{B}]}\right). \tag{3}$$

This is typically a necessary but not sufficient condition to determine better performance. If test cases (instances) can be classified into problem classes, the most interesting question is whether the behavior of a particular algorithm B is dominated by that of another algorithm A. It means one can take B out of consideration to solve problems from this class.

Notice that for population based algorithms that initially start with a randomly generated and evaluated population, the behavior of the record value is exactly the same as that of PRS up to the population has been generated and the mechanism of "reproduction", i.e. generating new trial points on the base of the current population, has started. Very low budgets on function evaluations are therefore not very interesting.

On the other hand, it is very well known from literature on stochastic global optimization (e.g. [9]) that if the effort N gets bigger, we are getting closer to the optimum f^* and for lower dimensional cases, the probability of not hitting a level set of level $f^* + \delta$ becomes very small for tens of thousands of points. This means that the difference between algorithms vanishes if one keeps on sampling. We stress this, because we observed tables in literature where two-dimensional instances were hit with tens of thousands of trial points. In the sequel, we will attempt to find an interesting region of budget N and test cases where well known algorithms can be distinguished.

3. Numerical illustration of the new concepts

In order to illustrate the concepts of the cumulative distribution $CDFR_N(y)$ of the record, we elaborate numerical results obtained by solving the Six-Hump Camel Back test problem [4] using Particle Swarm Optimization (PSO) [6], Genetic Algorithm (GA) [5, 2], and Controlled Random Search (CRS) [7], and confront them with the benchmarks of PRS and MS. The algorithms have been run for $N = 200, 500, 1000$ function evaluations with population size of $M = 50$, and repeating each experiment 1000 times. The results are presented in Figure 1, where the x-axis of the graphs is scaled by the maximum objective function value of PRS over 1000 repetitions using the corresponding number N of function evaluations.

One can see from the figure that visually for $N = 1000$ the performance of the population algorithms cannot be distinguished. Results can be better distinguished for the small budget $N = 200$. For this budget, MS can perform only 5 local searches with the MATLAB local search solver; this is well visible in the sense that the global minimum is not always reached. Thinking in terms of "generations", PSO and GA only refresh their population (swarm) four times. Nevertheless, the GA algorithm dominates the others, i.e. its curves are highest for all tested budgets N. None of the population algorithms is worse than PRS.

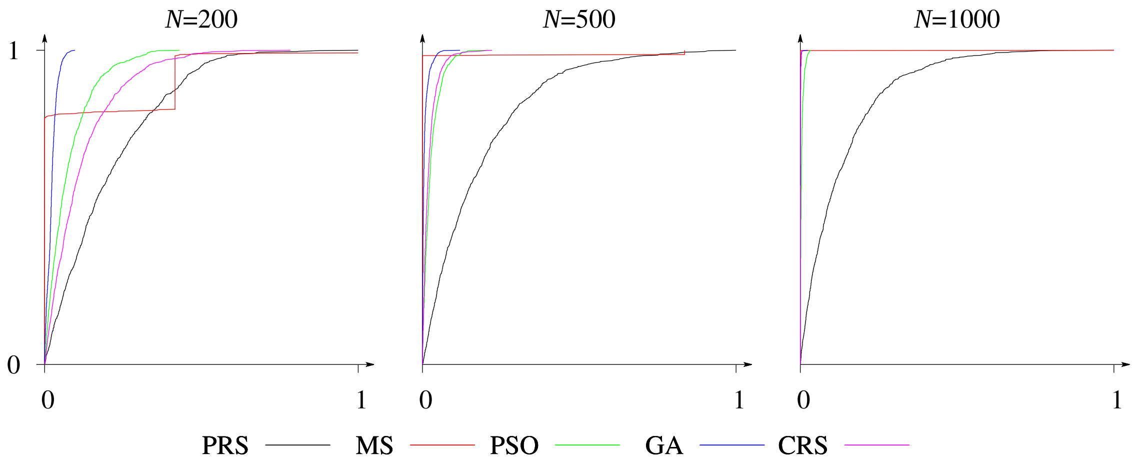


Figure 1: Plots of $CDFR_N^{[A]}(y)$ obtained by running algorithms A =PRS, MS, PSO, GA, CRS on the Six-Hump Camel Back test problem, $N = 200$ (left); $N = 500$ (middle); $N = 1000$ (right).

4. Summary

Heuristics for the box-constrained global optimization problem are often tested on a test-bed of instances. For the question of which algorithms behave well on which type of instances, we showed that the Cumulative Distribution Function of the record value provides the answer on domination. We argue that a good minimum performance benchmark is due to pure random search; i.e. algorithms should do better. The concepts have been illustrated for several well-known heuristic algorithms for global optimization.

Acknowledgments

This paper has been supported by the Spanish state (project TIN2012-37483) and Junta de Andalucía (P11-TIC-7176), in part financed by the European Regional Development Fund (ERDF). This research was funded by a grant (No. MIP-051/2014) from the Research Council of Lithuania.

References

[1] W.P. Baritompa and E.M.T. Hendrix. On the investigation of stochastic global optimization algorithms. *Journal of Global Optimization*, 31:567–578, 2005.

[2] L. Davis. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, 1991.

[3] L.C.W. Dixon and G.P. Szegö. *Towards Global Optimization*. North Holland, Amsterdam, 1975.

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

[5] J.H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, 1975.

[6] J. Kennedy and R.C. Eberhart. Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks*, pages 1942–1948. Piscataway, NJ, 1995.

[7] W.L. Price. A controlled random search procedure for global optimization. *The Computer Journal*, 20:367–370, 1979.

[8] A. Törn and A. Žilinskas. *Global Optimization*, volume 350 of *Lecture Notes in Computer Science*. Springer, Berlin, 1989.

[9] A. Zhigljavsky and A. Žilinskas. *Stochastic Global Optimization*, volume 9 of *Springer Optimization and Its Applications*. Springer, New York, 2008.

Clustering Categories in Support Vector Machines

Emilio Carrizosa[1], Amaya Nogales-Gómez[1], and Dolores Romero Morales[2]

[1]*Departamento de Estadística e Investigación Operativa, Facultad de Matemáticas, Universidad de Sevilla, Spain,* ecarrizosa@us.es, amayanogales@us.es

[2]*Saïd Business School, University of Oxford, United Kingdom,* dolores.romero-morales@sbs.ox.ac.uk

Abstract Support Vector Machines (SVM) is the state-of-the-art in Supervised Classification. A methodology to reduce complexity of the classifier in SVM is proposed by clustering the categories of categorical features. Four strategies are presented based on solving: the original SVM and two mathematical optimization formulations we propose in this talk. An empirical comparison shows the performance of the SVM classifier derived using the original data against that using the clustered data for the 2-cluster case. In the tested datasets our methodology achieves comparable accuracy to that of the SVM with the original data, while we illustrate the dramatic decrease in complexity by clustering the categories.

Keywords: Support Vector Machines, Mixed Integer (Non)Linear Programming, Categorical features, Clustering

1. Introduction

In Supervised Classification, we are given a set of objects Ω partitioned into classes and the aim is to build a procedure for classifying new objects when information about objects in Ω is only available in the so-called *training sample*, with dimension n. In its simplest form, each object $i \in \Omega$ has associated a vector (x_i, x'_i, y_i), where the feature vector x_i associated with J categorical features takes values on a set $X \subseteq \{0,1\}^{\sum_{j=1}^{J} K_j}$, the feature vector x'_i associated with the continuous features takes values on a set $X' \subseteq \mathbb{R}^{J'}$, and $y_i \in \{-1,+1\}$ is the class membership of object i. As seen above, the common approach in the literature is to binarize the different categories, obtaining for each categorical feature one binary feature for each category, that is, categorical feature j that has K_j different categories is split into K_j binary features. This can lead to a loss of information and accuracy because the structure of the original categorical features is disregarded.

In Section 2, the Cluster Support Vector Machines (CLSVM) methodology is introduced together with a Mixed Integer Nonlinear Programming problem (MINLP) formulation and a Mixed Integer Quadratic Programming problem (MIQP) formulation. In Section 3 four strategies are presented with the aim of reducing complexity in Support Vector Machines (SVM) by clustering the categories of categorical features. Section 4 concludes.

2. The CLSVM methodology

In this section, the Cluster Support Vector Machines (CLSVM) methodology is introduced. Then, an MINLP and an MIQP formulations are presented, based on the standard SVM and exploiting the information provided by categorical features. Table 1 shows the used symbols of the methodology.

A state-of-the-art method in Supervised Classification using a score function is the Support Vector Machine (SVM), [6, 11, 12]. The SVM aims at separating both classes by means of a hyperplane, $\omega^\top x + \omega'^\top x' + b = 0$, found by minimizing the squared l_2-norm of the weight

Table 1: Notation for the CLSVM methodology

Notation	Description
$\mathcal{J}$	Set of categorical features with cardinality J
$\mathcal{J}'$	Set of non-categorical features with cardinality J'
$\mathcal{K}_j$	Set of categories for feature j with cardinality K_j
$\mathcal{L}_j$	Set of clusters for feature j with cardinality L_j

vector (ω, ω') and the so-called *hinge loss*, with a regularization paramater C, separating it into two different sums, for J categorical features and J' non-categorical features. See [3] for a recent review on Mathematical Optimization and the SVM, and [1, 2, 4, 5, 7, 8, 9] for successful applications of the SVM.

The methodology proposed, the Cluster Support Vector Machines (CLSVM) methodology, is based on the SVM, but exploits the structure of categorical features. This methodology receives as input a dataset containing categorical features and as a first step, it performs a clustering for each categorical feature, defined by an assignment vector $z_{j,k,\ell}$, equal to 1 if category k from categorical feature j is assigned to cluster ℓ. Then, the dataset is clustered according to z as explained in Figure 1 and a separating hyperplane is obtained for the clustered dataset. The pseudocode of the CLSVM methodology can be found in Figure 2. To avoid symmetry between clustering solutions, the first category of each categorical feature is always assigned to the first cluster.

Figure 1: Pseudocode for the clustered dataset defined by the assignment variable z.

```
For each $i \in \Omega$:
Step 1. Input:
        original object $(y_i, x_i, x'_i)$, $x_i \in \{0,1\}^{\sum_{j=1}^{J} K_j}$, $x'_i \in \mathbb{R}^{J'}$
        assignment variable $z \in \{0,1\}^{\sum_{j=1}^{J} L_j K_j}$
Step 2. Output:
        clustered object $(y_i, \bar{x}_i, x'_i)$, $\bar{x}_i \in \{0,1\}^{\sum_{j=1}^{J} L_j}$, $x'_i \in \mathbb{R}^{J'}$
        where $\bar{x}_i = (\bar{x}_{i,1,1}, \ldots, \bar{x}_{i,J,L_J})$ with $\bar{x}_{i,j,\ell} = \sum_{k=1}^{K_j} z_{j,k,\ell} x_{i,j,k}$
```

Figure 2: Pseudocode for the CLSVM methodology.

```
Given a dataset $\Omega$:
Step 1. Find the assignment vector $z$, defining a clustering for the
        categorical features.
Step 2. Obtain the clustered dataset $\bar{\Omega}$ as in Figure 1.
Step 3. Find a separating hyperplane for $\bar{\Omega}$.
```

Now, we introduce the Cluster Support Vector Machines (CL) formulation, a mixed integer nonlinear problem (MINLP), [10]. It replaces ω with $\bar{\omega}$, which is the score vector associated with the clustered categorical features. For each categorical feature j, we have a subvector $(\bar{\omega}_{j,\ell})\ \ell = 1, \ldots, L_j$, which is the score for the cluster ℓ of the categorical feature j. The CL is formulated as follows:

$$\min_{\bar{\omega}, \omega', b, \xi, z} \sum_{j=1}^{J} \sum_{\ell=1}^{L_j} \frac{\bar{\omega}_{j,\ell}^2}{2} + \sum_{j'=1}^{J'} \frac{\omega_{j'}'^2}{2} + \frac{C}{n} \sum_{i=1}^{n} \xi_i \tag{1}$$

s.t. (CL)

$$y_i \left(\sum_{j=1}^{J} \sum_{\ell=1}^{L_j} \bar{\omega}_{j,\ell} \sum_{k=1}^{K_j} z_{j,k,\ell}\, x_{i,j,k} + \omega'^{\top} x_i' + b \right) \geq 1 - \xi_i \qquad \forall i = 1, \ldots, n \tag{2}$$

$$\xi_i \geq 0 \qquad \forall i = 1, \ldots, n \tag{3}$$

$$\sum_{\ell=1}^{L_j} z_{j,k,\ell} = 1 \qquad \forall j = 1, \ldots, J; \forall k = 1, \ldots, K_j \tag{4}$$

$$z \in \{0,1\}^{\sum_{j=1}^{J} L_j\, K_j} \tag{5}$$

$$\bar{\omega} \in \mathbb{R}^{\sum_{j=1}^{J} L_j} \tag{6}$$

$$\omega' \in \mathbb{R}^{J'} \tag{7}$$

$$b \in \mathbb{R}. \tag{8}$$

In order to obtain an MIQP formulation, one can relax the nonlinear term the product of variables $\bar{\omega}_{j,\ell} \sum_{k=1}^{K_j} z_{j,k,\ell}\, x_{i,j,k}$ in constraint (2) by introducing new *big M* constraints. This implies adding $\sum_{j=1}^{J} L_j K_j$ continuous variables, $\tilde{\omega}$, and $4 \cdot \sum_{j=1}^{J} L_j K_j$ big M constraints, (11)-(14).

$$\min_{\bar{\omega}, \omega', b, \xi, z} \sum_{j=1}^{J} \sum_{\ell=1}^{L_j} \left(\frac{\bar{\omega}_{j,\ell}^2}{2} \right) + \sum_{j'=1}^{J'} \frac{\omega_{j'}'^2}{2} + \frac{C}{n} \sum_{i=1}^{n} \xi_i$$

s.t. (CL-big M)

$$y_i \left(\sum_{j=1}^{J} \sum_{\ell=1}^{L_j} \tilde{\omega}_{j,k(i),\ell} + \omega'^{\top} x_i' + b \right) \geq 1 - \xi_i \qquad \forall i = 1, \ldots, n \tag{9}$$

$$\xi_i \geq 0 \qquad \forall i = 1, \ldots, n$$

$$\sum_{\ell=1}^{L_j} z_{j,k,\ell} = 1 \qquad \forall k = 1, \ldots, K_j, \ \ \forall j = 1, \ldots, J \tag{10}$$

$$\tilde{\omega}_{j,k,\ell} \leq \bar{\omega}_{j,\ell} + M(1 - z_{j,k,\ell}) \qquad \forall k = 1, \ldots, K_j, \ \ \forall \ell = 1, \ldots, L_j, \ \ \forall j = 1, \ldots, J \tag{11}$$

$$\tilde{\omega}_{j,k,\ell} \geq \bar{\omega}_{j,\ell} - M(1 - z_{j,k,\ell}) \qquad \forall k = 1, \ldots, K_j, \ \ \forall \ell = 1, \ldots, L_j, \ \ \forall j = 1, \ldots, J \tag{12}$$

$$\tilde{\omega}_{j,k,\ell} \leq M\, z_{j,k,\ell} \qquad \forall k = 1, \ldots, K_j, \ \ \forall \ell = 1, \ldots, L_j, \ \ \forall j = 1, \ldots, J \tag{13}$$

$$\tilde{\omega}_{j,k,\ell} \geq -M\, z_{j,k,\ell} \qquad \forall k = 1, \ldots, K_j, \ \ \forall \ell = 1, \ldots, L_j, \ \ \forall j = 1, \ldots, J \tag{14}$$

$$z \in \{0,1\}^{\sum_{j=1}^{J} L_j\, K_j} \tag{15}$$

$$\bar{\omega} \in \mathbb{R}^{\sum_{j=1}^{J} L_j} \tag{16}$$

$$\tilde{\omega} \in \mathbb{R}^{\sum_{j=1}^{J} L_j\, K_j} \tag{17}$$

$$\omega' \in \mathbb{R}^{J'} \tag{18}$$

$$b \in \mathbb{R}.$$

3. Strategies

In this section, four different strategies are proposed based on the two mathematical programming formulations, the CL and the CL-big M introduced in Section 2, and on the SVM formulation.

Strategy 1 is based on the original SVM. First, an SVM is solved for the original database, then each categorical feature j is clustered into L_j clusters by clustering the SVM scores. Then, the separating hyperplane is found by solving an SVM for the updated clustered database. Strategy 2 is based on the randomized rounding of the partial solution z from the continuous relaxation of the CL formulation. For each value of C, this strategy solves the continuous relaxation of the CL formulation, where constraint (5) is relaxed to $z \in [0,1]^{\sum_{j=1}^{J} L_j K_j}$. A randomized rounding procedure can be applied to derive the assignment variable z. Strategy 3 is based on solving the CL formulation. For each value of C, this strategy solves to optimality the CL formulation or returns the current solution after a given time limit. The last strategy, Strategy 4, tunes and trains the CL-big M formulation. For each value of C, this strategy solves the CL-big M formulation or returns the current solution after a given time limit.

4. Summary

This talk describes a methodology, two mathematical optimization formulations and four strategies for clustering categorical features in the SVM, and thus to reduce the complexity of the SVM classifier. The strategies have been tested on a test set of benchmark datasets publicly available. Results will be discussed in the talk.

References

[1] D. Bertsimas, M.V. Bjarnadóttir, M.A. Kane, J.Ch. Kryder, R. Pandey, S. Vempala, and G. Wang. Algorithmic prediction of health-care costs. *Operations Research*, 56(6):1382–1392, 2008.

[2] J.P. Brooks. Support vector machines with the ramp loss and the hard margin loss. *Operations Research*, 59(2):467–479, 2011.

[3] E. Carrizosa and D. Romero Morales. Supervised classification and mathematical optimization. *Computers and Operations Research*, 40:150–165, 2013.

[4] M. Cecchini, H. Aytug, G.J. Koehler, and P. Pathak. Detecting management fraud in public companies. *Management Science*, 56(7):1146–1160, 2010.

[5] W. A. Chaovalitwongse, Y.-J. Fan, and R. C. Sachdeo. Novel optimization models for abnormal brain activity classification. *Operations Research*, 56(6):1450–1460, 2008.

[6] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[7] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002.

[8] D. Martens, B. Baesens, T.V. Gestel, and J. Vanthienen. Comprehensible credit scoring models using rule extraction from support vector machines. *European Journal of Operational Research*, 183(3):1466–1476, 2007.

[9] D. Romero Morales and J. Wang. Forecasting cancellation rates for services booking revenue management using data mining. *European Journal of Operational Research*, 202(2):554–562, 2010.

[10] M. Tawarmalani and N. V. Sahinidis. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*. Kluwer Academic Publishers, Boston MA, 2002.

[11] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[12] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

Two-Swarm Cooperative Artificial Fish Algorithm for Bound Constrained Global Optimization*

Ana Maria A.C. Rocha[1], M. Fernanda P. Costa[2], and Edite M.G.P. Fernandes[1]

[1] *Algoritmi Research Centre, University of Minho, Braga, Portugal,* {arocha; emgpf}@dps.uminho.pt

[2] *Centre of Mathematics, University of Minho, Braga, Portugal* mfc@math.uminho.pt

Abstract This study presents a new two-swarm cooperative fish intelligence algorithm for solving the bound constrained global optimization problem. The master population is moved by a Lévy distribution and cooperates with the training population that follows mainly the classical fish behaviors. Some numerical experiments are reported.

Keywords: Global optimization, Swarm intelligence, Artificial fish, Lévy distribution

1. Introduction

In this study we are interested in solving the bound constrained global optimization (GO) problem using a swarm intelligence algorithm that is able to converge to the globally best point in the feasible region and requires a limited computational effort. The problem to be addressed has the form

$$\text{glob} \min_{x \in \Omega} f(x), \tag{1}$$

where f is a continuous nonlinear, possibly nonconvex function, and Ω is the hyperrectangle $\{x \in \mathbf{R}^n : l \leq x \leq u\}$. When solving complex optimization problems, like NP-hard problems, metaheuristics are able to perform rather well and generate good quality solutions in less time than the traditional optimization techniques [3]. Besides the variety of applications in some engineering areas, the motivation for the present study is the pressing and ongoing need to develop efficient algorithms for solving a sequence of problems, like (1), that emerge from a penalty function technique or an augmented Lagrangian based multiplier algorithm for constrained nonconvex global optimization, in reasonable time.

The artificial fish swarm (AFS) algorithm has been previously implemented within augmented Lagrangian paradigms [2, 10], which in turn have been compared with other metaheuristic-based penalty like algorithms to solving constrained GO problems. The numerical results have been shown that the fish swarm intelligence is a promising metaheuristic but further research is demanded so that efficiency can be improved.

2. Two-swarm cooperative paradigm

The present proposal for solving the problem (1) is a variant of the AFS algorithm. This metaheuristic relies on a swarm intelligence based paradigm to construct fish/point movements over the search space while converging to the optimal solution [2, 9, 10]. The new algorithm is termed two-swarm cooperative AFS (2S-AFS) and the crucial idea is to use two swarms

*This work has been supported by FCT (Fundação para a Ciência e Tecnologia, Portugal) in the scope of the projects: PEst-OE/MAT/UI0013/2014 and PEst-OE/EEI/UI0319/2014.

(instead of just one) where each one has its own task and supplies information to the other swarm, when attempting to converge to optimality. Other multi-swarm cooperative algorithms based on a master-slave model can be found in [6, 7]. Hereafter, the terms 'point' and 'population' (of points) will be used to represent (the position of) a fish and the swarm respectively. The position of a point in the space is represented by $x_j \in \mathbf{R}^n$ (the jth point of a population) and m is the number of points in the population. The component i of a point x_j is represented by $(x_j)_i$.

2.1 Classical AFS algorithm

The initial procedure of AFS algorithm consists of randomly generating the points x_j, $j = 1, \ldots, m$, of the population, in Ω. Then, each current point x_j produces the trial point y_j according to the number of points inside its 'visual scope' (VS). This is a closed neighborhood centered at x_j with a positive radius which varies with the maximum distance between x_j and the other points. When the VS is empty, a Random Behavior is performed, and when it is crowded, one of the behaviors, Searching or Random, is performed. However, when the VS is not crowded, one of the four following behaviors is selected: Chasing, Swarming, Searching or Random. The selection depends on the objective function values of x_j when compared with the function value of the best point inside the VS, the central point inside the VS, or a randomly chosen point of the VS. To choose the population for the next iteration, the current x_j and the trial y_j are compared in terms of f. The pseudo-code for the AFS algorithm is presented below.

```
AFS algorithm
{
    randomly generate the population $x_j \in \Omega$, $j = 1, \ldots, m$ and select $x_{best}$;
    while stopping condition is not met   {
      for each $x_j$, $j = 1, \ldots, m$    {
        if ('visual scope' is empty)
          {compute $y_j$ by Random Behavior}
        else if ('visual scope' is crowded)
            {compute $y_j$ by Searching/Random Behavior}
          else
            {compute $y_j$ by Chasing/Swarming/Searching/Random Behavior}.
        if ($f(y_j) \leq f(x_j)$)   {set $x_j = y_j$}    }
      select $x_{best}$ and perform random local search around it;   }
}
```

2.2 Two-swarm cooperative AFS algorithm

In order to improve the capability of searching the space for promising regions where the global minimizers lie, this study presents a new fish swarm-based proposal that defines two-populations, each one with its task goal but always sharing information with the other: one is the master and the other is the training population. The master population aims to explore the search space more effectively, thus defining trial points from the current ones throughout a stable stochastic distribution. Depending on the number of points inside the VS of x_j of the training population, the trial point is mainly produced by the classical AFS behaviors, although in some cases – when the VS is empty and when it is crowded – the stochastic distribution borrowed from the master population is used. The overall best point is shared between both populations. The algorithm is called 2S-AFS. To be able to produce a trial y_j, from the current x_j, ideas like those of Bare-bones particle swarm optimization in [4] and the model for mutation in evolutionary programming [5], may be used:

$$(y_j)_i = \gamma + \sigma \mathbf{Y}_i \tag{2}$$

where γ represents the center of the distribution that may be given by $(x_j)_i$ or $((x_j)_i+(x_{best})_i)/2$ (the average of $(x_j)_i$ and the best point $(x_{best})_i$), σ may represent an adaptive mutation defined by the distance between $(x_j)_i$ and $(x_{best})_i$, and each $\mathbf{Y}_i$ is an identically distributed random variable from the Gaussian distribution with mean 0 and variance 1. We note that $\mathbf{Y}$ may be the random variable of another probability distribution. The standard Lévy distribution is used since it can search a wider area of the search space and generate more distinct values in the search space than the Gaussian distribution. The Lévy distribution, denoted by $\mathbf{L}_i(\alpha,\beta,\gamma,\sigma)$, is characterized by four parameters. The parameter β gives the skewness ($\beta = 0$ means that the shape is symmetric relative to the mean). The shape of the Lévy distribution can be controlled with α. For $\alpha = 2$ it is equivalent to the Gaussian distribution, whereas for $\alpha = 1$ it is equivalent to the Cauchy distribution. The distribution is stable for $\alpha = 0.5$ and $\beta = 1$. σ is the scale parameter and is used to describe the variation relative to the center of the distribution. The location parameter γ gives the center. When $\gamma = 0$ and $\sigma = 1$, we get the standard form, simply denoted by $\mathbf{L}(\alpha)$ when $\beta = 0$.

Hence, the proposal for further exploring the search space and improve efficiency is the following. The points from the master population always move according to the Lévy distribution, i.e., each trial point y_j is generated component by component $i = 1,\ldots,n$ as follows:

$$(y_j)_i = \begin{cases} (x_j)_i + (\sigma_j)_i\mathbf{L}_i(\alpha) & \text{if } rand() \leq p \\ (x_{best})_i + (\sigma_j)_i\mathbf{L}_i(\alpha) & \text{otherwise} \end{cases} \tag{3}$$

where $(\sigma_j)_i = |(x_j)_i - (x_{best})_i|$, $\mathbf{L}_i(\alpha)$ is a random number generated for each i from the standard Lévy distribution with the parameter $\alpha = 0.5$, $rand()$ is a random number generated uniformly from $[0,1]$ and p is a user specified probability value for sampling around the best point to occur. On the other hand, each point in the training population either moves according to classical AFS behaviors if its VS is not crowded, or it moves using a Lévy distribution, as shown in (3), with $p = 0$ if the VS is empty, and $p = 1$ if the VS is crowded. Cooperation from the master population is also required if the best point belongs to the master population. The below presented algorithm is the pseudo-code for 2S-AFS algorithm.

```
2S-AFS algorithm
{
      randomly generate $x_j \in \Omega$, $j \in P \equiv \{1,\ldots,m\}$ and select $x_{best}$;
      randomly choose $x_j, j \in M \subset P$, where $\#M = \lfloor \frac{m}{3} \rfloor$, and move them according to (3) with $p = 0$;
      while stopping condition is not met   {
         for each $x_j$, $j = 1,\ldots,m$    {
           if ($j \in M$ – point in master population)
             {compute $y_j$ according to (3) with $p = 0.5$}
           else if ('visual scope' is empty)
                 {compute $y_j$ according to (3) with $p = 0$}
               else if ('visual scope' is crowded)
                     {compute $y_j$ according to (3) with $p = 1$}
                   else
                     {compute $y_j$ by Chasing/Swarming/Searching/Random Behavior}.
           if ($f(y_j) \leq f(x_j)$) {set $x_j = y_j$}   }
        select $x_{best}$ and perform random local search around it;   }
}
```

The algorithm stops when $|f(x_{best}) - f^*| \leq 0.001$ or $NFeval > 20000$ where $f(x_{best})$ is the best solution found thus far, f^* is the known optimal solution, and $NFeval$ gives the number of function evaluations.

3. Results and Conclusions

This section aims to compare the results of the proposed 2S-AFS with those of two AFS-based algorithms on benchmark problems with acronyms BR, CB6, GP, H3, H6, SBT, S5, S7 and S10 (with n ranging from 2 to 6) [1]. The algorithm was coded in C and the results were obtained on a PC with a 2.8 GHz Core Duo Processor P9700 and 6 GB of memory. Each problem was solved 30 times and $m = 10n$ points are used. Table 1 summarizes the results obtained in terms of the average number of function evaluations ('Nfe_{avg}') required by the algorithms to reach the optimal solution with the above defined accuracy. 'DbAFS' is a distribution-based AFS algorithm with the random local search (RLS) (see in [8]) and 'AFS' is the classical AFS with the same RLS (see also Figure 1). From the results we may conclude that 2S-AFS is quite efficient in converging to the optimal f^* on six problems but reached the maximum number of evaluations in some runs when solving problems S5, S7 and S10. These behaviors need further investigation and new strategies to enforce convergence.

	f^*	2S-AFS	DbAFS	AFS
BR	0.39789	362	690	815
CB6	-1.0316	241	293	639
GP	3.00000	494	710	830
H3	-3.86278	206	911	1273
H6	-3.32237	657	3864	6534
SBT	-186.731	415	1256	2803
S5	-10.1532	8382	1611	4568
S7	-10.4029	5793	1818	2931
S10	-10.5364	5837	1889	3067

Table 1: Comparison of AFS-based algorithms.

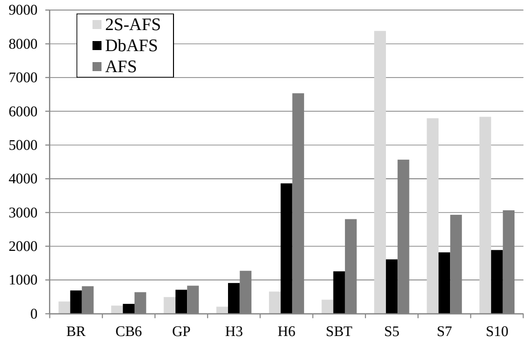


Figure 1: Bars of Nfe_{avg} for the tested algorithms.

References

[1] Ali, M.M., Khompatraporn, C., Zabinsky, Z. B. (2005). *A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems*. J. Glob. Optim. 31, 635–672.

[2] Costa, M.F.P., Rocha, A.M.A.C., Fernandes, E.M.G.P. (2014). *An artificial fish swarm algorithm based hyperbolic augmented Lagrangian method*. J. Comput. Appl. Math. 259, 868–876

[3] Gogna, A., Tayal, A. (2013). *Metaheuristics: review and application*. J. Exp. Theor. Artif. In. 25, 503–526

[4] Kennedy, J., Eberhart, R. (2001). *Swarm Intelligence*. San Mateo, CA: Morgan Kaufmann

[5] Lee, C., Yao, X. (2004). *Evolutionary programming using the mutations based on the Lévy probability distribution*. IEEE T. Evolut. Comput. 8, 1–13

[6] Niu, B., Zhu, Y., He, X., Wu, H. (2007). *MCPSO: A multi-swarm cooperative particle swarm optimizer*. Appl. Math. Comput. 85, 1050–1062

[7] K.E. Parsopoulos, K.E. (2012). *Parallel cooperative micro-particle swarm optimization: A master–slave model*. Appl. Soft Comput. 12, 3552–3579

[8] Rocha, A.M.A.C., Costa, M.F.P., Fernandes, E.M.G.P. (2013). *Distribution based artificial fish swarm in continuous global optimization*. Proc. XVI Congress APDIO, Oliveira, J.F., Vaz, C.B. (eds.) ISBN: 78-972-745-154-8, 306–312, Portugal, June 2013.

[9] Rocha, A.M.A.C., Fernandes, E.M.G.P., Martins, T.F.M.C. (2011) *Novel fish swarm heuristics for bound constrained global optimization problems*. Lect. Notes Comput. Sc. Vol. 6784, ICCSA 2011 Part III, B. Murgante et al. (eds.) 185–199

[10] Rocha, A.M.A.C., Martins, T.F.M.C., Fernandes, E.M.G.P. (2011). *An augmented Lagrangian fish swarm based method for global optimization*. J. Comput. Appl. Math. 235(16), 4611–4620

An extended supporting hyperplane algorithm for convex MINLP problems

Andreas Lundell, Jan Kronqvist, and Tapio Westerlund

Optimization and Systems Engineering, Åbo Akademi University,
Piispankatu 8, FI-20500 Turku, Finland, andreas.lundell@abo.fi

Abstract The extended cutting plane algorithm (ECP) is a deterministic optimization method for solving convex mixed-integer nonlinear programming (MINLP) problems to global optimality as a sequence of mixed-integer linear programming (MILP) problems. The algorithm is based on Kelley's cutting plane method for continuous nonlinear programming (NLP) problems. In this paper, an extended supporting hyperplane (ESH) algorithm is presented. It is based on similar principles as in the ECP algorithm, however instead of utilizing cutting planes supporting hyperplanes are generated.

Keywords: Convex MINLP, Extended supporting hyperplane (ESH) algorithm, Extended cutting plane (ECP) algorithm, Supporting hyperplanes, Cutting planes

1. Introduction

Solving convex MINLP problems efficiently may still be a difficult task even if there currently are several versatile solution algorithms available, such as outer approximation [2, 4], general Bender's decomposition [6], branch and bound techniques using different NLP subsolvers [1, 9] and the ECP algorithm [13]. Extensions of these algorithms are also available, *e.g.,* the ECP algorithm has been extended to handle quasi- and pseudoconvex [14] and nondifferentiable [3] MINLP problem classes. Various implementations of the algorithms exist and many are available in modeling frameworks or optimization systems like GAMS (www. gams. com), COIN-OR (www. coin-or. org) or the NEOS Server (www. neos-server. org). A review of MINLP methods can be found in [7]. The stability and efficiency of MINLP solvers is of paramount importance especially when utilized in real-world applications. Global solution techniques for nonconvex MINLP problems may also require convex MINLP subsolvers [5, 10], and then the performance of the parent solver is largely dependent on that of its subsolver.

In this paper, a new convex MINLP solution technique — the ESH algorithm — is proposed. It is loosely based on the ECP algorithm (itself an extension of Kelley's method in [8]) and has some similarities to the supporting hyperplane method in [12]. In the ECP algorithm MILP problems are iteratively solved until all nonlinear constraints of the MINLP problem are fulfilled to a given tolerance. In each iteration, the feasible region of the MILP problem is reduced by adding cutting planes. Each MILP solution provides a lower bound on the optimal solution of the MINLP problem. Important for the efficiency of cutting plane based algorithms is how and where cutting planes are generated. In the ECP algorithm, the solution point of the MILP problem is directly used, however, in the ESH algorithm only hyperplanes on the boundary of the nonlinear feasible region are generated. Two preprocessing steps to rapidly generate supporting hyperplanes, solving linear programming (LP) problems (instead of MILP problems) together with a line search strategy for selecting the generation point, are also used.

2. The extended supporting hyperplane algorithm

The ESH algorithm, described in this section, has connections to the ECP method [13], however instead of cutting planes, it is based on generating supporting hyperplanes. It also uses two preprocessing steps to efficiently get a tight linear approximation of the feasible region of the convex MINLP problem to be solved and thereafter finally one or a few MILP problems are solved to satisfy the integer requirements.

The ESH algorithm can be used to find the optimal solution x^* to the convex MINLP problem

$$x^* = \underset{x \in C \cap L \cap Y}{\arg\min}\; c^T x \tag{P}$$

where $x = [x_1, x_2, \dots, x_N]^T$ is a vector of variables in a bounded set

$$X = \{x \mid \underline{x}_i \leq x_i \leq \overline{x}_i,\ i = 1, \dots, N\} \tag{1}$$

and the feasible set $L \cap C \cap Y$ is defined by

$$\begin{aligned} C &= \{x \mid g_m(x) \leq 0,\ m = 1, \dots, M,\ x \in X\}, \\ L &= \{x \mid Ax \leq a,\ Bx = b,\ x \in X\}, \\ Y &= \{x \mid x_i \in \mathbb{Z},\ i \in I_{\mathbb{Z}},\ x \in X\}. \end{aligned} \tag{2}$$

X is a compact set of an N-dimensional Euclidean space $X \subset \mathbb{R}^N$ restricted by the variable bounds. The sets L and C are the convex regions satisfying the linear and (convex) nonlinear constraints respectively. If the problem is a NLP problem, $I_{\mathbb{Z}} = \varnothing$ and $Y = X$. If the variable vector x contains integer variables x_i included in the index set $I_{\mathbb{Z}}$, then Y corresponds to the nonrelaxed values these variables can assume. The objective function is written in linear form. In case of a nonlinear convex objective function f, a new objective function constraint $f(x) - x_{N+1} \leq 0$ is included in C and the objective is to minimize the auxiliary variable x_{N+1}.

2.1 NLP step

In the ESH algorithm an internal point $\tilde{x}_{\text{NLP}}$ is first obtained from the convex NLP problem

$$\tilde{x}_{\text{NLP}} = \underset{x \in X}{\arg\min}\, F(x), \quad \text{where } F(x) := \max_m \{g_m(x)\} \tag{P-NLP}$$

using a suitable method [11]. Observe that F is minimized within the region defined by variable bounds only. Since F is given by a max-function, it is convex if all constraint functions g_m are convex and generally quasiconvex if the functions g_m give rise to convex level sets. Note that (P-NLP) may be a nonsmooth problem if $M > 1$ even if all functions g_m are smooth. Assuming that (P) has a solution, there exists a solution to (P-NLP) such that $F(\tilde{x}_{\text{NLP}}) \leq 0$. After this step go to the first preprocessing step in Section 2.2. Note that it is not necessary to solve (P-NLP) to optimality if a strict feasible solution $F(\tilde{x}_{\text{NLP}}) < 0$ is obtained easier.

2.2 LP1 step

After the solution to (P-NLP) is obtained, a first iterative preprocessing step is performed where simple LP problems are solved (initially in X) and a line search procedure is conducted to obtain a tight overestimated set Ω_k of the convex set C. Initially, the counters $k = 1$, $J_0 = 0$, the set $\Omega_0 = X$, and the following relaxation of (P), only considering the variable bounds, is solved:

$$\tilde{x}^k_{\text{LP}} = \underset{\Omega_{k-1}}{\arg\min}\, c^T x. \tag{P-LP1}$$

Assuming there exists a solution to (P), then $F(\tilde{x}^k_{\text{LP}}) > 0$ or $F(\tilde{x}^k_{\text{LP}}) \leq 0$. In the latter case, stop iteration and go to the LP2 step in Section 2.3. Otherwise, *i.e.*, if $F(\tilde{x}^k_{\text{LP}}) > 0$, then the values $F(\tilde{x}_{\text{NLP}})$ and $F(\tilde{x}^k_{\text{LP}})$ have different signs (or $F(\tilde{x}_{\text{NLP}})$ is already equal to zero) and it is possible to obtain points to generate new supporting hyperplanes.

The set Ω_k where $C \subset \Omega_k$ is now defined as an ordered set defined by the J_k first supporting hyperplanes, *i.e.*,

$$\Omega_k = \{x \mid l_j(x) \leq 0,\ j = 1, \ldots, J_k,\ x \in X\}. \tag{3}$$

After solving (P-LP1), a line search is performed between $\tilde{x}_{\text{NLP}}$ and $\tilde{x}^k_{\text{LP}}$, *i.e.*, the equation

$$x^k = \lambda \tilde{x}_{\text{NLP}} + (1 - \lambda)\tilde{x}^k_{\text{LP}}, \tag{4}$$

is used to find the value of $\lambda = \lambda_F \in [0, 1]$ for which $F(x^k) = 0$. (In case $F(\tilde{x}_{\text{NLP}}) = 0$, then $\lambda_F = 1$). In the point x^k a supporting hyperplane

$$l_k = F(x^k) + \xi_F(x^k)^T(x - x^k) \leq 0 \tag{5}$$

is generated and added to Ω_k. $\xi_F(x^k)^T$ is a gradient or subgradient of the corresponding function F at x^k. The counter J_k is increased by one if the line search is performed on F and one supporting hyperplane thus only created for F. Supporting hyperplanes can also be added for other constraints where $g_m(x^k) > 0$. From the line search, it can be observed that for a violated constraint $0 < \lambda_m \leq \lambda_F$. If supporting hyperplanes are generated for a certain number of violated constraints (where $g_m(\tilde{x}^k_{\text{LP}}) > 0$), they can be selected based on decreasing values of the λ_m-values (from equation (4)) starting from $\lambda_m \leq \lambda_F$. The number of hyperplanes added at iteration k is $J_k - J_{k-1}$, where J_k is the total number of hyperplanes in Ω_k.

The problem (P-LP1) is repeatedly resolved (increasing the counter k with one for the next iteration) until a maximum number of iterations has been reached, *i.e.*, $k > K_{\text{LP1}}$, or until $F(\tilde{x}^k_{\text{LP}}) < \epsilon_{\text{LP1}}$ or Ω_k has reached a maximum number of supporting hyperplanes, *i.e.*, $J_k > J_{\text{LP1}}$. Then continue to the LP2 step in Section 2.3.

2.3 LP2 step

In this preprocessing step, a corresponding problem to (P-LP1), with the linear constraints in L included, is solved:

$$\tilde{x}^k_{\text{LP}} = \underset{\Omega_{k-1} \cap L}{\arg\min}\, c^T x. \tag{P-LP2}$$

The solution $\tilde{x}^k_{\text{LP}}$ gives $F(\tilde{x}^k_{\text{LP}}) > 0$ or $F(\tilde{x}^k_{\text{LP}}) \leq 0$. In the latter case, $\tilde{x}^k_{\text{LP}}$ is an optimal solution of (P) if it is a continuous problem. Otherwise $\tilde{x}^k_{\text{LP}}$ is an integer-relaxed solution and we continue to the MILP step in Section 2.4. If $\tilde{x}^k_{\text{LP}} > 0$, the same line search procedure and supporting hyperplane generation strategy as in the LP1 step is performed. Then k and J_k are increased and (P-LP2) is resolved. This continues until a maximum number of iterations has been reached, *i.e.*, $k > K_{\text{LP2}}$, or until $F(\tilde{x}^k_{\text{LP}}) < \epsilon_{\text{LP2}}$ or Ω_k has reached a maximum number of supporting hyperplanes, *i.e.*, $J_k > J_{\text{LP2}}$. After the preprocessing steps LP1 and LP2 have been performed, the set $L \cap C$ is already tightly overestimated by Ω_k. When solving a convex MINLP problem, the integer requirements should be considered. This is finally done in the MILP step. In case the original problem is continuous, terminate with $\tilde{x}^k_{\text{LP}}$ as the solution.

2.4 MILP step

In the final step of the ESH algorithm the integer requirements in (P) are considered by solving MILP relaxations of (P) in Ω_k, L and Y. The problems solved in this step are, thus, defined as

$$\tilde{x}^k_{\text{MILP}} = \underset{\Omega_{k-1} \cap L \cap Y}{\arg\min}\ c^T x. \tag{P-MILP}$$

Note that it is not necessary to solve the MILP problem to optimality in each iteration, the final MILP iteration need however be solved to optimality to guarantee that the solution is the global optimal one. Here the same MILP solution strategy as in [14] can be used.

If the termination criterion $F(\tilde{x}^k_{\text{MILP}}) < \epsilon_{\text{MILP}}$ is not fulfilled, more supporting hyperplanes are added to Ω_k similarly to the LP1 and LP2 steps, and the counters k and J_k are increased. If $F(\tilde{x}^k_{\text{MILP}}) < \epsilon_{\text{MILP}}$ and $\tilde{x}^k_{\text{MILP}}$ is a MILP optimal point, *i.e.*, $\tilde{x}^k_{\text{MILP}} \in Y$, then $\tilde{x}^k_{\text{MILP}}$ is the global solution of the original problem (P) (to a tolerance of ϵ_{MILP}) in a finite number of steps.

3. Conclusions

In this paper, an ESH algorithm for convex MINLP problems was presented. It incorporates two preprocessing steps utilizing LP to iteratively refine a set Ω including supporting hyperplanes rendering a tighter linear overestimation $\Omega_0 \supseteq \Omega_1 \supseteq \cdots \supseteq \Omega_k \supseteq C \subseteq C \cap L$ of the convex sets C and $C \cap L$. A MINLP optimal solution is finally guaranteed by subsequentially solving MILP relaxations including the integer restrictions and adding additional hyperplanes to Ω.

Acknowledgments

Financial support from the Foundation of Åbo Akademi University, as part of the grant for the Center of Excellence in Optimization and Systems Engineering, is gratefully acknowledged. AL also acknowledges financial support from the Ruth and Nils-Erik Stenbäck Foundation.

References

[1] R. J. Dakin. A tree-search algorithm for mixed integer programming problems. *The Computer Journal*, 8(3):250–255, 1965.

[2] M. A. Duran and I. E. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36(3):307–339, 1986.

[3] V.-P. Eronen, M. M. Mäkelä, and T. Westerlund. Extended cutting plane method for a class of nonsmooth nonconvex MINLP problems. *Optimization*, (available online):1–21, 2013.

[4] R. Fletcher and S. Leyffer. Solving mixed integer nonlinear programs by outer approximation. *Mathematical Programming*, 66(1-3):327–349, 1994.

[5] C.A. Floudas and C.E. Gounaris. A review of recent advances in global optimization. *Journal of Global Optimization*, 45(1):3–38, 2009.

[6] A. M. Geoffrion. Generalized Benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, 1972.

[7] I. E. Grossmann. Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and Engineering*, 3(3):227–252, 2002.

[8] J. Kelley, Jr. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.

[9] S. Leyffer. Integrating SQP and branch-and-bound for mixed integer nonlinear programming. *Computational Optimization and Applications*, 18(3):295–309, 2001.

[10] A. Lundell, A. Skjäl, and T. Westerlund. A reformulation framework for global optimization. *Journal of Global Optimization*, 57(1):115–141, 2013.

[11] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*. Kluwer Academic Publishers, 2004.

[12] A. F. Veinott Jr. The supporting hyperplane method for unimodal programming. *Operations Research*, 15(1):147–152, 1967.

[13] T. Westerlund and F. Pettersson. An extended cutting plane method for solving convex MINLP problems. *Computers & Chemical Engineering*, 19:131–136, 1995.

[14] T. Westerlund and R. Pörn. Solving pseudo-convex mixed-integer problems by cutting plane techniques. *Optimization and Engineering*, 3:253–280, 2002.

Solution methods for expensive optimization problems

Christine Edman

Department of Mathematics, University of Trier, 54286 Trier, Germany. edman@uni-trier.de

Abstract We consider expensive optimization problems, that is to say problems, where each evaluation of the objective function is expensive in terms of computing time, consumption of resources, or cost. This often happens in situations where the objective function is not available in analytic form. Therefore it is of central importance to use as few evaluations as possible within the optimization process. This necessitates a sophisticated strategy to determine the evaluation points. We discuss response surface methods that are tailored to the problems described above.

Keywords: black box optimization, global optimization, RBF-method

1. Introduction

In this talk we consider expensive optimization problems, that is to say problems where each evaluation of the objective function is expensive in terms of computing time, consumption of resources, or cost. This often happens in situations where the objective function is not available in analytic form. In this context we will study the global optimization problem $\min f(x)$ such that $x \in R$. Here the continuous function $f : \mathbb{R}^d \to \mathbb{R}$ is the expensive objective function and R is a d-dimensional rectangle. These problems arise in:

- traffic planning,
- crash tests,
- best composition of chemicals,
- soil contamination.

These examples show that we can generally see this ′expensiveness′ as a lack of analytical representation, which is why we also speak about ′black box functions′. Due to this expensivness it is of central importance to use as few function evaluations as possible within the optimization process. This necessitates a sophisticated strategy to determine the evaluation points.

In such a situation the use of response surface methods is advisable. Here at each stage the few already sampled points of the expensive function are used to create a model by a linear combination of special basis functions, matching the already given points. See Figure 1 for an example where the expensive function f is interpolated by the model function s_4 which is based on four sample points.

Based on this interpolating model, the next point to be evaluated is chosen. A good survey of response surface methods is [7]. Obviously, not only the surface model should be used to determine the global optimum of the expensive function because it mainly supports the local search within the optimization method. Instead, a further criterion is needed to improve the global search as well. Here the so called ′measure of smoothness′ [11] provides a useful quantity to determine the next iteration point. The following so called ′radial basis functions′ were investigated intensively by several authors ([5],[8],[10],[14],[15]).

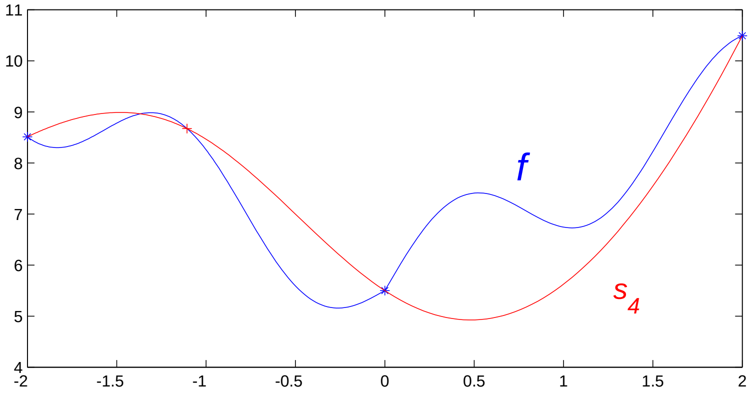


Figure 1: Expensive function f evaluated at four points and interpolating model s_4.

Let $\phi : \mathbb{R}_+ \to \mathbb{R}$ be one of the following functions, where $w \in \mathbb{N}$:

- surface splines: $\phi(r) = \begin{cases} r^\kappa & \text{with odd } \kappa \in \mathbb{N}, \\ r^\kappa \log r & \text{with even } \kappa \in \mathbb{N}, \end{cases}$
- multiquadrics: $\phi(r) = (r^2 + w^2)^\kappa$ with $\kappa > 0$ and $\kappa \notin \mathbb{N}$,
- inverse multiquadrics: $\phi(r) = (r^2 + w^2)^\kappa$ with $\kappa < 0$,
- Gaussians: $\phi(r) = \exp(-wr^2)$.

These functions are called radial basis functions. It turns out that their properties with respect to the above mentioned measure of smoothness are extremely helpful for the optimization process. Figure 1 shows an example of an interpolating model constructed by radial basis functions.

Using the advantageous properties of these functions, some authors developed efficient algorithms (see [3],[12]) including radial basis functions, where in [3] the name 'RBF-method' was established. Radial basis functions were used for interpolating models (see [1],[2],[4],[9], [12],[13]) as well as for approximation models (see [6]).

Algorithms based on the original RBF-method proposed by Gutmann [3], even Gutmann's method itself, often contain several subproblems. They depend on the optimization of the 'measure of smoothness' or 'measure of bumpiness', whose global solutions are not necessary for the convergence of the whole process. Therefore these subproblems, also called auxiliary problems, are usually solved heuristically or locally ([6],[9],[12],[13]). As the solution of these auxiliary problems determine the choice of the next evaluation point of the expensive function, we are increasingly interested in a deterministic solution method for calculating the global optimum. The main drawback is that these subproblems in general have a huge number of local optima, which increases with the number of interpolation points. Nevertheless global solutions of these subproblems are desirable ([1],[3]) in order to improve the overall process.

2. Conclusions

We present a method to solve those subproblems considering the positive characteristics of some of the radial basis functions mentioned above. Our talk includes the presentation of lower bounds which can be used in a branch-and-bound algorithm. This algorithm converges to the glolbal optimum of the auxiliary problem and finally investigates a sophisticated choice of evaluation points for the expensive function.

References

[1] M. Björkman and K. Holmström. Global Optimization of Costly Nonconvex Functions Using Radial Basis Functions. *Optimization and Engineering*, 1(4):373–397, 2000.

[2] A. Cassioli and F. Schoen. Global optimization of expensive black box problems with a known lower bound. *Journal of Global Optimization*, 57:177–190, 2013.

[3] H.-M. Gutmann. A Radial Basis Function Method for Global Optimization. *Journal of Global Optimization*, 19(3):201–227, 2001.

[4] K. Holmström. An adaptive radial basis algorithm (ARBF) for expensive black-box global optimization. *Journal of Global Optimization*, 41(3):447–464, 2008.

[5] A. Iske. *Charakterisierung bedingt positiv definiter Funktionen für multivariate Interpolationsmethoden mit radialen Basisfunktionen*. PhD thesis, Georg-August-Universität zu Göttingen, 1994.

[6] S. Jakobsson, M. Patriksson, J. Rudholm, and A. Wojciechowski. A method for simulation based optimization using radial basis functions. *Optimization and Engineering*, 11(4):501–532, 2009.

[7] D.R. Jones. A Taxonomy of Global Optimization Methods. *Journal of Global Optimization*, 21(4):345–383, 2001.

[8] C.A. Micchelli. Interpolation of Scattered Data: Distance Matrices and Conditionally Positive Definite Functions. *Constructive Approximation*, 2:11–22, 1986.

[9] J. Müller, C.A. Shoemaker, and R. Piché. SO-MI: A Surrogate Model Algorithm for Computationally Expensive Nonlinear Mixed-Integer Black-Box Global Optimization Problems. *Computers and Operations Research*, 40(5):1383–1400, 2013.

[10] M.J.D. Powell. A new iterative algorithm for thin plate spline interpolation in two dimensions. *Annals of Numerical Mathematics*, 4:519–527, 1997.

[11] M.J.D. Powell. Recent research at Cambridge on radial basis functions. *International Series of Numerical Mathematics*, 132:215–232, 1999.

[12] R.G. Regis and C.A. Shoemaker. Constrained Global Optimization of Expensive Black Box Functions Using Radial Basis Functions. *Journal of Global Optimization*, 31(1):153–171, 2005.

[13] R.G. Regis and C.A. Shoemaker. Improved strategies for radial basis function methods for global optimization. *Journal of Global Optimization*, 37(1):113–135, 2007.

[14] R. Schaback. Native Hilbert Spaces for Radial Basis Functions I. *International Series of Numerical Mathematics*, 132:255–282, 1999.

[15] M. Weinrich. *Charakterisierung von Funktionenräumen bei der Interpolation mit radialen Basisfunktionen*. PhD thesis, Georg-August-Universität zu Göttingen, 1994.

The hardness of the pooling problem

Dag Haugland[1]

[1]*Department of Informatics, University of Bergen, Bergen, Norway,* dag.haugland@ii.uib.no

Abstract The pooling problem is an extension of the minimum cost flow problem defined on a tripartite graph, where quality constraints are introduced at each terminal node. Flow entering the network at the source nodes has a given quality, at the internal nodes (pools) the entering flow is blended, and then sent to the terminal nodes where all entering flow streams are blended again. The resulting flow quality at the terminals has to satisfy given bounds. The objective is to find a cost-minimizing flow assignment that satisfies network capacities and the terminals' quality specifications.

Recently, it was proved that the pooling problem is NP-hard, and that the hardness persists when the network has a unique pool. In contrast, instances with only one source or only one terminal can be formulated as compact linear programs, and thereby solved in polynomial time. In this work, we study several important network classes, for which we prove that the pooling problem remains NP-hard. We also give examples of special cases in which the problem is solvable by linear programming. Finally, we point out some open problems that need to be addressed in order to identify more closely the borderline between polynomially solvable and NP-hard variants of the pooling problem.

Keywords: Pooling problem, NP-hard, Linear Programming

1. Introduction and definitions

The *pooling problem* is a network flow problem exhibiting linear relations that often confuse practitioners to believe that it can be formulated as a linear program. Years of experience with applications in e.g. oil refining, gas transportation and food production, have however established a consensus that linear formulations are achievable only in very special cases. The purpose of the current work is to identify conditions under which fast solution methods based on linear programming exist, and to identify conditions that render the problem intractable. To that end, we give theorems stating the NP-hardness of several special cases of the problem.

We consider the pooling problem as an extension of the minimum cost flow problem defined on a tripartite graph, where *quality constraints* are introduced at each terminal node. Flow entering the network has a given quality, which depends on the source node at which the flow enters. At all internal nodes, referred to as *pools*, the flow received from the sources is blended such that the resulting quality becomes a weighted average of the source qualities. Likewise, the flow sent from pools to terminals is blended at the terminals, where bounds on the resulting quality are specified. The objective is to find a cost-minimizing flow assignment that satisfies network capacities and the terminals' quality specifications.

More formally, the problem is introduced as follows. Let $D = (N, A)$ be a directed graph, where the node set $N = S \cup P \cup T$ consists of the sources S, pools P and terminals T, and the arc set $A \subseteq (S \times P) \cup (P \times T)$ links sources with pools and pools with terminals. We let K be a finite set, and refer to its elements as quality parameters. The unit cost of arc $(i, j) \in A$ and the capacity of node $i \in N$ are denoted c_{ij} and b_i, respectively. Further, we let q_s^k denote the quality of parameter $k \in K$ of the flow entering source s, and we let u_t^k denote the upper bound on the quality (small values indicate good quality) of the flow arriving at terminal t.

Introducing lower quality bounds does not lead to an extension of our problem. For all parameters $k \in K$ subject to lower bounds ℓ_t^k, we extend K by a new parameter k', let $q_s^{k'} = -q_s^k$ for all $s \in S$, and define the upper bounds $u_t^{k'} = -\ell_t^k$ for all $t \in T$. We therefore refer to instances with lower quality bounds as pooling problem instances, but whenever counting of the parameters is an issue, we count twice all parameters subject also to lower bounds. We have assumed that there are no direct arcs from sources to terminals. This assumption can be made without loss of generality, since any such arc can be replaced by a new pool along with two arcs connecting the pool to the source and the terminal, respectively.

To simplify writing, we introduce the neighbor sets $S_p = \{s \in S : (s,p) \in A\}$ and $T_p = \{t \in T : (p,t) \in A\}$ for each $p \in P$, along with $P_s = \{p \in P : s \in S_p\}$ and $P_t = \{p \in P : t \in T_p\}$.

2. Bilinear programming formulations

Introducing the variables x_{ij} and w_p^k, denoting the flow on arc $(i,j) \in A$ and the quality of parameter k at pool p, respectively, the problem is formulated as a bilinear program:

Problem 1.

$$\begin{array}{llll}
\min\limits_{x,w} & \sum_{(i,j)\in A} c_{ij}x_{ij} & & (1)\\
& \sum_{p\in P_s} x_{sp} \le b_s & s \in S & (2)\\
& \sum_{s\in S_p} x_{sp} \le b_p & p \in P & (3)\\
& \sum_{p\in P_t} x_{pt} \le b_t & t \in t & (4)\\
& \sum_{s\in S_p} x_{sp} - \sum_{t\in T_p} x_{pt} = 0 & p \in P & (5)\\
& \sum_{s\in S_p} q_s^k x_{sp} - w_p^k \sum_{t\in T_p} x_{pt} = 0 & p \in P, k \in K & (6)\\
& \sum_{p\in P_t} w_p^k x_{pt} - u_t^k \sum_{p\in P_t} x_{pt} \le 0 & t \in T, k \in K & (7)\\
& x \in \mathbb{R}_+^A & & (8)
\end{array}$$

Constraints (2)–(4) reflect the node capacities, (5) ensure flow conservation at pools, (6) imply that the quality variables w_p^k attain the correct value $\frac{\sum_{s\in S_p} q_s^k x_{sp}}{\sum_{t\in T_p} x_{pt}}$ if pool p receives non-zero flow, which in combination with (7) ensure that the quality $\frac{\sum_{p\in P_t} w_p^k x_{pt}}{\sum_{p\in P_t} x_{pt}}$ of parameter k at terminal t is within its bound u_t^k.

Haverly [8] was the first to formulate the problem in terms of quality variables. Later, formulations where quality variables are replaced by proportion variables have been suggested [1, 4, 9]. A proportion variable represents the fraction of the flow through a pool that originates from a given source, or the fraction destined for a given terminal. Despite the linearity of the blending operation at pools and terminals, all formulations mentioned here are bilinear.

In general, compact linear programs for the pooling problem exist only if $\mathcal{P} = \mathcal{NP}$ [1]. The goal of the current work is to provide hardness theorems identifying special cases of the problem that are provably NP-hard. We also point out some sufficient conditions under which the problem is solvable in polynomial time, and we give some cases for which neither hardness results nor polynomial time algorithms are known to date.

3. Exact solution methods

Exact algorithms with provable convergence to a global optimum were first given by Floudas and Visweswaran [6, 10], who developed an approach based on Lagrangian relaxation. Later, alternative Lagrangian-based techniques have been suggested [2, 3, 4]. Bilinear formulations have been the basis of methods exploiting relaxations by convex and concave envelopes [1, 7, 9]. When using the strongest available formulations, such exact algorithms are able to solve instances of moderate size quickly, but in line with what the complexity results predict, large scale instances seem to be unsolvable.

4. Hardness theorems

This section consists of theoretical results that will be proved in the full version of the paper.

4.1 Instances with a single source, pool or terminal

Proposition 1. *The pooling problem with $|P| = 1$ is NP-hard.*

The proof [1] is by a reduction from the independent vertex set problem, which for a graph $G = (V, E)$ asks for a vertex set $V' \subseteq V$ of maximum cardinality such that all pairs of vertices in V' are non-neighbors in G. In the corresponding pooling problem instance, all sets S, T and K have one element for each vertex $v \in V$, such that there is a one-to-one correspondence between sources and terminals, and between sources and quality parameters. Further, each edge in E defines a pair (s, t) such that no flow received at t should come from s. The unique pool is connected to each source and to each terminal. The quality vectors at the sources are distinct unit vectors, such that the quality of the flow leaving the pool tells from what sources the flow originates. At each terminal t, the quality bounds dictate that some flow must originate from its corresponding source, whereas no flow can originate from any source s where (s, t) corresponds to an edge in E.

Proposition 2. *For all $t_{\max} \geq 1$, the pooling problem with $|P| = 1$ and $|T| \leq t_{\max}$ is solvable in polynomial time.*

The proof is simply to observe that if the set of terminals to receive non-zero flow is known, the pooling problem with $|P| = 1$ is a compact linear program. Thus, the problem is solved by applying a polynomial LP-algorithm no more than $2^{t_{\max}}$ times. A similar argument [1] is used to prove that bounding the number of quality parameters has the same effect when $|P| = 1$.

Proposition 3. *For all $k_{\max} \geq 1$, the pooling problem with $|P| = 1$ and $|K| \leq k_{\max}$ is solvable in polynomial time.*

Open problem 1. *Does there exist some integer $s_{\max} \geq 2$, such that the pooling problem with $|P| = 1$ and $|S| \leq s_{\max}$ is solvable in polynomial time?*

Although the NP-hardness persists when there is only one pool, confining the digraph to have only one source or only one terminal has a strong effect. Slightly more general cases can be formulated as linear programs:

Proposition 4. *The pooling problem with $\min\{|S_p|, |T_p|\} = 1$ for all $p \in P$ is solvable in polynomial time.*

4.2 Instances with a single quality parameter

In this section, we answer positively the question, addressed by Dey and Gupte [5], whether the pooling problem remains NP-hard when $|K| = 1$. It turns out that restricting the number of quality parameters alleviates the computational challenge even less than allowing for only one pool. By Proposition 3, the problem can be solved quickly when $|P| = 1$ and $|K| = 2$. When the roles of P and K are swapped, the picture looks different.

Proposition 5. *The pooling problem with $|K| = 1$ and either $|P| = 2$ or $|S| = |T| = 2$ is NP-hard.*

It follows from Proposition 5 that constraining the number of neighbors of the sources or the terminals to any number above 1 leaves us with an NP-hard problem. Likewise, it follows from the same proposition that the problem remains NP-hard for $|K| = 1$ even if the in- and out-degrees of all pools are at most 2.

4.3 Instances with bounded node degrees

While Proposition 5 covers instances with small node degrees at the pools, we can also prove that the problem remains NP-hard even if there is no node with more than two entering arcs. An analogous result holds for the out-degrees.

Proposition 6. *The pooling problem with $|S_p| \leq 2$ for all $p \in P$ and $|P_t| \leq 2$ for all $t \in T$ is NP-hard.*

Proposition 7. *The pooling problem with $|T_p| \leq 2$ for all $p \in P$ and $|P_s| \leq 2$ for all $s \in S$ is NP-hard.*

Propositions 6–7 answer questions left open in [5]. Observe, however, that the propositions consider the number of quality parameters as arbitrary.

Open problem 2. *Does there exist an integer $k_{\max} \geq 1$ such that the pooling problem with $|S_p| \leq 2$ for all $p \in P$ and $|P_t| \leq 2$ for all $t \in T$ is solvable in polynomial time when $|K| \leq k_{\max}$?*

Open problem 3. *Does there exist an integer $k_{\max} \geq 1$ such that the pooling problem with $|T_p| \leq 2$ for all $p \in P$ and $|P_s| \leq 2$ for all $s \in S$ is solvable in polynomial time when $|K| \leq k_{\max}$?*

5. Summary

We have given instance classes for which the pooling problem is NP-hard. Notable among these are the networks with only one pool and the instances with only one quality parameter. In contrast, the problem can be solved in polynomial time if each pool is connected to only one source or only one terminal.

References

[1] Alfaki, M. and Haugland, D. (2013). Strong formulations for the pooling problem. *Journal of Global Optimization*, **56**(3), 897–916. doi 10.1007/s10898-012-9875-6.

[2] Almutairi, H., Elhedhli, S. (2009). A new Lagrangian approach to the pooling problem. *Journal of Global Optimization*. **45**(2), 237–257.

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

[4] Ben-Tal, A., Eiger, G., Gershovitz, V. (1994). Global minimization by reducing the duality gap. *Mathematical Programming*. **63**(1–3), 193–212.

[5] Dey, S.S., Gupte, A. (2014). Analysis of MILP techniques for the pooling problem. Optimization online.

[6] Floudas, C.A., Visweswaran, V. (1990). A global optimization algorithm (GOP) for certain classes of nonconvex NLPs–I. Theory. *Computers and Chemical Engineering*. **14**(12), 1397–1417.

[7] Foulds, L.R., Haugland, D., Jörnsten, K. (1992). A bilinear approach to the pooling problem. *Optimization*. **24**(1), 165–180.

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

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

[10] Visweswaran, V., Floudas, C.A. (1990). A Global Optimization Algorithm (GOP) for certain classes of nonconvex NLPs–II. Application of theory and test problems. *Computers and Chemical Engineering*. **14**(12), 1419–1434.

Global minimization using space-filling curves*

Daniela Lera[1] and Yaroslav Sergeyev[2]

[1] *Dipartimento di Matematica e Informatica, University of Cagliari, Italy* lera@unica.it

[2] *Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica, University of Calabria, Italy*
Software Department, University of Nizhni Novgorod, Gagarin Av. 23, Nizhni Novgorod, Russia yaro@si.dimes.unical.it

Abstract In this paper the global optimization problem of a multiextremal function satisfying the Lipschitz condition over a hyperinterval is considered. To solve it we propose algorithms that use Peano-type space-filling curves for reduction of dimensionality. The knowledge of the Lipschitz constant is not required. Local tuning on the behavior of the objective function and a new technique, named *local improvement*, are used in order to accelerate the search. Convergence condition are given. Numerical experiments show quite promising performance of the new technique.

Keywords: Global Optimization, space-filling curves approximations, set of Lipschitz constants

1. Introduction

Let us consider the following global optimization problem:

$$\min\{F(y) :\ y \in [a,b]\}, \tag{1}$$

where $[a,b]$ is a hyperinterval in R^N and F is a multiextremal function that satisfies the Lipschitz condition

$$|F(y') - F(y'')| \leq L\|y' - y''\|, \qquad y', y'' \in [a,b], \tag{2}$$

with a constant L, $0 < L < \infty$, generally unknown. In the literature, there exist numerous methods for solving the problems (1), (2), see, for example, [1], [5] [6], [11], [12], [13], [17], [18], [22], [23], [24]. In this paper, we consider an approach that uses numerical approximations of space-filling curves to reduce the original Lipschitz multidimensional problem to a univariate one satisfying the Hölder condition [10, 19]. These curves, first introduced by Peano (1890) [14], fill in the hyperinterval $[a,b] \subset R^N$, i.e., they pass through every point of $[a,b]$. It has been shown by Strongin (see [20, 21]) that, by using space filling curves, the multidimensional global minimization problem (1), (2) is turned into a one-dimensional problem. In particular, Strongin has proved that finding the global minimum of the Lipschitz function $F(y), y \in R^N$, over a hyperinterval is equivalent to determining the global minimum of the function $f(x)$:

$$f(x) = F(p(x)), \qquad x \in [0,1], \tag{3}$$

where $p(x)$ is the Peano curve. Moreover, the Hölder condition

$$|f(x') - f(x'')| \leq H|x' - x''|^{1/N}, \qquad x', x'' \in [0,1], \tag{4}$$

holds for the function f with the constant

$$H = 2L\sqrt{N+3}, \tag{5}$$

*This research was partially supported by the INdAM–GNCS 2014 Research Project of the Italian National Group for Scientific Computation of the National Institute for Advanced Mathematics "F. Severi".

where L is the Lipschitz constant of the multidimensional function $F(y)$. Thus, we can solve the problem (1), (2) by using algorithms proposed for minimizing functions in one dimension but it is required to use Hölder metric. Naturally, in order to realize the passage from the multi-dimensional problem to the one-dimensional one, computable approximations to the Peano curve should be employed in the numerical algorithms. In Fig. 1 we can see an approximation of level 5 of the curve in dimension $N = 2$. It can be seen that the objective function has been evaluated at points on the curve.

2. Geometric Methods

To solve the one-dimensional problem (3), (4), we propose several algorithms that use a geometrical approach [7, 19]. Using the Lipschitz condition these methods, at each iteration, construct an auxiliary function that is a minorant for the objective function and select a point at which the objective function is evaluated (trial point). At a generic iteration k, the search interval has been already subdivided into k subintervals and k trials have been executed at points $x_1, ..., x_k$. Then, for each subinterval $[x_{i-1}, x_i]$, a so-called characteristic M_i is defined (this can be done in various ways, e.g., in certain cases M_i is defined as the minimum of the auxiliary function in the considered subinterval), and the next trial is executed at the point x_{k+1} in that interval corresponding to the minimal value of the characteristic.

In Fig. 1 an example of convergence of the sequence of trial points generated by an algorithm called MGAS [9] that uses the geometric approach in dimension $N = 2$ using the approximation of the level $M = 5$ to the Peano curve is given. In this example a function test generated by the GKLS-generator from [4] has been considered. We can see, in black, the Peano curve in the domain $[-1, 1]$; the zone with the high density of the trial points corresponds to the global minimizer.

One of the main issues regarding these methods is related to the treatment of the Lipschitz constant L. We propose different techniques for acquiring the Lipschitz information that can be distinguished with respect to the way the Lipschitz constant is estimated during the process of optimization. In particular, in order to solve the problem (1), (2), we consider techniques that use either a global estimate of the Lipschitz constant valid for the whole search region $[a, b]$, or local estimates L_i valid only for some subregions $[a_i, b_i] \subseteq [a, b]$. Naturally, a balancing between the local and global information must be performed in an appropriate way in order to avoid the missing of the global solution. Moreover, we propose a new approach, called "local improvement" [7, 8, 15] in oder to accelerate the convergence of the methods. This new technique forces the global optimization method to make a local improvement of the best approximation of the global minimum immediately after a new approximation better than the current one was found.

We performed several series of numerical experiments, involving more than 800 test functions. In particular, we compared our basic algorithm, named AG (see [7]), with the original Direct algorithm proposed by Jones, Perttunen, and Stuckman (see [6]) and its recent locally biased modification LBDirect introduced by Gablonsky and Kelley (see [2, 3]). Fig. 2-left shows the behavior of the three methods on a classe of 100 test functions generated by the GKLS-generator (see [4] for a detailed explanation, examples of its usage) in dimension $N = 4$: it can be seen that after 10000 function evaluations the LBDirect has found the solution at 58 problems, Direct at 73 problems and the AG at 93 problems). Fig. 2-right illustrates the results of the experiment on a class of 100 test functions in dimension $N = 2$, by considering the basic method AG with a method, named ALI (see [7]), in which we have used the local improvement technique in order to accelerate the search. Note that after 500 iterations the stopping rule in the ALI was verified for 84 functions and all the minima have been found, whereas the algorithm AG stopped only at 2 functions.

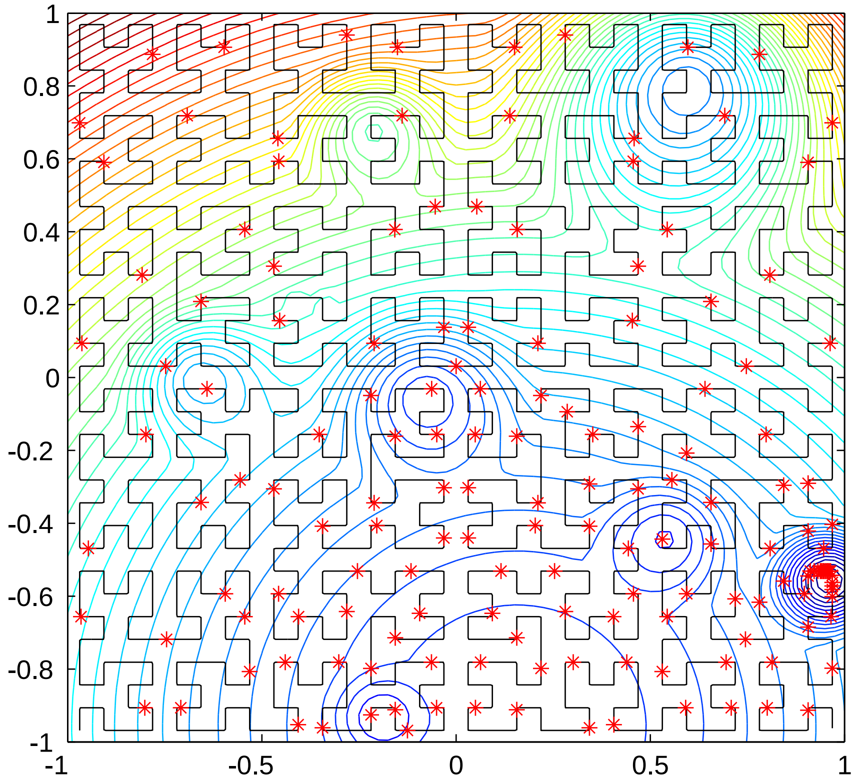


Figure 1: Trial points produced by the MGAS [9] and Peano curve approximation of level 5 (in black) while optimizing a function generated by the GKLS-generator [4].

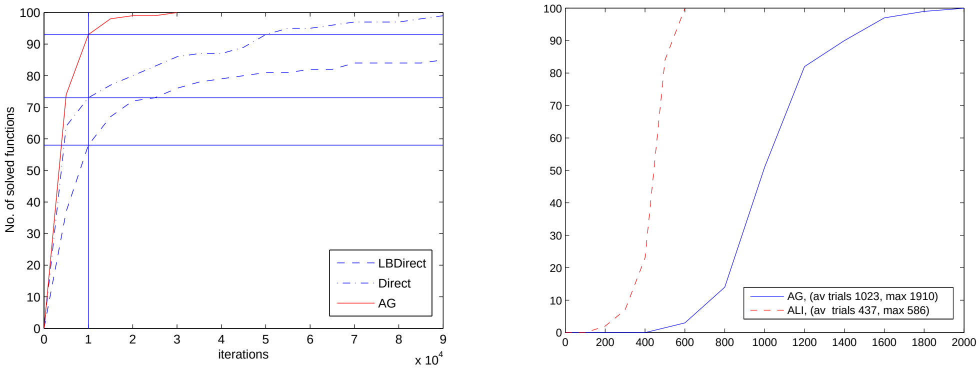


Figure 2: *Methods AG [7], Direct and LBDirect, N=4, left; methods AG and ALI [7], N=2, right.*

References

[1] Evtushenko, Yu. G. (1971). “Numerical methods for finding global extrema of a nonuniform mesh”. USSR Comput. Math. Math. Physics Vol. 11, 1390–1403.

[2] Gablonsky M.J. (2001). “Modifications of the DIRECT Algorithm”. Ph.D thesis, North Carolina State University, Raleigh, NC.

[3] Gablonsky M.J. and Kelley C.T. (2001). “A locally-biased form of the DIRECT Algorithm”. Journal of Global Optimization, Vol. 21, 27–37.

[4] M. Gaviano, D. E. Kvasov, D. Lera, and Ya. D. Sergeyev (2003). “Algorithm 829: Software for generation of classes of test functions with known local and global minima for global optimization”, ACM Transactions on Mathematical Software, Vol. 29(4), 469–480.

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

[6] Jones, D.R., Perttunen, C.D., Stuckman, B.E. (1993). “Lipschitzian optimization without the Lipschitz constant”. Journal of Optimization Theory and Applications Vol. 79, 157–181.

[7] Lera D., Sergeyev Ya.D. (2010). “Lipschitz and Hölder global optimization using space-filling curves”. Applied Numerical Mathematics, Vol. 60(1-2), 115–129.

[8] Lera D., Sergeyev Ya.D. (2010). “An information global minimization algorithm using the local improvement technique”. Journal of Global Optimization, Vol. 48(1), 99–112.

[9] Lera D., Sergeyev Ya.D. (2014). “Deterministic global optimization using space-filling curves and multiple estimates of Lipschitz and Hölder constants”. Submitted.

[10] Lera D., Sergeyev, Ya.D. (2013). “Acceleration of univariate global optimization algorithms working with Lipschitz functions and Lipschitz first derivatives”. SIAM J. Optim. Vol. 23(1), pp. 508–529.

[11] R. Paulavičius, J. Zilinskas (2014). *Simplicial Global Optimization*. Springer, New York.

[12] Paulavičius R, Zilinskas J, Grothey A (2010). “Investigation of selection strategies in branch and bound algorithm with simplicial partitions and combination of Lipschitz bounds”. Optimization Letters, Vol. 4, 173–183.

[13] Pintér, J (1996). *Global Optimization in Action (Continuous and Lipschitz Optimization Algorithms, Implementations and Applications)*. Kluwer Academic Publishers, Dordrecht.

[14] Peano, G. (1890). “Sur une courbe, qui remplit toute une aire plane”. Math. Annalen Vol. 36, 157–160.

[15] Sergeyev Ya.D. (1995). “An information global optimization algorithm with local tuning”. SIAM Journal on Optimization, Vol. 5(4), 858–870.

[16] Sergeyev Ya.D., Pugliese P., Famularo D. (2003). “Index information algorithm with local tuning for solving multidimensional global optimization problems with multiextremal constraints”. Mathematical Programming, Vol. 96(3), 489–512.

[17] Sergeyev Ya.D., Kvasov, D.E. (2006). “Global search based on efficient diagonal partitions and a set of Lipschitz constants”. SIAM J. Optim. Vol. 16, pp. 910–937.

[18] Sergeyev Ya.D., Kvasov D.E. (2008). *Diagonal global optimization methods*. Fizmatlit, Moscov, (in Russian).

[19] Sergeyev Ya.D, Strongin R.G., Lera D. (2013). *Introduction to Global Optimization Exploiting Space-Filling Curves*. Springer, New York.

[20] Strongin, R.G. (1978). *Numerical Methods in Multiextremal Problems*. Nauka, Moscow. (In Russian).

[21] Strongin, R.G. (1992). “Algorithms for multi-extremal mathematical programming problems employing the set of joint space-filling curves”. J. Glob. Optim. Vol. 2, pp. 357–378.

[22] Strongin, R.G., Sergeyev, Ya.D. (2000). *Global Optimization with Non-convex Constraints: Sequential and Parallel Algorithms*. Kluwer Academic Publishers, Dordrecht.

[23] Strongin R.G., Sergeyev Ya.D. (2003). “Global optimization: fractal approach and non-redundant parallelism”. Journal of Global Optimization, Vol. 27(1), 25–50.

[24] A. A. Zhigljavsky and A. Zilinskas (2008). *Stochastic Global Optimization*. Springer, New York.

A Variable Neighborhood Search Matheuristic for the Heterogeneous P-Median Problem*

Éverton Santi[1], Daniel Aloise[1], and Simon J. Blanchard[2]

[1] *Universidade Federal do Rio Grande do Norte, Natal-RN, Brazil, 59078-970,*
santi.everton@gmail.com, aloise@dca.ufrn.br

[2] *McDonough School of Business, Georgetown University, Washington, DC 20057, USA,*
sjb247@georgetown.edu

Abstract

The p-median problem is a model that partitions n objects into p clusters by minimizing the sum of distances from each point to the center of its cluster using, as input, a single dissimilarity matrix between the objects. This involves simultaneously selecting the p cluster centers (necessarily selected among the n objects) and assigning objects to clusters. In some settings, multiple dissimilarity matrices are available. Given that the p-median model uses only one matrix as input, researchers typically aggregate their data into a single matrix resulting in p-median results that mask the true nature of the data. The Heterogeneous P-Median Problem was recently proposed as an alternative model to the classical p-median model for clustering. It consists of a three-way partitioning problem that identifies groups of individuals with similar category structures. In this work, we present a Variable Neighborhood Search heuristic for the problem based on the exact exploration of large neighborhoods modeled as mixed-integer programs. Preliminary computational experiments show that the heuristic is very efficient for a set of synthetic instances.

Keywords: p-median, matheuristic, variable neighborhood search

1. Introduction

Let n individuals sort m objects such that a matrix data $D^i = (d^i_{jk})$ is obtained for $i = 1, \dots, n$, representing the dissimilarities between pairs of objects j and k as perceived by individual i, and c^i, for $i = 1, \dots, n$ as the number of categories individual i wants to classify the objects in the dataset. Then, the Heterogeneous P-median Problem (HPM) proposed by [6] can be formulated as:

$$\text{Minimize} \sum_{i=1}^{n}\sum_{g=1}^{G} p^{ig}\left[\sum_{j=1}^{m-1}\sum_{k=j+1}^{m} d^i_{jk} e^g_{jk}\right] \tag{1}$$

subject to

$$\sum_{k=1}^{m} e^g_{jk} = 1, \forall g = 1, \dots, G, \forall j = 1, \dots, m \tag{2}$$

$$\sum_{g=1}^{G} p^{ig} = 1, \forall i = 1, \dots, n \tag{3}$$

$$\sum_{j=1}^{m} e^g_{jj} \leq \frac{\sum_{i=1}^{n} c^i p^{ig}}{\sum_{i=1}^{n} p^{ig}}, \forall g = 1, \dots, G \tag{4}$$

*This work has been partially supported by CNPq Brazilian agency

$$e^g_{jk} \leq e^g_{kk}, \forall g = 1, \ldots, G, \forall j, k = 1, \ldots, m, j \neq k \tag{5}$$

$$e^g_{jk} \in \{0, 1\}, \forall g = 1, \ldots, G, \forall j, k = 1, \ldots, m, j \neq k \tag{6}$$

$$p^{ig} \in \{0, 1\}, \forall g = 1, \ldots, G, \forall i = 1, \ldots, n \tag{7}$$

The n individuals must be partitioned into G groups. The decision variables p^{ig} express the assignment of individual i to the group g. Variables e^g_{jk} are equal to 1 if object j is linked to object k in group g, 0 otherwise. The objective is to minimize (1), i.e., the sum of dissimilarities between each object and its assigned category, conditional on (individual) group membership. Constraints (2) impose that each object j must be assigned to exactly one median in each group g. Constraints (3) ensure that each individual is assigned to exactly one group. Constraints (4) ensure that the total number of medians for each group g must be smaller or equal to the average number of medians expected by the individuals in that group. The optimization process guarantee that $\sum_{j=1}^{J} e^g_{jj}$ is as big as possible since more medians in a group imply lower values in the objective function.

The problem is nonconvex due to the objective function (1) and constraints (4). Besides, the HPM is already NP-hard for $G = 1$ since, in this case, it is equivalent to the p-median problem [4].

2. Variable Neighborhood Search algorithm for the HPM

Variable Neighborhood Search (VNS) is a metaheuristic developed to solve combinatorial and global optimization problems by changing neighborhoods in its local descent step for intensification as well as in its shaking step for diversification (see [3] for a survey).

An initial solution is needed in the VNS framework. Our constructive algorithm first solves the problem of partitioning, with the p-median model, the dissimilarity matrices made by the individuals according to the Frobenius norm. Then, individual assignments to groups are performed following the partition obtained, so that if a pair of individuals have their dissimilarity matrices assigned to the same cluster in the partition then these individuals are assigned to the same group in the solution of the HPM. Finally, a complete solution to HPM, including objects assignments variables, is obtained through the solution of subproblems M_g, for $g = 1, \ldots, G$, given by:

$$\text{Minimize } M_g = \sum_{j=1}^{m-1} \sum_{k=j+1}^{m} \overline{d}^g_{jk} e^g_{jk} \tag{8}$$

subject to

$$\sum_{k=1}^{m} e^g_{jk} = 1, \forall g = 1, \ldots, G, \forall j = 1, \ldots, m \tag{9}$$

$$\sum_{j=1}^{J} e^g_{jj} = \lfloor \Omega_g \rfloor, \tag{10}$$

$$e^g_{jk} \leq e^g_{kk}, \forall g = 1, \ldots, G, \forall j, k = 1, \ldots, m, j \neq k \tag{11}$$

$$e^g_{jk} \in \{0, 1\}, \forall g = 1, \ldots, G, \forall j, k = 1, \ldots, m, j \neq k \tag{12}$$

where $\overline{d}^g_{jk} = \sum_{i=1}^{I} d^i_{jk} p_{ig}$, and $\Omega_g = \frac{\sum_{i=1}^{I} c_i p_{ig}}{\sum_{i=1}^{I} p_{ig}}$. Remark that problem (8)-(12) corresponds to the p-median problem.

The shaking component is implemented by means of random moves in the *swap* neighborhood, which encompasses all the moves of removing a individual from a group and adding it to another one. Thus, if the second neighborhood is used for shaking, then two random swap moves are performed for two individuals; if the third one is used, then three swap moves are performed, and so on.

Our local search is devised following the Variable Neighborhood Descent (VND) framework, which generalizes the VNS principles to descent methods. The first explored neighborhood $\mathcal{N}_1$ solves problem (8)-(12) for each group affected by the shaking procedure, thereby trying to find the best medians given the current individuals assignments.

The second neighborhood structure $\mathcal{N}_2$ considers the medians and objects assignments as fixed, i.e. variables e, and looks for a better solution by optimizing individuals assignments variables p over all groups. For that purpose, it solves the following binary program:

$$\text{Minimize}\, W = \sum_{i=1}^{n}\sum_{g=1}^{G} p_{ig}\tilde{d}_{jk}^{ig} \tag{13}$$

subject to

$$\frac{\sum_{i=1}^{n} c_i p_{ig}}{\sum_{i=1}^{n} p_{ig}} \geq \omega_g, \forall g = 1, \ldots, G \tag{14}$$

$$\sum_{g=1}^{G} p_{ig} = 1, \forall i = 1, \ldots, n \tag{15}$$

$$p_{ig} \in \{0, 1\}, \forall g = 1, \ldots, G, \forall i = 1, \ldots, n, \tag{16}$$

where $\tilde{d}_{jk}^{ig} = \sum_{j=1}^{m}\sum_{k=1}^{K} e_{jk}^{g} d_{jk}^{i}$, and $\omega_g = \sum_{j=1}^{m} e_{jj}^{g}$.

Remark that the first two neighborhoods structures do not allow to modify the number of medians in a group. With that in mind, a third neighborhood structure $\mathcal{N}_3$ was conceived. This neighborhood explores neighbors for which the number of medians in a group is augmented in one unity. Its exploration considers each group in turn. For a fixed p, the solution becomes infeasible when the number of medians of a group g^* is augmented. In spite of that, in its first exploration phase, $\mathcal{N}_3$ solves the p-median subproblem M_{g^*} with Ω_{g^*} replaced by $\Omega_{g^*} + 1$. Then, in a second phase, neighborhood exploration proceeds by solving problem (13)-(16) with ω_{g^*} replaced by $\omega_{g^*} + 1$. If the latter is infeasible, it means that we are not able to support the augmentation in the number of medians of group g^* through the reassignments of individuals.

The three neighborhood structures presented in this section are considered as large neighborhoods in the sense of Ahuja et al. [1]. They are each explored within the VND local descent by solving mixed-integer programs. Therefore, the proposed VNS method for the HPM problem can be seen as a matheuristic [5].

3. Preliminary results

We executed our VNS algorithm in a set of synthetic instances from [2]. The tests were performed in a AMD Phenom II with a 800 MHz clock and 8 Gb of RAM memory. Exact solutions were obtained through convexification of model (1)-(7) (cf. [6]) solved by CPLEX 12.2. Table 1 presents the computational results obtained by CPLEX and VNS. CPLEX was executed until optimality was attained while the VNS heuristic was allowed to run for one minute of CPU time. The first column contains the instance id. The second, third and fourth columns show for each instance the number of individuals n, objects m and groups G, respectively. The fifth column reports the optimal solution obtained by CPLEX whereas the CPU time (in seconds) needed to obtain it is shown in the sixth column. Results regarding the constructive heuristic, denoted CH, are presented in the seventh column. Finally, VNS results are presented in the eighth column.

We note from Table 1 that the proposed VNS finds the optimal solution values for all tested instances in less than one minute of CPU time.

Table 1: Computational results obtained by CPLEX, CH and VNS for a set of instances from [2]

Id	n	m	G	CPLEX	Time(s)	CH	VNS
1	300	18	2	2399.94	2183.57	2407.34	2399.94
2	150	18	2	1874.98	193.33	1877.65	1874.98
3	300	18	2	2399.94	3472.22	2405.34	2399.94
4	300	30	2	5760.00	24238.99	5766.20	5760.00
5	150	18	6	1199.90	56562.94	1270.00	1199.90
6	150	30	2	3645.00	25345.88	3648.00	3645.00
7	150	18	6	1199.90	18220.32	1453.00	1199.90
8	150	18	2	1874.98	247.84	1878.32	1874.98

4. Summary

In this abstract, we described the heterogeneous p-median problem which is a nonconvex model that can be used to cluster heterogeneous data collected from different individuals. In the sequel, we sketched a VNS matheuristic to the problem based on the exploration of large neighborhoods formulated as mixed-integer problems. In the talk, we will present the model and the method in more detail besides presenting further experiments for real-data applications.

References

[1] Ahuja, R. K., Ergun Ö, Orlin J.B. and Punnen, A.P. (2002). "A survey of very large-scale neighborhood search techniques". Discrete Applied Mathematics, v. 123, 75–102.

[2] Blanchard, S.J., Aloise, D. and DeSarbo W.S. (2012). "The heterogeneous p-median problem for categorization based clustering". Psychometrika, v. 77, 741–762.

[3] Hansen, P., Mladenović, N. and Pérez J.A.M. (2008). "Variable neighborhood search: methods and applications". 4OR, v. 6 319–360.

[4] Kariv, O. and Hakimi, S.L. (1979). "An algorithmic approach to location problems, part ii: p-medians". Journal of Applied Mathemathics, v. 37 539–560.

[5] Maniezzo, V. Stüetzle T. and Voss T. (2009). "Hybridizing Metaheuristics and Mathematical Programming". Series: Annals of Information Systems, v. 10, Springer.

[6] Santi, É., Aloise, D. and Blanchard, S.J. (2014). "The heterogeneous p-median problem: Models and algorithms for analysis of sorting task data", submitted.

A Quadratic Branch and Bound with Alienor Method for Global Optimization

Aaid Djamel[1], Noui Amel[1], Zidna Ahmed[2], Ouanes Mohand[3], and Le Thi Hoai An[2]

[1]*Batna University, Batna, Algeria,* djamelaaid@gmail.com

[2]*Lorraine University, Metz, France,* ahmed.zidna@univ-lorraine.fr

[3]*Tizi-Ouzou University, Tizi-Ouzou, Algeria,* ouanesmohand@yahoo.fr

Abstract A new method for the underestimation of multivariate nonconvex functions is presented in this article. The method is based on a piecewise quadratic underestimator allowing to choose a better lower bound close to the value of the objective which accelerates the convergence of the two sequences: lower bounds and upper bounds to the optimum. which produces a set of convex piecewise underestimator. The resulting underestimators are very tight, the enormous advantages resides in the finest possible partitioning of the domain and also the problem of the lower bound uses local optimizers, since it has explicit solutions. The method was applied to a series of test functions presented previously in the literature and the results indicate that the method produces underestimators high quality in terms of tightening.

Keywords: Global optimization , Branch and Bound, Alienor's Method.

1. Introduction

Deterministic branch and bound methods for the solution of general nonlinear programs have become increasingly popular during the last decade or two, with increasing computer speed, algorithmic improvements, and multiprocessors. These methods are mostly based on the construction of a convex underestimating problem which allows the generation of two converging sequences of upper and lower bounds. The computation of a good convex lower bound function is very important in global optimization, since the tightness of the lower bound of nonconvex functions has a strong influence on the amount of computation. Constant and affine lower bound functions are extensively used in global optimization because of their simplicity and ease of computation.In [6] they introduced a new class of convex underestimators for twice continuously differentiable NLPs, studied their theoretical properties, and proved that the resulting convex relaxation is improved compared to the αBB one. Furthermore, they presented computational results of the new class of convex underestimators embedded in a branch-and-bound framework for box-constrained NLPs [7].They also proposed a hybrid global optimization method that includes the random-linkage stochastic approach with the aim at improving the computational performance. In [8] they proposed novel convex underestimators for trigonometric functions which are trigonometric functions themselves. The underestimation method can be applied to onedimensional as well as multi-dimensional problems involving trigonometric polynomials, since the product of trigonometric functions can always be decomposed into the sum of sin and cos functions with arguments that are linear combinations of the problem variables. In [9] they proposed two new classes of convex underestimators for general C2 NLPs which combine the αBB underestimators within a piecewise quadratic perturbation, derived properties for the smoothness of the convex underestimators, and showed the improvements over the classical αBB convex underestimators for box-constrained optimization problems. Sherali et al. [10] proposed a new cutting plane methodology that is based on the construction of a partial convex hull representation for a given $0-1$ mixedinteger programming problem by using the reformulation–linearization technique (RLT). The cuts are generated by projecting the extended space of the RLT formulation into the original space, and the authors investigated several variable selection rules for performing this convexification in a computationally efficient manner. [11] developed tight convex underestimators for univariate C2-continuous functions of arbitrary structure. These are based on a piecewise application of the αBB underestimators and it is theoretically proven that a finite number of pieces is sufficient for the method to yield the a priori unknown convex envelope of the function. The methodology was extended to handle multivariate functions [12], through appropriate projections of the function's epigraph into select one-dimensional spaces. Orthonormal transformations were also employed to improve the quality of the underestimation. Unfortunately, they use the local minimizers to determine the upper bound, while the authors proposed a quadratic underestimator whose minimum is explicitly calculated which can be used [5]:

$$q(s) = L_h f(s) - \frac{1}{2}Kh^2, \forall s \in S \tag{1}$$

to solve the problem of global optimization with simple constraints defined as follows:

$$(P) : \alpha = \min f(s), \ s \in S = [a, b] \tag{2}$$

we assume that f is twice differentiable on S on their second derivatives are bounded, ie there are positive numbers K

$|f''(s)| \leq K$ for all s $\in$ S. Such a K can be defined in several ways in practice: is it possible to know a priori values, or they are estimated in a manner to course the algorithm. In our approach, K are assumed to be known.

Let $\{s_1, s_2, ..., s_m\}$ be a uniform discretization with mesh size h of $S = [a, b]$ where $s_1 = a$ and $s_m = b$.Let$\{w_1, w_2, ..., w_m\}$ be a finite sequence of functions defined as ([1], [3])

$$\omega_i = \begin{cases} \frac{s-s_{i-1}}{s-s_{i-1}} & if \ s_{i-1} \leq \ s \leq s_i \\ \frac{s_{i+1}-s}{s_{i+1}-s_i} & if \ \ s_i \leq \ s \leq s_{i-1} \end{cases}, \tag{3}$$

where $s_0 = s_1$ and $s_{m+1} = s_m$. We have ([3], [5])

$$\sum_{i=1}^{m} \omega_i(s) = 1, \forall s \in S. \tag{4}$$

$$\omega_i(s_j) = \begin{cases} 0 \, if \ i \neq j \\ 1 \quad \text{otherwise} \end{cases} \tag{5}$$

Let $L_h f$ be the piecewise linear interpolant to f at points $s_1, ...s_m$ ([2],[3])

$$L_h f \ = \sum_{i=1}^{m} \omega_i(s) f(s_i). \tag{6}$$

Our contribution consists in providing a better lower bound in a short period of time, while preserving the advantage of the explicit solution.

2. Tightness of proposed underestimation

Let $f(s)$ be a univariate function that needs to be underestimated in $S = [a, b]$. We select an integer $N > 1$ and partition the complete domain in N segments of equal length. Thus, the i^{th} subdomain would be defined as $D_i = [s_i, s_{i+1}]; i = 0, ..., N-1$, where : $s_i = a + \left(\frac{b-a}{N}\right) i$, $i = 0, ..., N$. For every subdomain $D_i, i = 0, ..., N-1$, we construct the corresponding $PQBB$ (piecewise quadratic branch and bound) underestimator:

$$\left\{ \begin{array}{c} P_i(s) = K_i \frac{(s-s_{i+1})(s-s_i)}{2} + L_i(s), \\ with\ L_i(s) = \frac{s-s_{i+1}}{s_{i+1}-s_i} f(s_{i+1}) + \frac{s-s_{i+1}}{s_i - s_{i+1}} f(s_i) \\ such\ as\ K \geq K_i \geq |f''(s)| \end{array} \right. \tag{7}$$

where K_i is a upper bound of the second derivative that is valid for the entire subdomain D_i. Note that although an underestimator $P_i(s)$ can be defined outside its respective subdomain, its convexity is only guaranteed for $s \in [s_i, s_{i+1}]$.To calculate the upper bound K_i, we need of some definitions.

Theorem 1. *[Tightness of proposed underestimation] We define $P(s)$, $s \in [a,b]$ to be the following piecewise function:*

$$P(s) = P_i(s), \forall s_i \leq s \leq s_{i+1},\ i = 0, ..., N-1, \tag{8}$$

this function is a piecewise convex valid underestimator of $f(s)$ for all $s \in [a,b]$, and it is tighter than the quadratic underestimator $q(x)$ introduced in [4].

$$q(s) \leq p(s) \leq f(s),\ \ \forall s \in [a,b] \tag{9}$$

For all $s \in [s_i, s_{i+1}],\ i = 0, ..., N-1$, the lower bounds are computed explicitly

$$s_i^* = \left\{ \begin{array}{cc} \mu = \frac{1}{2}(s_i + s_{i+1}) - \frac{1}{K_i}(f(s_{i+1}) - f(s_i)) & if\ \mu \in [s_i, s_{i+1}] \\ s_i & if\ \mu \leq s_i \\ s_{i+1} & if\ \ \mu \geq s_{i+1} \end{array} \right. \tag{10}$$

Now, we compare the $P_i(s_i^*)$ for the best

$$LB_i^k = \min P_i(s_i^*),\ i = 0, ..., N-1 \tag{11}$$

The upper bound is calculated by the following comparisons and maitain the best ever.the objective function is evaluated quite different point which has to determine the upper bound:

$$UB^k = \min\{f(s_i^*),\ f(s_i), i = 0, ..., N-1\} \tag{12}$$

2.1 Branch and bound algorithm

```
**Initialization** $a, b, N, \varepsilon, k = 0$
Compute $UB_k = \min\{f(a), f(b), f(s)\}$
          $LB_k = q(s)$,
            $M = \{[a,b]\}$
While $(UB_k - LB_k > \varepsilon)$ Do
      $a \longleftarrow \quad x_{\bar{i}}$
      $b \longleftarrow \quad x_{\bar{i}+1}$
Compute $x_i = a + \left(\frac{b-a}{N}\right) i$,
for all $i = 0, ...., N$
          If $f(x_i) \leq UB_k$
             $UB_k \longleftarrow f(x_i)$
Compute $K_i$ and $S_i$ for all $i = 0, ...., N-1$
and   $\min\{\min f(s_i), UB_k\}$
      If $f(s_i) \leq UB_k$
        $UB_k \longleftarrow f(s_i)$
        $LB_i$
      If $UB_k - q_k(s_i) \geq \varepsilon$
        integrate $[x_i, x_{i+1}]$ in $M$
```

```
  for all $i = 0, ...., N - 1$
    If $UB_k - q_k(s_i) < \varepsilon$
        $[x_i, x_{i+1}]$ will remove from $M$
      $\min q_k(s_i)$
      $N = E\left(\frac{N}{2}\right) + 1$
    $k = k + 1$
$s^k$ is the optimal solution corresponding to the best $UB_k$
```

Theorem 2. *[Convergence of the algorithm]*
Either the algorithm is finite or it generates a bounded sequence$\{s_k\}$. Any accumulation point of the sequence is a global optimal solution of (P).We have $UBk \searrow \alpha$,$LBk \nearrow \alpha$.

3. Conclusion

Our contribution helps build lower bounds piecewise functions that give more information about the optimal solution and it also provides better initial solutions and lower bounds that make it the quickest and the most effective method. The Theoretical and algorithmic extensions of the method for application on multivariate functions is developed and without loss of the required benefits.

References

[1] Casado, L.G., Martínez, J.A., García, I. and Sergeyev, Ya.D. New Interval Analysis Support Functions Using Gradient Information in a Global Minimization Algorithm, Journal of Global Optimization, 2003.

[2] Ciarlet, P.G. The Finite Element Method for Elliptic Problems Studies in Math.and its Appl., 1979.

[3] De Boor, C. A Practical Guide to Splines, Applied Mathematical Sciences, Springer Verlag, 1978.

[4] Le Thi, H.A. and Pham Dinh Tao. A Branch-and-Bound method via D.C. Optimization Algorithm and Ellipsoidal technique for Box Constrained Nonconvex Quadratic Programming Problems, Journal of Global Optimization, 13 (1998), pp. 171-206.

[5] Le Thi, H.A. and Ouanes, M. Convex quadratic underestimation and Branch and Bound for univariate global optimization with one nonconvex constraint, RAIRO Operations Research 40(2006) 285-302.

[6] Akrotirianakis, I.G. and Floudas, C.A. A new class of improved convex underestimators for twice continuously differentiable constrained NLPs. J. Glob. Optim. 30(4), 367–390 (2004)

[7] Akrotirianakis, I.G. and Floudas, C.A. Computational experience with a new class of convex underestimators: box-constrained NLP problems. J. Glob. Optim. 29(3), 249–264 (2004)

[8] Caratzoulas, S. and Floudas, C.A. A trigonometric convex underestimator for the base functions in Fourierspace. J. Optim. Theory Appl. 124(2), 339–362 (2005)

[9] Meyer, C.A. and Floudas, C.A. Convex underestimation of twice continuously differentiable functions by piecewise quadratic perturbation: spline αBB underestimators. J. Glob. Optim. 32, 221–258 (2005)

[10] Sherali, H.D., Lee, Y. and Kim, Y. Partial convexification cuts for 0-1 mixed-integer programs. Eur. J. Oper.Res. 165(3), 625–648 (2005)

[11] Gounaris, C.E. and Floudas, C.A. Tight convex underestimators for C2-continuous problems: I. Univariate functions.J. Glob. Optim. 42(1), 51–67 (2008)

[12] Gounaris, C.E. and Floudas, C.A. Tight convex underestimators for C2-continuous problems: II. Multivariate functions. J. Glob. Optim. 42(1), 69–89 (2008)

Lipschitz Global Optimization with Derivatives*

Dmitri E. Kvasov[1,2] and Yaroslav D. Sergeyev[1,2]

[1] *DIMES, University of Calabria, Via P. Bucci, 42C – 87036, Rende (CS), Italy*

[2] *Software Department, N. I. Lobachevsky State University, Nizhni Novgorod, Russia*

{kvadim, yaro}@si.dimes.unical.it

Abstract In this work, a global optimization problem is considered where both the objective function $f(x)$ and its gradient $f'(x)$ are black-box multiextremal functions. It is supposed that $f'(x)$ satisfies the Lipschitz condition over the search hyperinterval with an unknown Lipschitz constant K. A number of geometric Lipschitz global optimization methods based on constructing auxiliary functions with the usage of different estimates of the Lipschitz constant K are presented in this communication. Results of their systematic experimental investigation are also given.

Keywords: Black-box global optimization, Lipschitz derivatives, Diagonal algorithms, Numerical comparison

1. Problem statement

A global optimization problem is considered where the objective function $f(x)$ is a multidimensional multiextremal and hard to evaluate function and its gradient $f'(x)$ satisfies the Lipschitz condition over a hyperinterval D with an unknown Lipschitz constant K, $0 < K < \infty$:

$$f^* = f(x^*) = \min_{x \in D} f(x), \quad D \subset \mathbb{R}^n, \tag{1}$$

$$D = [l, u] = \{x \in \mathbb{R}^n : l(j) \leq x(j) \leq u(j),\ j = 1, \dots, n\}, \tag{2}$$

and

$$\|f'(x) - f'(y)\| \leq K \,\|x - y\,\|, \quad x, y \in D. \tag{3}$$

A number of methods for solving this problem has been proposed (see, e. g., [3, 4, 5, 9, 12, 18, 19, 20, 21, 22]). They can be distinguished either by the mode in which information about the Lipschitz constant K from (3) is obtained or by the strategy of exploration of the search hyperinterval D from (2).

In particular, several ways to specify the Lipschitz constant K can be considered: this constant can be given a priori (see, e. g., [1, 2, 3]); its adaptive estimates (local or global) can be obtained during the search (see, e. g., [5, 7, 10, 11, 16, 18, 20, 22]); multiple estimates of the Lipschitz constant can be also used (see, e. g., [8, 9]).

In exploring the multidimensional search domain, various adaptive partitioning strategies can be applied. For example, one-point-based algorithms subsequently subdivide the search region in smaller ones and evaluate the objective function and its gradient at one point within each subregion (see, e. g., [3, 9]). Diagonal partitions that evaluate $f(x)$ and $f'(x)$ at two points within each subregion are very interesting for practical applications with expensive black-box functions (see, e. g., [14, 17, 18]). More complex partitions, based on simplices, auxiliary functions of various nature, and so on, can be also used (see, e. g., [4, 7, 12, 13, 21, 23]).

*This research was partially supported by the INdAM–GNCS 2014 Research Project of the Italian National Group for Scientific Computation of the National Institute for Advanced Mathematics "F. Severi".

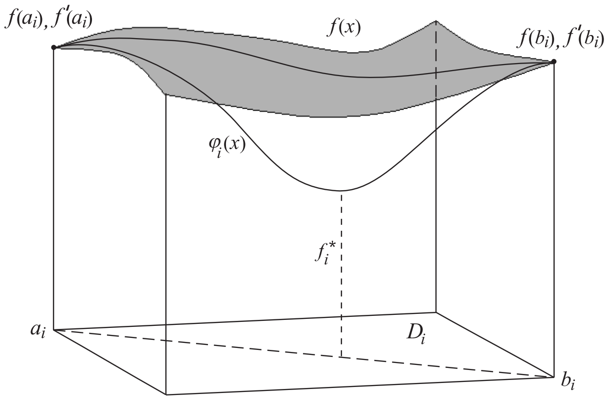


Figure 1: Obtaining the lower bound f_i^* for the objective function $f(x)$ with the Lipschitz gradient $f'(x)$ over hyperinterval $D_i = [a_i, b_i]$ by using smooth auxiliary function $\varphi_i(x)$ along the main diagonal $[a_i, b_i]$ of D_i.

2. New geometric global optimization methods using derivatives

The Lipschitz condition (3) is used in this work to obtain the lower bound of the global minimum value (1) of the objective function $f(x)$ at each iteration of a Lipschitz global optimization algorithm, thus allowing one to construct global optimization algorithms and to prove their convergence in a unified manner. The methods of this type form the class of geometric algorithms that are based on constructing, updating, and improving auxiliary piecewise functions built by using an estimate of the Lipschitz constant K from (3) (see, e. g., [1, 2, 8, 9, 10, 11, 20, 21]). It should be noted in this connection, that similar ideas are used in many other surrogate-based optimization methods (see, e.g.,[4]).

Since at each point $x \in D$ from (2) it is possible to evaluate both the objective function $f(x)$ and its gradient $f'(x)$, more information about the problem is available (especially, regarding its local properties expressed by the gradient values). The usage of this information allows one to construct auxiliary functions that fit closely the objective function and to accelerate the global search.

Different geometric Lipschitz global optimization methods based on constructing auxiliary functions with the usage of various estimates of the Lipschitz constant K from (3) are presented in this communication. A particular attention is given to the local tuning approach (see [16, 17, 20]) and a recently proposed technique (see [8, 9]) for using multiple estimates of K.

These methods use either the one-point-based partition strategy or the diagonal one (see, e. g., [14, 18]). In both the cases, the initial hyperinterval D from (2) is partitioned into a set of smaller hyperintervals D_i, the objective function and its gradient are evaluated only at one or two vertices corresponding to the main diagonal of hyperintervals of the current partition of D (see points a_i and b_i of a hyperinterval D_i in Figure 1 for the case of a diagonal algorithm), and the results of these evaluations are used to select a hyperinterval for the further subdivisions. The diagonal approach has a number of attractive theoretical properties and has proved to be efficient in solving applied problems (see, e. g., [14, 17, 18, 20]).

Particularly, the diagonal approach allows one to easily perform an extension of efficient univariate global optimization algorithms to the multidimensional case (see, e.g., [11, 15, 16, 17, 18, 19, 20]), as in both the cases of the local tuning approach (with the construction of smooth auxiliary functions, see [16, 20]) and of the usage of multiple estimates of K (see [8, 9]). In fact, in order to estimate the lower bound of $f(x)$ over a multidimensional subregion D_i, some one-dimensional bounds can be used as prototypes. After an appropriate transformation they can be applied to the one-dimensional segment being the main diagonal of the hyperinterval D_i (see a Lipschitz-based smooth auxiliary function $\varphi_i(x)$ in Figure 1, under investigation in [20]).

3. Numerical comparison

A special attention in the talk is dedicated to testing the proposed methods and to comparing them with some well-known Lipschitz global optimization algorithms (see, e. g., [4]) with the usage of the GKLS-generator [6]. This generator constructs three types (non-differentiable, continuously differentiable, and twice continuously differentiable) of classes of multidimensional and multiextremal test functions with *known* local and global minima. The generation procedure consists of defining a convex quadratic function systematically distorted by polynomials. Each test class provided by the generator consists of 100 functions and is defined by the following parameters: (i) problem dimension, (ii) number of local minima, (iii) global minimum value, (iv) radius of the attraction region of the global minimizer, (v) distance from the global minimizer to the quadratic function vertex. The other necessary parameters are chosen randomly by the generator for each test function of the class. More information can be found at: *http://wwwinfo.dimes.unical.it/∼yaro.*

Table 1: Number of trials performed by the methods for solving 800 GKLS test problems.

n	ε	Class	Trials			Improvement	
			DIRECT	DIRECT*l*	MULT*K*	w.r.t. DIRECT	w.r.t. DIRECT*l*
2	10^{-4}	simple	1159	2318	335	3.46	6.92
2	10^{-4}	hard	3201	3414	1075	2.98	3.18
3	10^{-6}	simple	12507	13309	2043	6.12	6.51
3	10^{-6}	hard	>1000000 (4)	29233	2352	>425.17	12.43
4	10^{-6}	simple	>1000000 (4)	118744	16976	>58.91	6.99
4	10^{-6}	hard	>1000000 (7)	287857	20866	>47.92	13.80
5	10^{-7}	simple	>1000000 (1)	178217	16300	>61.35	10.93
5	10^{-7}	hard	>1000000 (16)	>1000000 (4)	88459	>11.30	>11.30

For the sake of example, let us report some numerical results obtained by using the Lipschitz global optimization method from [9]. This method (let us call it MULT*K*) is based on an efficient one-point-sampling diagonal partition strategy and uses multiple estimates of the Lipschitz constant K. In [9], this algorithm has been compared on the GKLS-generator with two algorithms belonging to the same class of methods for solving problem (1)–(3) — the DIRECT algorithm and its locally-biased modification DIRECT*l* (see, e. g. [4]). Continuously differentiable GKLS-classes of dimensions $n = 2$, 3, 4, and 5 were considered; for each particular problem dimension n a 'simple' and a 'hard' classes have been taken for the comparison (see [4] for a detailed description of the classes). Numbers of trials (evaluations of both $f(x)$ and $f'(x)$) required for a given method to solve (in terms of obtaining a relative ε-approximation of the global minimizer x^* from (1), see [9] for details) *all* 100 functions of a particular test class are given in Table 1. The notation '$>1\,000\,000\ (j)$' in Table 1 means that after 1 000 000 evaluations the method under consideration was not able to solve j problems. The last two columns of this Table represent the ratio between the maximal number of trials performed by DIRECT and DIRECT*l* with respect to the corresponding number of trials performed by the MULT*K* algorithm.

According to these results, the MULT*K* algorithm requires (on a given set of 800 test problems) much fewer trials than the other two methods to ensure a thorough examination of the search domain. Moreover, the advantage of the proposed method becomes even more pronounced as the problem dimension grows or the problem complexity increases.

References

[1] W. Baritompa and A. Cutler, "Accelerations for global optimization covering methods using second derivatives", *Journal of Global Optimization*, **4**(3):329–341, 1994.

[2] L. Breiman and A. Cutler, "A deterministic algorithm for global optimization", *Mathematical Programming*, **58**(1-3):179–199, 1993.

[3] Yu. G. Evtushenko and M. A. Posypkin, "A deterministic approach to global box-constrained optimization", *Optimization Letters*, **7**(4):819–829, 2013.

[4] C. A. Floudas and P. M. Pardalos, (Eds.), *Encyclopedia of Optimization (6 Volumes)*, Springer, New York, 2009.

[5] J. M. Fowkes, N. I. M. Gould, and C. L. Farmer, "A branch and bound algorithm for the global optimization of Hessian Lipschitz continuous functions", *Journal of Global Optimization*, **56**(4):1791–1815, 2013.

[6] M. Gaviano, D. E. Kvasov, D. Lera, and Ya. D. Sergeyev, "Algorithm 829: Software for generation of classes of test functions with known local and global minima for global optimization", *ACM Transactions on Mathematical Software*, **29**(4):469–480, 2003.

[7] S. Yu. Gorodetsky, "Paraboloid triangulation methods in solving multiextremal optimization problems with constraints for a class of functions with Lipschitz directional derivatives", *Vestnik of Lobachevsky State University of Nizhni Novgorod*, **1**(1):144–155, 2012. In Russian.

[8] D. E. Kvasov and Ya. D. Sergeyev, "A univariate global search working with a set of Lipschitz constants for the first derivative", *Optimization Letters*, **3**(2):303–318, 2009.

[9] D. E. Kvasov and Ya. D. Sergeyev, "Lipschitz gradients for global optimization in a one-point-based partitioning scheme", *Journal of Computational and Applied Mathematics*, **236**(16):4042–4054, 2012.

[10] D. E. Kvasov and Ya. D. Sergeyev, "Univariate geometric Lipschitz global optimization algorithms", *Numerical Algebra, Control and Optimization*, **2**(1):69–90, 2012.

[11] D. Lera and Ya. D. Sergeyev, "Acceleration of univariate global optimization algorithms working with Lipschitz functions and Lipschitz first derivatives", *SIAM Journal on Optimization*, **23**(1):508–529, 2013.

[12] R. Paulavičius and J. Žilinskas, *Simplicial Global Optimization*, Springer, New York, 2014.

[13] R. Paulavičius, Ya. D. Sergeyev, D. E. Kvasov, and J. Žilinskas, "Globally-biased DISIMPL algorithm for expensive global optimization", *Journal of Global Optimization*, **59**(2-3):545–567, 2014.

[14] J. Pintér, *Global Optimization in Action (Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications*, Kluwer, Dordrecht, 1996.

[15] Ya. D. Sergeyev, "An information global optimization algorithm with local tuning", *SIAM Journal on Optimization*, **5**(4):858–870, 1995.

[16] Ya. D. Sergeyev, "Global one-dimensional optimization using smooth auxiliary functions", *Mathematical Programming*, **81**(1):127–146, 1998.

[17] Ya. D. Sergeyev and D. E. Kvasov, "Global search based on efficient diagonal partitions and a set of Lipschitz constants", *SIAM Journal on Optimization*, **16**(3):910–937, 2006.

[18] Ya. D. Sergeyev and D. E. Kvasov, *Diagonal Global Optimization Methods*, FizMatLit, Moscow, 2008. In Russian.

[19] Ya. D. Sergeyev and D. E. Kvasov, "Lipschitz global optimization", in J. J. Cochran et al., (Eds.), *Wiley Encyclopedia of Operations Research and Management Science (in 8 volumes)*, John Wiley & Sons, New York, **4**:2812–2828, 2011.

[20] Ya. D. Sergeyev and D. E. Kvasov, "A deterministic global optimization using smooth diagonal auxiliary functions", *Submitted*, 2014.

[21] Ya. D. Sergeyev, R. G. Strongin, and D. Lera, *Introduction to Global Optimization Exploiting Space-Filling Curves*, Springer, New York, 2013.

[22] R. G. Strongin and Ya. D. Sergeyev, *Global Optimization with Non-Convex Constraints: Sequential and Parallel Algorithms*, Kluwer Academic Publishers, Dordrecht, 2000.

[23] A. Zhigljavsky and A. Žilinskas, *Stochastic Global Optimization*, Springer, New York, 2008.

Piecewise linearisation of the first order loss function for families of arbitrarily distributed random variables

Roberto Rossi[1] and Eligius M. T. Hendrix[2]

[1] *Business School, University of Edinburgh, Edinburgh, United Kingdom,* roberto.rossi@ed.ac.uk

[2] *Department of Computer Architecture, Málaga University, Málaga, Spain* eligius.hendrix@wur.nl

Abstract We discuss the problem of computing optimal linearisation parameters for the first order loss function of a family of arbitrarily distributed random variable. We demonstrate that, in contrast to the problem in which parameters must be determined for the loss function of a single random variable, this problem is nonlinear and features several local optima and plateaus. We introduce a simple and yet effective heuristic for determining these parameters and we demonstrate its effectiveness via a numerical analysis carried out on a well known stochastic lot sizing problem.

Keywords: First order loss function, piecewise linear bounds, Jensen, Edmundson-Madanski, lot sizing

1. Introduction

Consider a random variable $\boldsymbol{\omega}$ with expected value μ and density function g_ω, and a scalar variable x. The first order loss function is defined as $\mathcal{L}_\omega(x) = \mathrm{E}_\omega[\max(\boldsymbol{\omega} - x, 0)]$, where E denotes the expected value. The complementary first order loss function is defined as $\bar{\mathcal{L}}_\omega(x) = \mathrm{E}_\omega[\max(x - \boldsymbol{\omega}, 0)]$. Note that $\mathcal{L}_\omega(x) = \mu - x + \bar{\mathcal{L}}_\omega(x)$. The first order loss function $\mathcal{L}_\omega(x)$ and its complementary function $\bar{\mathcal{L}}_\omega(x)$ are extensively used in several application domains, such as inventory control [8] and finance (see e.g. [5]).

In general, $\mathcal{L}_\omega(x)$ does not admit a closed form and cannot be evaluated without resorting to numerical approximations (see e.g. [1]). $\mathcal{L}_\omega(x)$ and its numerical approximations are nonlinear in x and cannot directly be embedded in mixed integer linear programming (MILP) models.

In [7] the authors discuss piecewise linear upper and lower bounds for the first order loss function, which can be immediately embedded in MILP models. These bounds are particularly convenient for a number of reasons: they rely on constant parameters that are independent of the mean and standard deviation of the normal distribution of interest; it is easy to obtain bounds for generic, i.e. non standard, normally distributed random variables via a simple linear transformation. Optimal linearisation parameters are derived following an approach similar to the one discussed in [2, 3], which minimise the maximum approximation error.

In this work, we extend the approach in [7] to the case of generic, i.e. non normal, distributions and we discuss how to embed into a MILP model piecewise linear upper and lower bounds of the first order loss function for a predefined family of generic random variables $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \ldots, \boldsymbol{\omega}_N$. These bounds are computed in such a way as to minimise the maximum approximation error over the given family of random variables. We demonstrate the effectiveness of this technique to address a well known stochastic lot sizing problem. Because of the relation that exists between the function $\mathcal{L}_\omega(x)$ and its complement $\bar{\mathcal{L}}_\omega(x)$, the following discussion will be limited to the complementary first order loss function $\bar{\mathcal{L}}_\omega(x)$.

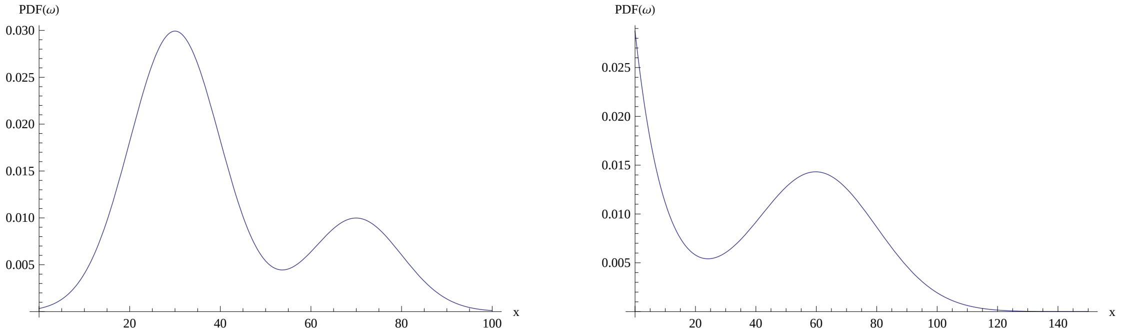


Figure 1: Probability density function (PDF) of ω_1 (left) and ω_2 (right).

2. Piecewise linear upper and lower bounds

The complementary first order loss function function $\bar{\mathcal{L}}_\omega(x)$ is convex in x regardless of the distribution of $\boldsymbol{\omega}$. For this reason, both Jensen's lower bound and Edmundson-Madanski upper bound are applicable [4]. Consider a partition of the support Ω of $\boldsymbol{\omega}$ into W disjoint compact subregions $\Omega_1, \dots, \Omega_W$. We define, for all $i = 1, \dots, W$

$$p_i = \Pr\{\boldsymbol{\omega} \in \Omega_i\} = \int_{\Omega_i} g_\omega(t)\, dt \quad \text{and} \quad \mathrm{E}[\boldsymbol{\omega}|\Omega_i] = \frac{1}{p_i} \int_{\Omega_i} t g_\omega(t)\, dt. \tag{1}$$

Lemma 1. *Let $\Omega_1, \dots, \Omega_W$ be a partition of the support Ω of random variable $\boldsymbol{\omega}$, p_i and $E[\boldsymbol{\omega}|\Omega_i]$ defined by (1) and lower bounding function $\Lambda^i_\omega(x) = x \sum_{k=1}^i p_k - \sum_{k=1}^i p_k E[\boldsymbol{\omega}|\Omega_k]$. Then lower bounding function*

$$\Lambda_\omega(x) = \max\left(\max_i \Lambda^i_\omega(x), 0\right) \le \bar{\mathcal{L}}_\omega(x)$$

is a piecewise linear function with $W+1$ segments.

This lower bound is a direct application of Jensen's inequality. Let us then consider the maximum approximation error $e_W = \max_x(\bar{\mathcal{L}}_\omega(x) - \Lambda_\omega(x))$ for the lower bound in Lemma 1 associated with a given partition. A piecewise linear upper bound, i.e. the Edmundson-Madanski bound, is $\Lambda_\omega(x) + e_W$, which is obtained by shifting up the lower bound in Lemma 1 by a value e_W. These lower and upper bounds for any random variable $\boldsymbol{\omega}$ can directly be used in an MILP model.

Having established this results, the question is how to partition the support Ω in order to obtain good bounds. A number of works discussed how to obtain an optimal partitioning of the support under a framework that minimises the maximum approximation error [2, 3]. In short, these works demonstrate that, in order to minimise the maximum approximation error, one must find parameters ensuring approximation errors at piecewise function breakpoints are all equal. This result unfortunately does not hold when optimal linearisation parameters must be found for complementary first order loss functions of a family of generic random variables.

Consider the complementary first order loss functions for a family of generic random variables $\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_n, \dots, \boldsymbol{\omega}_N$. From (1) it is clear that $\mathrm{E}[\boldsymbol{\omega}|\Omega_i]$ is uniquely determined by the choice of p_i. From this choice of the coefficients p_i follows for each $\boldsymbol{\omega}_n$ the function $\Lambda_\omega(x)$. Within an MILP model, one can select the desired bounding function via a binary selector variable y_n:

$$\Lambda(x) = \sum_{n=1}^N \max\left(\max_i \left(x \sum_{k=1}^i p_k - y_n \sum_{k=1}^i p_k \mathrm{E}[\boldsymbol{\omega}_n|\Omega_k]\right), 0\right)$$

adding $\sum_{n=1}^N y_n = 1$. These expressions generalise those discussed in [7], which only hold for normally distributed random variables.

The challenge is, of course, to compute an optimal partition of random variable supports into W disjoint compact subregions with probability masses $p_1, \ldots, p_i, \ldots, p_W$. We shall first demonstrate that this is a nonconvex optimisation problem in contrast to computing optimal linearisation parameters for a single loss function. To do so, we consider a simple instance involving two random variables $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ with probability density function as shown in Fig. 1.

We split the support of $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ into five regions with probability mass $p_1, \ldots, p_5$, respectively. In Fig. 2 we plot the maximum approximation error

$$e_W = \max\left(\max_x(\bar{\mathcal{L}}_{\omega_1}(x) - \Lambda_{\omega_1}(x)), \max_x(\bar{\mathcal{L}}_{\omega_2}(x) - \Lambda_{\omega_2}(x))\right),$$

when p_1 and p_4 are free to vary, $p_2 = 0.3$, $p_3 = 0.1$ and $p_5 = 1 - p_1 - p_2 - p_3 - p_4$; note that this is a standard 2-simplex in $\mathbb{R}^3$ projected in $\mathbb{R}^2$ and coloured to reflect the value of e_W. It is clear that this function has a number of local minima. In fact, the function is also constant in some regions, i.e. it has wide plateaus. It is intuitive to observe this, if one considers the fact that the slope of the i-th linear segment is given by $\sum_{i=1}^{k} p_i$ and that by varying the slope of one or more segments the maximum approximation error — attained at one or more breakpoints — may easily remain the same. Finding a global optimum of this function is a challenge.

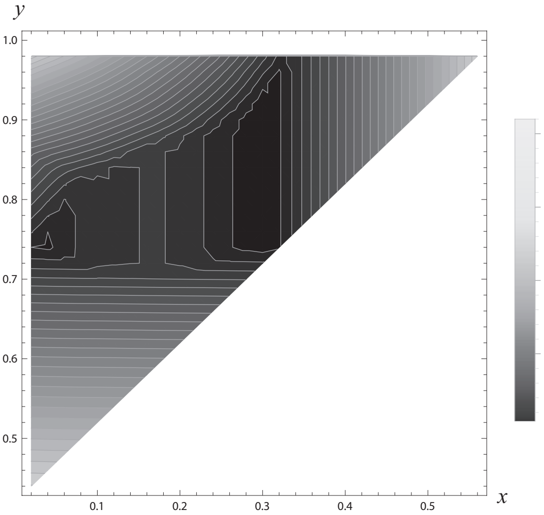


Figure 2: Maximum approximation error of our piecewise linear lower bound when $p_1 + p_4 + p_5 \leq 1 - p_2 - p_3$; the x axis represents p_1, i.e. the slope of segment 1, the y axis represents $p_1 + p_2 + p_3 + p_4$, i.e. the slope of segment 4; darker regions mean lower error.

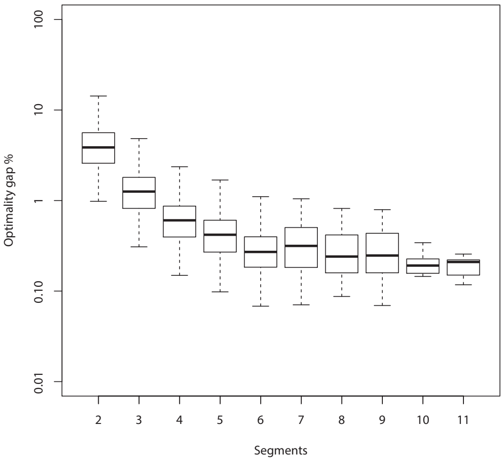


Figure 3: Optimality gap trend as a function of the number of segments used in the linearisation. The optimality gap is presented as a percentage of the optimal solution obtained with the model embedding a piecewise linear upper bound.

Therefore we developed a metaheuristic to find good, but not necessarily optimal parameter values. The heuristic is a combination of simple random sampling (SRS) and coordinate descent (CD) from the best solution produced by the SRS. For the above example, this strategy produced the following partitioning: $p_1 = 0.24$, $p_2 = 0.18$, $p_3 = 0.215$, $p_4 = 0.175$, $p_5 = 0.19$, with associated maximum approximation error of 0.639. This approximation error amounts to 1.5% of the expected value of $\boldsymbol{\omega}_1$ and to 1.3% of the expected value of $\boldsymbol{\omega}_2$. As we will demonstrate, this strategy produced fairly good outcomes in practical applications.

3. An application to stochastic lot-sizing

We applied the metaheuristic to compute near optimal linearisation parameters for the stochastic lot sizing problem discussed in [6]. We tested the approach over a test bed discussed in [6] comprising 270 instances. In our experiments, in contrast to the original test bed in which demand is normally distributed, demand follows different distributions in different periods: normal, Poisson, exponential and uniform. All instances took just a few seconds to be solved. Note that each of these instances comprises $N = 15$ periods in which demand is observed. Consequently, there are $N(N + 1)/2$ loss functions in the family for which optimal linearisation parameters must be computed; these parameters must be computed for each instance separately, this is why a lightweight heuristic is desirable. The average optimality gap trend — the difference between the optimal solution of the model embedding our piecewise linear upper bound and that of the model embedding our piecewise linear lower bound — as a function of the number of segments used in the linearisation is shown in Fig. 3. As we see, the gap drops initially with the number of segments and it is well below 1% of the optimal cost even when just four segments are employed. For higher number of segments the gap fluctuates. This is due to the fact that the simple heuristic here proposed gets stuck in local minima for high dimensional spaces. Future research should therefore investigate more effective approaches to comput optimal parameters for linearisations involving a large number of segments.

4. Conclusions

We have shown that finding optimal linearisation parameters to approximate loss functions when several non-normal random variables are considered, is a challenging global optimisation problem. We discuss how to handle this in a practical setting to generate reasonable results that can be used in MILP models for inventory control.

References

[1] S.K. De Schrijver, El-H. Aghezzaf, and H. Vanmaele. Double precision rational approximation algorithms for the standard normal first and second order loss functions. *Applied Mathematics and Computation*, 219(4):2320–2330, November 2012.

[2] M.M. Gavrilović. Optimal approximation of convex curves by functions which are piecewise linear. *Journal of Mathematical Analysis and Applications*, 52(2):260–282, November 1975.

[3] A. Imamoto and B. Tang. Optimal piecewise linear approximation of convex functions. In S.I. Ao, C. Douglas, W.S. Grundfest, L. Schruben, and J. Burgstone, editors, *Proceedings of the World Congress on Engineering and Computer Science 2008 WCECS 2008, October 22 - 24, 2008, San Francisco, USA*, pages 1191–1194. IAENG, 2008.

[4] P. Kall and S.W. Wallace. *Stochastic Programming (Wiley Interscience Series in Systems and Optimization)*. John Wiley & Sons, August 1994.

[5] R. Mansini, W. Ogryczak, and M.G. Speranza. Conditional value at risk and related linear programming models for portfolio optimization. *Annals of Operations Research*, 152(1):227–256, July 2007.

[6] R. Rossi, O.A. Kilic, and S.A. Tarim. Piecewise linear approximations for the static-dynamic uncertainty strategy in stochastic lot-sizing, December 2013.

[7] R. Rossi, S.A. Tarim, S. Prestwich, and B. Hnich. Piecewise linear lower and upper bounds for the standard normal first order loss function. *Applied Mathematics and Computation*, 231:489–502, March 2014.

[8] E.A. Silver, D.F. Pyke, and R. Peterson. *Inventory Management and Production Planning and Scheduling, 3rd Edition*. Wiley, 3 edition, January 1998.

A branch and bound method for global robust optimization*

Emilio Carrizosa[1] and Frédéric Messine[2]

[1] *University of Sevilla, Sevilla, Spain,* ecarrizosa@us.es

[2] *University of Toulouse, ENSEEIHT-LAPLACE, Toulouse, France* Frederic.Messine@n7.fr

Abstract In this paper, we study general nonlinear and nonconvex robust optimization problems. This leads us to create a Branch and Bound algorithm based on interval arithmetic. This algorithm can provide the exact global solution of such difficult problems arising in many real life applications. A code was developed in `MatLab` and was used to solve some small robust nonconvex problems with a few number of variables. This first numerical study showed the interest of this approach providing global optimum of such difficult robust nonconvex optimization problems.

Keywords: Robust Optimization, Interval Arithmetic, Branch and Bound

1. Introduction

While most papers in the literature on robust optimization address the fully convex case, some efforts have been made to cope with the more realistic situation in which nonconvexities appear. For instance, [5] addresses nonconvex problems, for which a first-order approximate robust model is proposed, and thus applicable when a good approximate of the uncertain parameters are known. Robust local-search procedures for problems in which the objective may be evaluated via simulations are described in [2, 3]. See [1, 4] for recent references containing an updated review of models, algorithmic tools and applications fields. In summary, research efforts on the theory of robust optimization have focused on creating and analyzing distributionally robust approaches as well as developing connections between uncertainty sets and risk theory.

In this paper, we develop a new algorithm based on a Branch and Bound scheme to provide the global solution of a general class of robust nonlinear and nonconvex problems. Some properties are first studied in the following section and the algorithm is then provided. In this paper, just the main steps of the algorithm are presented. An example will validate our approach by solving a difficult robust nonlinear and nonconvex optimization problem.

2. Problem statement and properties

Given a function f, consider the *nominal problem* of optimizing f over the box $\tilde{X} \subset \mathbb{R}^n$ as

$$\min_{x \in \tilde{X}} f(x). \tag{1}$$

The *robust counterpart* of (1) is obtained when each solution $x \in \tilde{X}$ is perturbed by a vector $y \in \tilde{Y}$, and a worst-case analysis is performed. This leads us to the following optimization problem:

$$\min_{x \in \tilde{X}} \max_{y \in \tilde{Y}} f(x+y). \tag{2}$$

*This work of F. Messine has been funded by Junta de Andalucía (P11-TIC7176)

The set $\tilde{Y}$ of perturbations, called the *uncertainty set*, is assumed here to be a box in $\mathbb{R}^n$ (compare e.g. with [2, 3], in which an Euclidean ball is used as uncertainty set), and f is assumed to be continuous on $\tilde{X} + \tilde{Y}$.

Defining $z : \tilde{X} \longrightarrow \mathbb{R}$ as

$$z(x) = \max_{y \in \tilde{Y}} f(x + y), \tag{3}$$

we write our problem as the one of maximizing z on $\tilde{X}$:

$$\min_{x \in \tilde{X}} z(x). \tag{4}$$

In the following part of this section, some properties are derived on problem (4).

Proposition 1. *If f is convex on $\tilde{X} + \tilde{Y}$, then z is convex on $\tilde{X}$, and thus any local optimum of (4) is a global optimum.*

Here we are interested in the nonconvex case, for which global optimization tools are needed. In particular, a branch and bound algorithm is proposed here to find a global optimum of (4). We assume in what follows that we have available an inclusion function F for f. For any box $I \subset \tilde{X} + \tilde{Y}$, let $ubF(I)$ (respectively $lbF(I)$) denotes the upper bound (respectively the lower bound) of the interval $F(I)$.

Lower and upper bounds of $\min_{x \in X} z(x)$ are easily obtained from the inclusion function F, as shown in the following properties.

Proposition 2. *Given the box $X \subset \tilde{X}$, one has*

$$\min_{x \in X} z(x) \geq lbF(X + y^*) \qquad \forall y^* \in \tilde{Y}. \tag{5}$$

Proposition 3. *Given the box $X \subset \tilde{X}$, suppose $Y^1, \ldots, Y^k \subset \tilde{Y}$ are boxes known to satisfy*

$$z(x) = \max_{y \in \bigcup_{j=1}^k Y^j} f(x + y) \qquad \forall x \in X.$$

Then

$$\min_{x \in X} z(x) \leq \max_{1 \leq j \leq k} ubF(x^* + Y^j) \qquad \forall x^* \in X$$

Proposition 4. *Given boxes $X \subset \tilde{X}$, $Y \subset \tilde{Y}$, if $ubF(X + Y) < lbF(X + y_0)$ for some $y_0 \in \tilde{Y}$, then*

$$f(x + y) < z(x) \qquad \forall x \in X.$$

In other words, the box Y is useless in order to compute z at points in the box X, and can thus be eliminated from the list of boxes to be inspected.

Proposition 5. *Let f be differentiable in $\tilde{X} + \tilde{Y}$, and let F'_j denote an inclusion function for its partial derivative with respect to the j-th coordinate. Given boxes $X \subset \tilde{X}$, $Y = \prod_{j=1}^n Y_j \subset \tilde{Y} = \prod_{j=1}^n \tilde{Y}_j$, Suppose $x^* \in X$ and $y^* \in Y$ exist such that x^* is an optimal solution to (4) and $z(x^*) = f(x^* + y^*)$. If $ubF'_j(X + Y) < 0$, then one has*

1. *$lbY = lb\tilde{Y}$*
2. *$lbX = lb\tilde{X}$*

Proof. We have by assumption that $f'_j(x^* + y^*) < 0$, and then, the function $t \longmapsto f(x^* + y^* + te_j)$ is decreasing in a neighborhood of 0. This implies that $f(x^* + y^* - te_j) > f(x^* + y^*) = z(x^*)$ for some t close to 0. Hence no such $t > 0$ makes $y^* - te_j \in \tilde{Y}$, which implies condition 1, and no such $t > 0$ makes $x^* - te_j \in \tilde{X}$, which implies condition 2.

In the same way one obtains the counterpart for $lbF'_j(X+Y)$.

Proposition 6. *Let f be differentiable in $\tilde{X}+\tilde{Y}$, and let F'_j denote an inclusion function for its partial derivative with respect to the j-th coordinate. Given boxes $X \subset \tilde{X}$, $Y = \prod_{j=1}^n Y_j \subset \tilde{Y} = \prod_{j=1}^n \tilde{Y}_j$. Suppose $x^* \in X$ and $y^* \in Y$ exist such that x^* is an optimal solution to (4) and $z(x^*) = f(x^*+y^*)$. If $lbF'_j(X+Y) > 0$, then one has*

1. *$ubY = ub\tilde{Y}$*
2. *$ubX = ub\tilde{X}$*

Propositions 5-6 are key for the following test: Given the pair (X,Y) if $ubF'_j(X+Y) < 0$, then the pair $(X,Y) = (\prod_{k=1}^n X_k, \prod_{k=1}^n Y_k)$ can be replaced in the list by the degenerate pair $(\prod_{k=1}^n X_k^*, \prod_{k=1}^n Y_k^*)$, with

$$X_k^* = \begin{cases} X_k, & \text{if } k \neq j \\ ub\tilde{X}_j, & \text{if } k = j \end{cases} \qquad Y_k^* = \begin{cases} Y_k, & \text{if } k \neq j \\ ub\tilde{Y}_j, & \text{if } k = j \end{cases}$$

if $ub\tilde{X}_j = ubX_j$ and if $ub\tilde{Y}_j = ubY_j$, and otherwise the pair (X,Y) can be eliminated from further consideration, since it is then known that either Y is useless to evaluate z at points in $x \in X$, or points in X cannot be optimal to (4). Similarly, if $lbF'_j(X+Y) > 0$, then the j-th component of (X,Y) can be replaced by the degenerate interval consisting of the lower bounds, or eliminated.

When a box X is small enough, it can be replaced by its midpoint, since z cannot oscillate too much between X. This is formalized in the following.

Proposition 7. *Given the box $X = \prod_{j=1}^n X_j \subset \tilde{X}$, suppose $Y^1, \ldots, Y^k \subset \tilde{Y}$ are boxes known to satisfy*

$$z(x) = \max_{y \in \bigcup_{j=1}^k Y^j} f(x+y) \qquad \forall x \in X.$$

Let x_m denote the midpoint of X and, for $i = 1, 2, \ldots, k$, let y_m^i denote the midpoint of the box Y^i. Suppose f be continuously differentiable in $\tilde{X}+\tilde{Y}$, and let F'_j denote an inclusion function for its partial derivative with respect to the j-th coordinate. For $i = 1, \ldots, k$, define ε^i as

$$\varepsilon^i = \max\left\{ \sum_{j=1}^n ub|F'_j(X+y_m^i)|ub(|x_{mj}-X_j|), \sum_{j=1}^n ub|F'_j(x_m+Y^i)|ub(|y_m^i - Y_j^i|) \right\}.$$

Then

$$\left| \max_{1\leq i\leq k} f(x_m+y_m^i) - \min_{x\in X} z(x) \right| \leq \max_{1\leq i\leq k} \varepsilon^i. \qquad (6)$$

Hence, as soon as a box X is such that all associated boxes Y^i are small enough so that the length of the largest interval side is smaller than a given value, then we can replace X by its midpoint x_m, and $z(X)$ by the expression $\max_{1\leq i\leq k} f(x_m+y_m^i)$, and stopping further branching of X.

3. Branch and Bound Algorithm

The main idea of the code is that a list of element has a box X, a list of boxes Y^i and a list of points inside $\cup Y^i$. Then, from the above properties, we can develop a Branch and Bound algorithm. However, even if this algorithm is based on a standard interval Branch and Bound code, it differs in some points inside the main loop:

- An element of the list is constituted by: (i) a box X which is an interval vector; (ii) a list of boxes Y^i; (iii) a list of points inside $\cup Y^i$; (iv) a lower bound of f over $X \times \left(\cup Y^i\right)$.
- The element Z of the list with the lowest lower bound is taking first.
- In Z, the box X is bisected by the middle of its largest edge. In both cases the same lists of Y^i and of points inside $\cup Y^i$ have to be kept.
- In Z, the largest box of boxes Y^i (belonging in the list) the bisected following its largest edge if the length of this edge is greater than a fixed constant M. The midpoint of the two sub boxes of Y^i are inserted in the list of points inside $\cup Y^i$.
- Monotonicity test on y: For all, sub boxes Y^i in the list associated to box X, check if in one direction on y, $f(x, y)$ is monotonous over all the box X. In that case, eliminate Y^i in the list corresponding to box X or reduce Y^i to the side of the initial box Y. Moreover, if no sub boxes Y^i remains in the list, eliminate the box X of the main list. If the list of Y^i has changed then update the list of points associated to box X by inserting inserting new points generated by the middle of each new Y^i and by discarding points belonging in removed boxes.
- Monotonicity test on x: check the monotonicity about x if all the sub boxes in the list of Y^i are points.
- The computations of lower and upper bounds follow the proposition defined in the previous section.

In order to validate our algorithm, we solved the following robust optimization problem:

$$\min_{x\in[-10,10]^2} \max_{y\in[-0.1,0.1]} f(x,y) = \sum_{i=1}^{2} \left((x_i + y_i) - 2)^6 + 0.2\right) \ln\left(1 + (x_i + y_i)^2\right). \tag{7}$$

A global robust minimum $(1.5222, 1.5271)$ was obtained in 530 iterations corresponding to 12 minutes of CPU-time on a standard laptop. The value of the minimum is 0.52466 and is certified via interval arithmetic computations at 10^{-3}.

4. Conclusion

In this paper, we propose an exact algorithm to solve global robust nonconvex optimization problems. This algorithm is derived from some properties previously established. A first code was developed in MatLab and provides first solutions on these difficult problems. An example is described here in the last section showing the efficiency of our method.

References

[1] Bertsimas, D., Brown, D.B. and Caramanis, C. "Theory and Applications of Robust Optimization". *SIAM Review* **53** (2011) 464–501.

[2] Bertsimas, D., Nohadani, O. and Teo, K.M. "Nonconvex Robust Optimization for Problems with Constraints". *INFORMS Journal on Computing* **22** (2010) 44–58.

[3] Bertsimas, D., Nohadani, O. and Teo, K.M. "Robust optimization for unconstrained simulation-based problems". *Operations Research* **58** (2010), 161-178.

[4] Gabrel, V., Murat, C., and Thiele, A. "Recent Advances in Robust Optimization: An Overview". Technical report, 2013. Available as `www.optimization-online.org/DB_FILE/2012/07/3537.pdf`

[5] Zhang, Y. "General robust-optimization formulation for nonlinear programming". *Journal of Optimization Theory and Applications* **132** (2007) 111-124.

Node Selection Heuristics Using the Upper Bound in Interval Branch and Bound

Bertrand Neveu[1], Gilles Trombettoni[2], and Ignacio Araya[3]

[1] *LIGM, Université Paris Est, France,* Bertrand.Neveu@enpc.fr

[2] *LIRMM, Université Montpellier, France,* Gilles.Trombettoni@lirmm.fr

[3] *Pontificia Universidad Católica, Valparaiso, Chile,* rilianx@gmail.com

Abstract We present in this article a new strategy for selecting the current node in an interval Branch and Bound algorithm for constrained global optimization. The standard best-first strategy selects the node with the lowest lower bound of the objective estimate. We propose in this article new node selection policies where an *upper* bound of each node/box is also taken into account. These new strategies obtain better experimental results than classical best-first search on difficult instances.

Keywords: Global Optimization, Interval Branch and Bound, Node Selection

1. Introduction

The paper deals with continuous global optimization (nonlinear programming) deterministically handled by interval branch and bound (B&B). Several works have been performed for finding good branching heuristics [3], but very little work for the node selection itself. The solvers generally follow a best first search strategy (BFS), with some studies for limiting its exponential memory growth [1, 6]. Different variants of BFS have been proposed for discrete problems, such as K-Best-First Search [4].

To our knowledge, only Casado et al. in [1] and Csendes in [2] proposed node selection heuristics for interval B&Bs. One criterion to maximize, called C_3 in [5], and suitable for unconstrained global optimization is:

$$\frac{f^* - lb}{ub - lb}$$

where $[lb, ub] = [f]_N([x])$ is the interval obtained by the natural interval evaluation of the real-valued objective function f in the current box $[x]$: $[lb, ub]$ is a range interval including all real images of any point in $[x]$ by f. f^* is the optimum. When f^* is not known, $\widetilde{f}$, the cost of the best feasible point found so far can be used as an approximation of f^*. This criterion favors small boxes (i.e., a small interval width $(ub - lb)$) and nodes with good lb. For constrained optimization, another criterion (to maximize) called C_5 is equal to $C_3 \times fr$. It takes into account a feasibility ratio fr computed from all the inequality constraints. The criterion C_7 proposes to minimize $\frac{lb}{C_5}$.

This paper proposes two other policies taking into account an accurate upper bound of the optimum cost.

2. Standard Interval B&B

An interval $[x_i] = [\underline{x_i}, \overline{x_i}]$ defines the set of reals x_i s.t. $\underline{x_i} \leq x_i \leq \overline{x_i}$. A *box* $[x]$ is a Cartesian product of intervals $[x_1] \times ... \times [x_i] \times ... \times [x_n]$.

The paper deals with continuous global optimization under inequality constraints defined by: $\min_{x \in [x]} f(x)$ subject to $g(x) \leq 0$, where $f : \mathbb{R}^n \to \mathbb{R}$ is the real-valued objective function and $g : \mathbb{R}^n \to \mathbb{R}^m$ is a vector-valued function. $x = \{x_1, ..., x_i, ...x_n\}$ is a vector of variables varying in a domain/box $[x]$. x is said to be *feasible* if it satisfies the constraints.

A standard interval (or spatial) B&B scheme for continuous constrained global optimization (also known as nonlinear programming) is described in algorithms below. The B&B maintains two main types of information during search: $\widetilde{f}$, which is the cost of the best feasible point found so far, and f_{min} the minimum value of the lower bounds $lb([x])$ of the boxes/nodes $[x]$ to explore. In other terms, in every box $[x]$, there is a guarantee that no feasible point exists with a cost lower than $lb([x])$.

Let us first ignore the bold-faced instructions corresponding to the new strategy and detailed in Section 3. The algorithm is launched with the set g of constraints, the objective function f and with the initial box put into a list B of boxes to be handled. ϵ_{obj} is the required precision on the objective cost and is used as stopping criterion. We add a variable x_{obj} in the system (the vector x of variables) corresponding to the image (cost) of the objective function, and a constraint $f(x) = x_{obj}$. The criterion generally used in existing interval B&Bs is denoted in this paper by LBBox. It consists in selecting a node/box $[x]$ with a minimum lower bound estimate of the objective function ($lb([x])$). The selected box $[x]$ is then split into two sub-boxes $[x]_1$ and $[x]_2$ along one dimension (selected by another and more studied branching heuristic). Both sub-boxes are then handled by the Contract&Bound procedure.

```
IntervalBranch&Bound($B$, $g$, $f$) {
    While ($B \neq \emptyset$ and $\widetilde{f} - f_{min} > \epsilon_{obj}$) {
        criterion := criterionChoice(LBBox, UBBox, UBProb)
        $[x]$ := bestBox ($B$, criterion); $B := B \setminus [x]$
        $([x]_1, [x]_2)$ := bisect ($[x]$)
        $[x]_1$ := Contract&Bound($[x]_1$, $g$, $f$)
        $[x]_2$ := Contract&Bound($[x]_2$, $g$, $f$)
        $B := B \cup \{[x]_1\} \cup \{[x]_2\}$
        $f_{min}$ := $\min_{[x] \in B} lb([x])$
    }
}
```

```
Contract&Bound($[x]$, $g$, $f$) {
    UBBox := $\widetilde{f} - \epsilon_{obj} + \mathbf{0.1\epsilon_{obj}}$
    $g'$ := $g \cup \{x_{obj} \leq$ UBBox $\}$
    $[x]$ := contraction($[x]$, $g'$, $f$)
    if ($[x] \neq \emptyset$) {
        ($x_{ub}$, cost):= FeasibleSearch($[x]$,$g'$)
        if (cost $< \widetilde{f}$) {
            $\widetilde{f}$ := cost
            $\mathbf{ub([x])} := \widetilde{\mathbf{f}} - \epsilon_{\mathbf{obj}}$
        }
    }
    Return $[x]$
}
```

A constraint $x_{obj} \leq$ UBBox is first added to the system for decreasing the upperbound of the objective function in the box. This aims at finding a solution better than the current best feasible point. The procedure then *contracts* the handled box without loss of feasible part. In other words, some infeasible parts at the bounds of the domain are discarded by constraint programming (CP) and convexification algorithms. Since this contraction works on the extended box including the objective variable x_{obj}, it may improve both bounds of the objective on the current box.

The last part of the procedure carries out upperbounding. *FeasibleSearch* calls one or several heuristics searching for a feasible point that improves the best cost found so far.

Note that the search tree is traversed in best-first order so that an exponential memory may be required to store the nodes to handle.

3. New Strategies Using Upper Bounds

In optimization, the selection of the next node to expand is crucial for obtaining a good performance. The best node we can choose is such that it will improve the most the upperbound. Indeed, the upperbound improvement reduces *globally* the feasible space due to the constraint: $x_{obj} \leq \widetilde{f}$. There exist two phases in a branch and bound: a phase where we try to find the optimal solution, and a second phase where we prove that this solution is optimal, which requires us expand all the remaining nodes. Therefore the node selection matters only in the first phase.

We define new strategies aggregating two criteria for selecting the current box:

1. LBBox: The well known criterion used by BFS and minimizing $lb([x])$ (for all boxes $[x]$ in the set B). This criterion is optimistic since we hope to find a solution with cost f_{min}, in which case the search would end. For each box, $lb([x])$ is computed by the Contract&Bound procedure and the computed value labels the node stored into the set B of boxes.
2. UBBox: This criterion selects the node having the smallest goal upperbound. Thus, if a feasible point was found inside this box, it would more likely improve the best cost found so far.

The UBBox criterion is symmetric to the LBBox one: for every box, $ub([x])$ is computed by the Contract&Bound procedure and labels the node before storing it in the set B of boxes. In particular, constraint programming techniques like 3BCID [7] can improve $ub([x])$ by discarding small slices at the upper bound of $[x_{obj}]$ (shaving process).

We think that a key of success of the UBBox criterion is that it evaluates more accurately the objective upperbound than the natural interval evaluation would do (i.e., ub computed by $[lb, ub] := [f]_N([x])$).

We propose two main ways to aggregate these two criteria.

LB+UBBox. This strategy selects the node $[x]$ with the lowest value of the sum $lb([x]) + ub([x])$. This corresponds to minimizing both criteria with the same weight, i.e. minimizing the middle of the interval of the objective estimate in the box.

Alternating both criteria. In this second strategy, the next box to handle is chosen using *one* of the two criteria. A random choice is made by the $criterionChoice$ function at each node selection, with a probability UBProb of choosing UBBox. If UBBox (resp. LBBox) is chosen and several nodes have the same cost $ub([x])$ (resp. $lb([x])$), then we use the other criterion LBBox (resp. UBBox) to tie breaks.

Experiments showed that the performance is not sensitive to a fine tuning of the UBProb parameter provided it remains between 0.2 and 0.8, so that the parameter has been fixed to 0.5. The experiments in Section 5 highlight the positive impact of this criterion on performance.

These results suggest that it is important to invest both in intensification (UBBox) and diversification (LBBox). In other words, the use of a second criterion allows the search to avoid the drawback of using one criterion alone, i.e. (for LBBox) choosing promising boxes with no feasible point and (for UBBox) going deeply in the search tree where only slightly better solutions will be found trapped inside a local minimum.

3.1 Details on the criterion UBBox

All candidate boxes in the set B, have a UB label depending on the best cost found so far ($\widetilde{f}$) when the boxes were handled by Contract&Bound. All labels fall in four main cost ranges categories given by $\widetilde{f}$ and explaining with which priority the boxes are chosen using the UBBox criterion. The label is:

1. lower than $\widetilde{f} - \epsilon_{obj}$ if the contraction procedure reduced the maximum estimate of the objective in the box,
2. equal to $\widetilde{f} - \epsilon_{obj}$, if the box is a descendant of the box containing the current best feasible point $\widetilde{f}$,
3. equal to $\widetilde{f} - 0.9\,\epsilon_{obj}$ if the box was handled after the last update of best cost,
4. greater than $\widetilde{f} - 0.9\,\epsilon_{obj}$ in the remaining case.

As shown in the Contract&Bound pseudocode, the additional term $0.1\,\epsilon_{obj}$ allows penalizing the boxes that are not issued by a bisection from boxes where the current best feasible point was found.

4. Implementation of the Set B of Boxes

The set B was initially implemented by a heap structure ordered on the LBBox criterion. The implementation `s1-05` keeps this unique data structure for taking into account the two criteria in the randomized strategy, but the node selection using UBBox comes at a linear cost in the number N of nodes. In practice, this takes about 10% of the total cost when N exceeds 50,000. We have then built a variant `s1-05-01` changing dynamically the probability UBProb in the following way: UBProb $= 0.5$ if $N \leq 50,000$ and UBProb $= 0.1$ if $N > 50,000$.

Finally, we tried a cleaner implementation, `s2-nodiv`, with two heaps, one for each criterion. All the operations are then in $log_2(N)$, except for the heap filtering process launched occasionnally during search. (This "garbage collector" is performed in all our implementations and removes from B all the nodes with $lb([x]) > \widetilde{f}$.)

The `s2-nodiv` implementation brings less diversification than the first one which reconstructs the heap at each criterion change. Indeed, there often exist several nodes with the same UBBox and LBBox values, so it is useful to periodically rebuild the data structures randomly for breaking the ties. So we propose variants that use a second parameter corresponding to a diversification period. We obtained good results by fixing it to 50 in the `s2-50` variant or 100 in the `s2-100` variant, the first parameter UBProb being still fixed to 0.5. To be fair, we also applied the first strategy with the minUB+LB aggregative criterion running heap filtering every 100 nodes (`s0-100`).

5. Experiments

We have run experiments on 82 problems, issued from the series 1 and 2 of the Coconut benchmark. The best strategies `s2-50` and `s2-100` obtain a gain of about 40% w.r.t. the total time and 23% on average, meaning that greater gains are obtained on difficult problems.

6. Summary

The node selection policy is a promising line of research to improve performance of interval B&B. We have obtained good results by taking into account for each node a lower bound but also an upper bound, provided that this upper bound is made accurate by box contraction operations, by a random selection between both criteria and by a work on heap data structures. In a short term, we are going to investigate how relevant components of criteria proposed by Markot and al. in [5] can improve the current policies, especially the feasibility ratio.

References

[1] L.G. Casado, J.A. Martinez, and I. Garcia. Experiments with a New Selection Criterion in a Fast Interval Optimization Algorithm. *Journal of Global Optimization*, 19:247–264, 2001.

[2] T. Csendes. New Subinterval Selection Criteria for Interval Global Optimization. *Journal of Global Optimization*, 19:307–327, 2001.

[3] T. Csendes and D. Ratz. Subdivision Direction Selection in Interval Methods for Global Optimization. *SIAM Journal on Numerical Analysis*, 34(3), 1997.

[4] A. Felner, S. Kraus, and R. E. Korf. KBFS: K-Best-First Search. *Annals of Mathematics and Artificial Intelligence*, 39, 2003.

[5] M.C. Markot, J. Fernandez, L.G. Casado, and T. Csendes. New Interval Methods for Constrained Global Optimization. *Mathematical Programming*, 106:287–318, 2006.

[6] J. Ninin and F. Messine. A Metaheuristic Methodology Based on the Limitation of the Memory of Interval Branch and Bound Algorithms. *Journal of Global Optimization*, 50:629–644, 2011.

[7] G. Trombettoni and G. Chabert. Constructive Interval Disjunction. In *Proc. CP*, volume 4741 of *LNCS*, pages 635–650. Springer, 2007.

Proceedings of MAGO 2014, pp. 61 – 64.

Largest Inscribed Ball and Minimal Enclosing Box for Convex Maximization Problems

Guillaume Guérard[1] and Ider Tseveendorj[1]

[1] *Laboratoire PRiSM, Université de Versailles*
45, avenue des États-Unis, 78035 Versailles Cedex, France
guillaume.guerard@prism.uvsq.fr
Ider.Tseveendorj@prism.uvsq.fr

Abstract Many important classes of decision models give rise to the problem of finding a global minimum of a concave function over a convex set. Since such a function may have many local minima, finding the global minimum is a computationally difficult problem, where standard nonlinear programming procedures fail. The two proposed methods are simple and quick, using the largest inscribed ball and the minimal enclosing box as approximation for cutting-plane method.

Keywords: convex maximization, ball center

1. Introduction

In certain classes of nonlinear problems the local solution is always the global one. For example, in minimization problems with a convex (or quasi-convex) objective function subject to convex constraints the local minimum is the global solution. For non-convex functions there may be many local minima so that no local criteria will give information about the global minimum.

In this article, we consider the non-convex optimization problem, also known as concave minimization, concave programming or convex maximization:

$$\begin{cases} \text{maximize } f(x), \\ \text{subject to } x \in D \end{cases} \tag{1}$$

where $f : \mathcal{R}^n \to \mathcal{R}$ is a convex continuous function and D is a nonempty, convex compact in $\mathcal{R}^n$, a polytope defined by

$$D = \{x \in \mathcal{R}^n \mid Ax \leq b\} = \{x \in \mathcal{R}^n \mid \langle a^i, x\rangle \leq b_i,\ i = 1, \dots, m\}.$$

For the state-of-the-art in convex maximization including various algorithms and abundant applications, we refer to the textbooks [9, 10] and to survey [1].

An important property of convex functions is that every local and global maximum is achieved at some extreme point of the feasible domain [13]. Several interesting necessary and sufficient global optimality conditions characterizing a point $z \in D$ satisfying $f(z) \geq f(x),\ \forall x \in D$ have been proposed [2, 6, 7, 15, 16].

An obvious way to solve the concave programming problem is a complete enumeration of the extreme points. Although most of the algorithms in the worst case will degenerate to complete inspection of all vertices of the polytope, this approach is computationally infeasible for large problems [12].

In this article, algorithms are specialized for solving the following problem, maximization of distance to the origin, a quadratic problem:

$$\begin{cases} \text{maximize } \|x\|^2, \\ \text{subject to } x \in D \end{cases} \tag{2}$$

where D is a full dimensional polytope.

2. Optimization methods

Firstly, let us concentrate on the local search and the cutting plane method. A local search with starting point x for (1) is relatively easy due to the method [4, 14]:

$$x^{k+1} = argmax\{\langle \nabla f(x^k), x\rangle \mid x \in D\}.$$

When $x^{k+1} = x^k$, then x^k is a local maximum of D.

Let a local maximum $y \in D$ be a vertex of the full dimensional polytope D. Following n edges at y, we find n points $y^1, y^2, ..., y^n$, which are the intersections of the edges with level set $\{x \mid f(x) = f(y)\}$. Then, hyper-plane $\{x \mid \langle c, x\rangle = \gamma\}$ that contain the points are built [9].

By the convexity of the objective function $f(\cdot)$ problem (2) is equivalent to:

$$\begin{cases} \text{maximize } \|x\|^2, \\ \text{subject to } x \in D, \ \ \langle c, x\rangle \geq \gamma \end{cases}$$

In other words, one cuts off a part of D, where values of function $f(\cdot)$ are less or equal than $f(y)$. The same procedure is then applied to the remaining part of the feasible set whenever this part is not empty.

However, despite such nice theoretical idea, this approach suffers, in practice, from the tailing off effect, i.e. cutting planes become closer or nearly parallel due to rounding errors so that they generate more and more local maxima. It remains the challenge in global search step: how to escape from a local maximum area?

Proposed methods are based on two sub-problems. The first one is the maximum value of (2) where D is a ball, noted. The second one is the maximum value of (2) where D is a box.

Lemma 1.

$$\begin{cases} \textit{maximize } \|x\|^2, \\ \textit{subject to } \|x - w\|^2 \leq r^2 \end{cases} \tag{3}$$

The optimal solution is $u = \left(1 + \frac{r}{\|w\|}\right) w$.

The largest ball inscribed into the polytope D is based on Murty et al. research works [11]. The radius of an inner ball is the minimal distance from its center x to the constraints of the feasibility domain. The center of the largest ball inscribed into D is the solution of the maximum value of the minimum radius of a point $x \in D$, with $\|a^i\| = 1$:

$$\begin{cases} \text{maximize } x_{n+1}, \\ \text{subject to } \left\langle a^i, x\right\rangle + x_{n+1} \leq b_i,\ i = 1, \dots, n \end{cases} \tag{4}$$

The second sub-problem is the maximum value of (2) where D is a box.

Lemma 2.

$$\begin{cases} \textit{maximize } \|x\|^2, \\ \textit{subject to } L_i \leq x_i \leq U_i, i = 1, \dots, n \end{cases} \tag{5}$$

The global optimum is $v = \left(\ max\{\mid L_i \mid, \mid U_i \mid\}\ \right), i = 1, \dots, n.$

In order to calculate the outer approximation, let U and L be the upper and lower bounds for each dimension for D. The domain is convex, so for all $i = 1, \dots, n:$

$$U_i = argmax\left\{\left\langle e^i, x\right\rangle \ :\ Ax \leq b\right\},\ L_i = argmin\left\{\left\langle e^i, x\right\rangle \ :\ Ax \leq b\right\},\ e^i = (0, \dots, 1, \dots, 0)^T$$

Both methods have the same process. One method uses an inner approximation (IA) with the largest inscribed ball. The second one is based on an inner and outer approximation (IOA) using largest inscribed ball and minimal enclosed box. We expose the main step of the two methods:

1. Find a candidate: IA resolves (4) then (3). IOA resolves (4) then (3), and (5).
2. Find a local maximum from the candidate.
3. Use cutting-plane until the domain is empty.
4. Return global maximum.

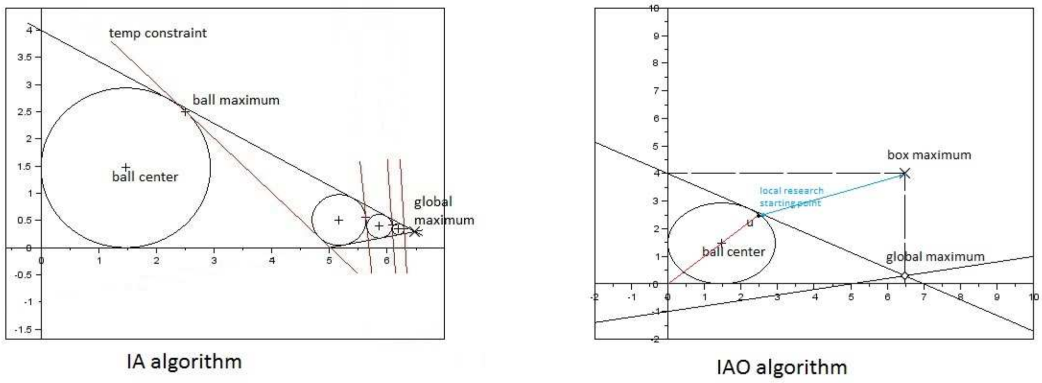


Figure 1: One iteration of IA and IOA algorithm.

3. Results and discussions

The IA and IAO algorithms have been tested on a bunch of convex maximization problems taken from "A collection of test problems for constrained global optimization algorithms" [5] (noted: TP plus the chapter of the test) and "An algorithm for maximizing a convex function over a simple set" [3] (noted: P plus the number of the test).

We present the numerical results in the table below and the meanings for all columns in the table follow: number of variables; number of local searches for IA; number of local searches for IOA; the best value found; the global optimal known value; the average computing time for IA in seconds; the average computing time for IOA in seconds.

Problem	n	IA LS	IOA LS	the best value	optimal value	time IA	time IOA
TP2.1	5	3	3	-17	-17	0.1	0.2
TP2.6	10	2	4	-39	-39	0.3	0.7
TP2.7.1	20	3	2	-394.7506	-394.7506	1.5	1.0
TP2.7.3	20	3	2	-8695.01193	-8695.01193	1.5	1.0
P4	2	1	1	42.0976	42.0976	0.1	0.1
P6	2	2	1	162	162	0.1	0.1
P11	100	2	1	1541089	1541089	6	2.5
P11	200	2	1	4150.41013	4150.41013	35	14.7

The best known solutions are found for all test problems considered in few local searches. Average computing time is calculated in the case of all conditions are checked at each iteration (full dimensional, active constraints, normal cone, etc.).

4. Conclusion

In this article, in order to find the global solution of a quadratic convex maximization problem, two algorithms are described. They are based on using the largest inscribed ball and the minimal enclosing box as an approximation for cutting-plane method. These methods are simple, quick and provide the global optimum.

Currently, we use box built on the orthonormal basis. In future work, a box with an other orthonormal set will be used (also named cuboid) thanks to Gram-Schmidt algorithm [8] in order to build the minimal box enclosing the domain D.

References

[1] H. P. Benson. Concave minimization: Theory, applications and algorithms. In *Handbook of global optimization*. Kluwer Academic, Dordrecht/Boston/London, 1995.

[2] M. Dür, R. Horst, and M. Locatelli. Necessary and sufficient global optimality conditions for convex maximization revisited. *J. Math. Anal. Appl.*, 217(2):637–649, 1998.

[3] R. Enhbat. An algorithm for maximizing a convex function over a simple set. *Journal of Global Optimization*, 8(4):379–391, 1996.

[4] R. Enkhbat. *Algorithm for global maximization of convex functions over sets of special structures*. Ph.D. thesis, Irkutsk State University., 1991.

[5] C. A. Floudas and P. M. Pardalos. *A collection of test problems for constrained global optimization algorithms*, volume 455 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1990.

[6] J.-B. Hiriart-Urruty. Conditions for global optimality. In *Handbook of global optimization*, volume 2 of *Nonconvex Optim. Appl.*, pages 1–26. Kluwer Acad. Publ., Dordrecht, 1995.

[7] J.-B. Hiriart-Urruty and Y.S. Ledyaev. A note on the characterization of the global maxima of a (tangentially) convex function over a convex set. *J. Convex Anal.*, 3(1):55–61, 1996.

[8] W. Hoffmann. Iterative algorithms for gram-schmidt orthogonalization. *Computing*, 41(4):335–348, 1989.

[9] R. Horst, P.M. Pardalos, and N.V. Thoai. *Introduction to global optimization*, volume 3 of *Nonconvex Optimization and its Applications*. Kluwer Academic Publishers, Dordrecht, 1995.

[10] R. Horst and H. Tuy. *Global optimization*. Springer-Verlag, Berlin, second edition, 1993. Deterministic approaches.

[11] K.G. Murty. Sphere methods for lp. In *Optimization for Decision Making*, pages 417–444. Springer, 2010.

[12] P.M. Pardalos and J.-B. Rosen. Methods for global concave minimization: A bibliographic survey. *Siam Review*, 28(3):367–379, 1986.

[13] R.T. Rockafellar. *Convex analysis*, volume 28. Princeton university press, 1997.

[14] A. S. Strekalovskii. *Elements of Nonconvex Optimization*. (in russian). NAUKA, Novosibirsk, 2003.

[15] A.S. Strekalovsky. Global optimality conditions for nonconvex optimization. *J. Global Optim.*, 12(4):415–434, 1998.

[16] I. Tsevendorj. On the conditions for global optimality. *J. Mong. Math. Soc.*, (2):58–61, 1998.

On the minimum number of simplices in a longest edge bisection refinement of a unit simplex*

G. Aparicio[1], L.G. Casado[2], E.M.T. Hendrix[3], I. García[3], and B.G.-Tóth[4]

[1] *Research Group TIC146: High Performance Computing - Algorithms, University of Almería (CeiA3), Spain,* guillermoaparicio@ual.es

[2] *Dept. Informatics, University of Almería (CeiA3), Spain,* leo@ual.es

[3] *Dept. Architectura de Computadores, Universidad de Málaga, Spain,* {Eligius,igarciaf}@uma.es

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

Abstract In simplicial based Global Optimization branch-and-bound methods on the unit n-simplex refinement by bisecting the longest edge leads to a binary search tree. Irregular sub-simplices generated in the refinement process may have more than one longest edge for a dimension higher than 3. The question is how to choose the longest edge to be bisected such that the number of simplices appearing in the binary tree is as small as possible.

The refinement usually selects the first longest edge and ends when the size of the sub-simplices generated in the refinement is smaller than a given accuracy ϵ. This research focuses on longest edge selection heuristics that aim to minimize the number of generated simplices in such a binary tree. New heuristics are presented which obtain the same or better results than previous studied heuristics by the authors. Preliminary results show that the heuristic achieving the smallest number of simplices depends on n.

Keywords: Regular Simplex, Longest Edge Bisection, Binary Tree

1. Introduction

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

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

We study the binary tree implicitly generated by the refinement of the n-simplex where the simplex division is defined by the Longest Edge Bisection rule (LEB) [1, 5]. We investigate the effect of the LEB rule on the number of generated simplices. The question is how to generate the smallest number of simplices in the binary tree by determining a new selection heuristic for longest edge bisection. Some heuristics, so-called LEB$_1$, LEB$_\alpha$, LEB$_C$ and LEB$_W$, were studied in [2]. Two new heuristics, LEB$_{W_2}$ and LEB$_M$ are studied here. Their results are compared with those obtained with preceding heuristics.

The paper is organized as follows. Section 2 shows the simplex refinement process based on the longest edge bisection. Section 3 describes the compared longest edge selection heuristics.

*This work has been funded by grants from the Spanish Ministry (TIN2008-01117 and TIN2012-37483) and Junta de Andalucía (P11-TIC-7176), in part financed by the European Regional Development Fund (ERDF), and also by the project ICT COST Action TD1207 (EU).

Preliminary results are provided in Section 4. Section 5 outlines an exhaustive longest edge analysis method which can help in the development of better heuristics. Finally, Section 6 concludes.

2. Simplex refinement using longest edge bisection

Figure 1 is an illustrative example of the binary tree generated by the refinement of a 2-simplex. The refinement ends when the width of the simplex S ($\omega(S)$), as the length of the longest edge, is smaller or equal than a given threshold ϵ.

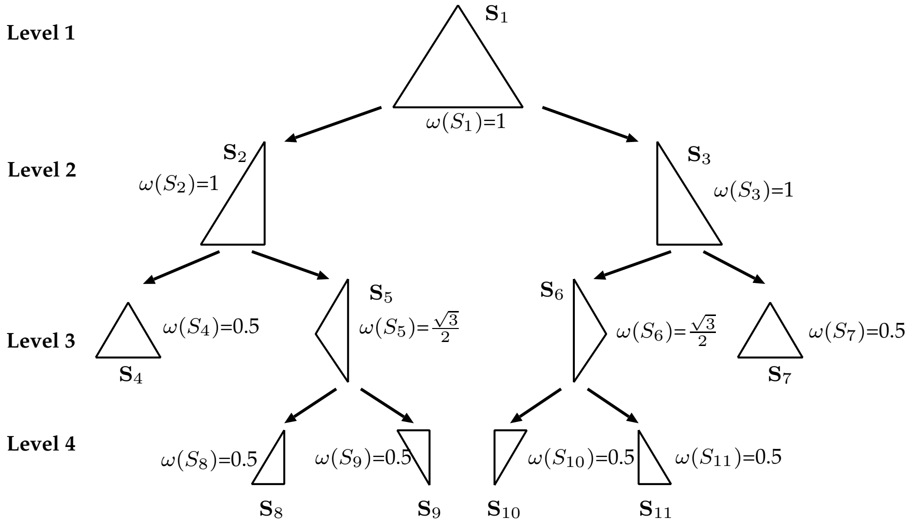


Figure 1: Binary tree generated by the refinement of a 2-simplex with $\epsilon = 0.5$

The longest-edge bisection algorithm [4] is based on splitting a simplex using the hyperplane that connects the mid point of the longest edge of a simplex with the opposite vertices. A 2-simplex is regular or it has just one longest edge.

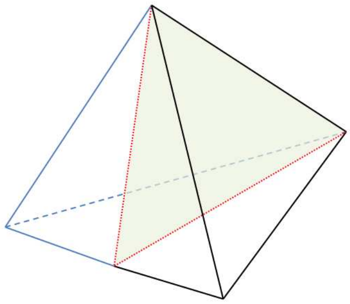


Figure 2: First longest-edge bisection on a regular 3-simplex

As illustrated by Fig. 2, an irregular 3-simplex in the refinement may have more than one longest edge.

The main purpose of the investigation is to select one of the longest edges making the binary tree as small as possible.

3. Heuristics for longest edge selection

The following heuristics are investigated, of which the last two are new.

First longest edge, LEB$_1$: A simple way to select a longest edge is to take the first one found. This rule is used as a benchmark in measuring performance.

Longest edge with the smallest angles, LEB$_\alpha$: As described in [3], for each vertex in a longest edge, the sum of the angles between edges at that vertex is determined and the longest edge having the smallest sum of angles is selected. LEB$_\alpha$ tries to avoid division of the smallest angles.

Longest edge with its midpoint furthest from the centroid, LEB$_C$: In this way the distance of the new halved edge to the new centroid is reduced.

Longest edge with vertices fewest involved in division, LEB$_W$: This heuristic, which works with weights w_i, tries to divide the longest edge with vertices participating in the smallest number of divisions. Every vertex v_i is given a weight $w_i \in \{0, \ldots, n-1\}$ during the refinement, which is increased if v_i is connected to a halved edge. Initially, vertices have weight 0. A new vertex obtains a weight of 1 because it is connected to the smallest edge. The weight of v_i is increased by one, if $\{v_i, v_j\}$ is split for any v_j. In that case, v_i obtains a value $w_i := (w_i + 1) \mod n$. LEB$_W$ selects the longest edge of which the sum of the weights of its vertices has the smallest value.

Combining LEB$_W$ with a second criterion, LEB$_{W_2}$: Here a second criteria is added to reduce the cases where there are several longest edges matching the weight requirements. For the first criterion, weights are calculated as in LEB$_W$, but now the new vertex has the weight of the vertex being substituted plus one. As second criterion the oldest longest edge from those having the smallest sum of weights at its vertices is selected.

Longest edge with its midpoint furthest from the other vertices, LEB$_M$. Edges at a new vertex will be the same for both sub-simplices. Therefore, this heuristic promotes more compact sub-simplices.

There could be several longest edges with the same value for a given heuristic. In such cases, the first longest edge with the best value is selected.

4. Preliminary Results

Table 1 shows the number of simplices generated by using each heuristic in the refinement for different values of n. Values for ϵ are taken small enough from $(1/2)^k$ to have a representative number of simplices in the binary tree and to have tractable problems.

Table 1: Number of simplices in a refinement using each heuristic.

	$n = 3$ and $\epsilon = 0.015625$	$n = 4$ and $\epsilon = 0.0625$	$n = 5$ and $\epsilon = 0.125$
LEB$_1$	2,114,687	1,577,047	2,510,297
LEB$_\alpha$	1,398,271	1,196,327	1,988,255
LEB$_C$	1,398,271	1,189,213	1,968,215
LEB$_W$	1,822,377	1,291,883	1,806,301
LEB$_{W_2}$	1,766,521	1,252,505	1,721,937
LEB$_M$	1,398,271	1,185,751	1,928,615

LEB$_\alpha$, LEB$_C$ and LEB$_M$ return the best result for n=3. Among them, the new proposed LEB$_M$ has the smallest computational requirements. LEB$_M$ provides the best result for n=4 as well. That is not the case for n=5, where the other new heuristic, LEB$_{W_2}$, is the best. The ambiguity of the heuristics increases with n and the selection of the first longest edge from the set satisfying the heuristic can produce the best results by coincidence. The reduction of the ambiguity in the heuristics needs a further research.

5. Smallest Binary Tree determination algorithm

In order to know the sequence of longest edges to be bisected to produce the smallest binary tree, an exhaustive search algorithm which tests every single longest edge bisection possibility is needed. This combinatorial optimization problem may require the use of high performance parallel computing due to its high computational cost. The search should avoid symmetries. If there are two longest edges whose bisection results in similar subtrees, just one of the trees should be calculated. Additionally, if the two descendants of a bisected simplex are similar, just the subtree of one of them should be calculated and its size counts twice. The result of this algorithm will be a set of sequences of indices indicating the selected longest edge at each bisection to obtain the smallest binary tree in the refinement process of an n-simplex.

Based on the results of this algorithm, the main objective of future research is to develop a new heuristic reproducing this longest edge selection criterion.

6. Conclusions

Selecting the first longest edge in the refinement of a regular n-simplex seems to be convenient, but appears to generate a larger binary tree than other heuristics for longest edge selection. New heuristics are investigated being the best ones in terms of computational complexity and achieved results for different values of n. Reduction of the ambiguity of the heuristics deserves further research. An algorithm to determine the optimal longest edge selection is outlined. Its results will help in the development of more effective heuristics.

References

[1] A. Adler. On the Bisection Method for Triangles. *Mathematics of Computation*, 40(162):571–574, 1983. DOI: 10.1090/S0025-5718-1983-0689473-5.

[2] G. Aparicio, L.G. Casado, E.M.T. Hendrix, B.G.-Tóth, and I. García. Heuristics to reduce the number of simplices in longest edge bisection refinement of a regular n-simplex. *International Conference on Computational Science and Its Applications*, Submitted.

[3] G. Aparicio, L.G. Casado, E.M.T. Hendrix, I. García, and B.G.-Tóth. On computational aspects of a regular n-simplex bisection. In *Proceedings of the 2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, pages 513–518. IEEE, Compiegne, France, October 2013. DOI: 10.1109/3PGCIC.2013.88.

[4] A. Hannukainen, S. Korotov, and M. Křížek. On numerical regularity of the face-to-face longest-edge bisection algorithm for tetrahedral partitions. *Science of Computer Programming*, 90:34–41, 2014. in press, DOI: 10.1016/j.scico.2013.05.002.

[5] R. Horst. On generalized bisection of n-simplices. *Mathematics of Computation*, 66(218):691–698, 1997. DOI: 10.1090/S0025-5718-97-00809-0.

Narrowing the difficulty gap for the Celis-Dennis-Tapia problem

Immanuel M. Bomze[1] and Michael L. Overton[2]

[1]*Department of Statistics and Operations Research, University of Vienna, Austria,* immanuel.bomze@univie.ac.at

[2]*Courant Institute of Mathematical Sciences, New York University, U.S.A.*

Abstract We study the *Celis-Dennis-Tapia (CDT) problem*: minimize a non-convex quadratic function over the intersection of two ellipsoids. In contrast to the well-studied trust region problem where the feasible set is just one ellipsoid, the CDT problem is not yet fully understood. Our main objective here is to narrow the difficulty gap that occurs when the Hessian of the Lagrangian is not positive-semidefinite at all Karush-Kuhn-Tucker points. We prove new sufficient and necessary conditions both for local and global optimality, based on copositivity, giving a complete characterization in the degenerate case.

Keywords: copositive matrices, global optimality conditions, non-convex optimization, polynomial optimization, trust region problem

1. Introduction

We study the *Celis-Dennis-Tapia (CDT) problem* [2]: minimize a non-convex quadratic function over the intersection of two ellipsoids. This problem is a natural extension of the well-studied *trust region problem* [3] in which there is just one ellipsoidal constraint. Such problems arise quite naturally in iterative non-linear optimization procedures where in one iteration step, the objective and the constraints are approximated by quadratic models. However, while any trust region problem can be solved both in theory and in practice quite efficiently, the additional constraint makes the CDT problem substantially more challenging. Many articles have treated the analysis of this and related problems, for references see [1].

After scaling the constraints and an affine transformation, we can reduce any CDT problem to the following form:

$$\left.\begin{array}{rcl} z^* & := & \min\left\{f(\mathbf{x}) : r(\mathbf{x}) \le 0 \text{ and } s(\mathbf{x}) \le 0\right\}, \text{ with} \\ f(\mathbf{x}) & := & \frac{1}{2}\mathbf{x}^\top \mathsf{Q}\mathbf{x} + \mathsf{q}^\top\mathbf{x} \\ r(\mathbf{x}) & := & \frac{1}{2}(\mathbf{x}^\top\mathbf{x} - 1) \le 0 \quad \text{and} \\ s(\mathbf{x}) & := & \frac{1}{2}(\mathbf{x}^\top \mathsf{A}\mathsf{A}^\top\mathbf{x} - 2\mathbf{a}^\top\mathsf{A}^\top\mathbf{x} + \|\mathbf{a}\|^2 - 1) \le 0\,. \end{array}\right\} \qquad (1)$$

where Q is a real symmetric $n \times n$ matrix which is not positive-semidefinite (psd), A is an $n \times m$ matrix with full row rank n while $\mathsf{q} \in \mathbb{R}^n$ and $\mathbf{a} \in \mathbb{R}^m$.

The gradients at a point $\bar{\mathbf{x}}$ feasible to the CDT-problem (1) read

$$\bar{\mathsf{g}} := \nabla f(\bar{\mathbf{x}}) = \mathsf{Q}\bar{\mathbf{x}} + \mathsf{q}\,, \quad \bar{\mathbf{x}} = \nabla r(\bar{\mathbf{x}})\,, \quad \text{and} \quad \bar{\mathbf{y}} := \nabla s(\bar{\mathbf{x}}) = \mathsf{A}\mathsf{A}^\top\bar{\mathbf{x}} - \mathsf{A}\mathbf{a}\,.$$

To avoid cases where the feasible set is empty or consists only of a single point, we assume Slater's condition: there exists $\hat{\mathbf{x}} \in \mathbb{R}^n$ such that $\max\left\{\|\hat{\mathbf{x}}\|, \|\mathsf{A}^\top\hat{\mathbf{x}} - \mathbf{a}\|\right\} < 1$. This can be checked by solving a convex trust region problem.

Consider the following two trust region problems:

$$\min\left\{f(\mathbf{x}) : \|\mathbf{x}\| \le 1\right\} \quad \text{and} \quad \min\left\{f(\mathbf{x}) : \|\mathsf{A}^\top\mathbf{x} - \mathbf{a}\| \le 1\right\}. \qquad (2)$$

Any *global solution* to either of the trust region problems (2) that is also feasible for the other one constitutes a global solution to the CDT problem (1). Moreover, any *local solution* $\bar{\mathbf{x}}$ to (1) where at most one of the constraints is binding, i.e. which satisfies $\min\left\{\|\bar{\mathbf{x}}\|, \|\mathsf{A}^\top\bar{\mathbf{x}} - \mathbf{a}\|\right\} < 1$, is necessarily a local solution to one of the trust region problems (2), and we know that there can be at most one local, non-global solution to a trust region problem [4].

This leaves us with only one problematic region of the feasible set, namely

$$B := \left\{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| = 1 \text{ and } \|\mathsf{A}^\top\mathbf{x} - \mathbf{a}\| = 1\right\} = \{\mathbf{x} \in \mathbb{R}^n : r(\mathbf{x}) = s(\mathbf{x}) = 0\}$$

where both constraints are binding. We focus on this case in what follows.

Our main objective in this study is to narrow the so-called difficulty gap. As long as the Hessian $\mathsf{H}(\bar{u}, \bar{v}) := \mathsf{Q} + \bar{u}\,\mathsf{I}_n + \bar{v}\,\mathsf{A}\mathsf{A}^\top$ of the Lagrangian is psd at some Karush-Kuhn-Tucker (KKT) point $\bar{\mathbf{x}}$ with multipliers $(\bar{u}, \bar{v})$, trust region problem methods can be employed, so these cases are considered easy. However, it may happen that the Hessian of the Lagrangian is not psd at all KKT points [5], and this phenomenon is usually called "difficulty gap".

2. Optimality conditions and copositivity

To discuss local and global optimality conditions, we first need the *linearized tangent cone* at a (1)-feasible $\bar{\mathbf{x}}$, i.e.

$$\Gamma(\bar{\mathbf{x}}) := \left\{ \begin{array}{ll} \left\{\mathsf{d} \in \mathbb{R}^n : \bar{\mathbf{x}}^\top\mathsf{d} \leq 0 \text{ and } \bar{\mathbf{y}}^\top\mathsf{d} \leq 0\right\} & \text{if } \bar{\mathbf{x}} \in B \\ \left\{\mathsf{d} \in \mathbb{R}^n : \bar{\mathbf{x}}^\top\mathsf{d} \leq 0\right\} & \text{if } s(\bar{\mathbf{x}}) < r(\bar{\mathbf{x}}) = 0 \\ \left\{\mathsf{d} \in \mathbb{R}^n : \bar{\mathbf{y}}^\top\mathsf{d} \leq 0\right\} & \text{if } r(\bar{\mathbf{x}}) < s(\bar{\mathbf{x}}) = 0 \\ \mathbb{R}^n & \text{if } \max\{r(\mathbf{x}), s(\bar{\mathbf{x}})\} < 0 \end{array} \right\}.$$

If $\bar{\mathbf{x}}$ is locally optimal for the CDT problem (1), Slater's condition implies the local first-order condition

$$\bar{\mathsf{g}}^\top\mathsf{d} \geq 0 \quad \text{for all } \mathsf{d} \in \Gamma(\bar{\mathbf{x}})$$

which is equivalent to $\bar{\mathbf{x}}$ being a KKT point, i.e., a feasible point satisfying the KKT conditions

$$\bar{\mathsf{g}} + \bar{u}\bar{\mathbf{x}} + \bar{v}\bar{\mathbf{y}} = \mathsf{o} \quad \text{and} \quad \bar{u}r(\bar{\mathbf{x}}) = \bar{v}s(\bar{\mathbf{x}}) = 0 \tag{3}$$

for some (not necessarily unique) multiplier pair $(\bar{u}, \bar{v}) \in \mathbb{R}^2_+$. We refer to $(\bar{\mathbf{x}}; \bar{u}, \bar{v})$ as a KKT triple. Clearly, the second condition in (3) holds automatically when $\bar{\mathbf{x}} \in B$.

A KKT point $\bar{\mathbf{x}} \in B$ is *nondegenerate* if the constraint gradients are linearly independent and therefore the multiplier pair is unique. In the degenerate case where $\bar{\mathbf{y}} = \alpha\bar{\mathbf{x}}$ for some $\alpha > 0$, we have $\bar{\mathsf{g}} = -(\bar{u} + \alpha\bar{v})\bar{\mathbf{x}}$. Then $(\tilde{u}, 0) := (\|\bar{\mathsf{g}}\|, 0)$ and $(0, \tilde{v}) := (0, \frac{\|\bar{\mathsf{g}}\|}{\alpha})$ are both KKT multiplier pairs for $\bar{\mathbf{x}}$, as are all pairs in their convex hull, which is a line segment in $\mathbb{R}^2_+$ of the form

$$(\bar{u}(t), \bar{v}(t)) := (t\tilde{u}, (1-t)\tilde{v}) : \ \ t \in [0, 1]. \tag{4}$$

Because of the nonnegativity condition, no other multiplier pairs for $\bar{\mathbf{x}}$ exist. Interestingly enough, the degenerate case allows for no difficulty gap, at least for some $t \in [0, 1]$; see below.

Next, we need to introduce the *reduced* polyhedral tangent cone comprising all feasible directions along which no first-order change in the objective is possible:

$$\Gamma_{\text{red}}(\bar{\mathbf{x}}) := \left\{\mathsf{d} \in \Gamma(\bar{\mathbf{x}}) : \bar{\mathsf{g}}^\top\mathsf{d} = 0\right\}.$$

An important property of symmetric matrices is that of *copositivity*. For a given cone $\Gamma \subset \mathbb{R}^n$, recall that a symmetric $n \times n$ matrix S is said to be Γ–*copositive* if and only if

$$\mathsf{d}^\top\mathsf{S}\mathsf{d} \geq 0 \quad \text{for all } \mathsf{d} \in \Gamma,$$

i.e., if S generates a quadratic form taking no negative values over the cone Γ. Therefore, any psd matrix S is Γ-copositive, regardless of Γ, but not conversely. A matrix S is said to be *strictly* Γ*–copositive* if and only if

$$\mathsf{d}^\top \mathsf{S} \mathsf{d} > 0 \quad \text{for all } \mathsf{d} \in \Gamma \setminus \{\mathsf{o}\} \; .$$

Any positive-definite matrix is strictly Γ-copositive, but again, not conversely.

To formulate a hierarchy of global and local optimality conditions, it is convenient to denote by $\psi(\mathsf{M})$ the number of negative eigenvalues of a symmetric matrix M, counting their multiplicities. Let $(\bar{\mathbf{x}}; \bar{u}, \bar{v})$ be a nondegenerate KKT triple for (1). Then the following implications hold (all proofs can be found in [1, Section 2]):

$$\boxed{\begin{array}{ll} & \mathsf{H}(\bar{u},\bar{v}) \text{ is positive-semidefinite} \\ \Rightarrow & \mathsf{H}(\bar{u},\bar{v}) \text{ is } \Gamma(\bar{\mathbf{x}})\text{-copositive} \\ \Rightarrow & \bar{\mathbf{x}} \text{ solves CDT globally and } \psi(\mathsf{H}(\bar{u},\bar{v})) \leq 1\,; \\ \hline & \mathsf{H}(\bar{u},\bar{v}) \text{ is strictly } \Gamma_{\text{red}}(\bar{\mathbf{x}})\text{-copositive} \\ \Rightarrow & \bar{\mathbf{x}} \text{ solves CDT locally} \\ \Rightarrow & \mathsf{H}(\bar{u},\bar{v}) \text{ is } \Gamma_{\text{red}}(\bar{\mathbf{x}})\text{-copositive} \\ \Rightarrow & \psi(\mathsf{H}(\bar{u},\bar{v})) \leq 2\,. \end{array}}$$

In general, checking Γ-copositivity of a matrix H is NP-hard. However, for $\Gamma = \Gamma(\bar{\mathbf{x}})$ here, this question can be solved in polynomial time even if $\mathsf{H}(\bar{u},\bar{v})$ fails to be psd [1, Section 3]. Therefore the difficulty gap is narrowed.

Still stronger results hold in the degenerate case. Let $\bar{\mathbf{x}}$ be a degenerate KKT point for (1), with the line segment of multiplier pairs in $\mathbb{R}^2_+$ given in (4). Then the following equivalence and implications hold (again, all proofs are in [1, Section 2]):

$$\boxed{\begin{array}{ll} & \mathsf{H}(\bar{u}(t),\bar{v}(t)) \text{ is positive-semidefinite for some } t \in [0,1] \\ \Leftrightarrow & \bar{\mathbf{x}} \text{ solves CDT globally;} \\ \hline & \mathsf{H}(\bar{u}(t),\bar{v}(t)) \text{ is strictly } \Gamma_{\text{red}}(\bar{\mathbf{x}})\text{-copositive for some } t \in [0,1] \\ \Rightarrow & \bar{\mathbf{x}} \text{ solves CDT locally} \\ \Rightarrow & \mathsf{H}(\bar{u}(t),\bar{v}(t)) \text{ is } \Gamma_{\text{red}}(\bar{\mathbf{x}})\text{-copositive for some } t \in [0,1] \\ \Rightarrow & \psi(\mathsf{H}(\bar{u}(t),\bar{v}(t))) \leq 1 \; \text{ for some } t \in [0,1]\,. \end{array}}$$

3. Experiments

We conducted some numerical experiments to observe how often the various cases occurred on randomly generated CDT problems. The entries of Q, A, q and $\mathbf{a}$ were independently generated from the normal distribution, and Q was replaced by its real symmetric part; then a vector $\tilde{\mathbf{x}}$ was generated in the same way, normalized to have length one, and then A and $\mathbf{a}$ were scaled by $1/\|\mathsf{A}^\top \tilde{\mathbf{x}} - \mathbf{a}\|$, guaranteeing the existence of at least one feasible point and therefore, generically, that the Slater condition holds. The vector $\tilde{\mathbf{x}}$ was then discarded and a candidate $\bar{\mathbf{x}}$ for the global solution of each problem obtained by using BFGS to minimize the exact penalty function $p(\mathbf{x}) = f(\mathbf{x}) + \rho \max(r(\mathbf{x}), 0) + \rho \max(s(\mathbf{x}), 0)$, for some $\rho > 0$ that was increased as needed to ensure feasibility, in a (tenfold) multistart fashion. In by far the majority of cases, global optimality was confirmed, and in all except one of 70,000 tests at least local optimality was confirmed. Details are given in Table 1.

# binding	2	2	2	1	1	0
condition	psd	Γ-copos	$\Gamma_{\rm red}$-copos	psd	$\Gamma_{\rm red}$-copos	psd
$n=2$	2591	56	215	6455	488	194
$n=3$	3618	50	448	5572	296	16
$n=4$	4214	39	418	5151	178	0
$n=5$	4396	40	409	5043	112	0
$n=6$	4582	26	361	4954	77	0
$n=7$	4646	18	291	4985	60	0
$n=8$	4688	14	244	5007	47	0

Table 1: Number of times the psd and copositivity conditions on $\mathsf{H}(\bar{u},\bar{v})$ occur at computed minimizers $\bar{\mathsf{x}}$ of 10,000 randomly generated instances of feasible CDT problems for each n from 2 to 8, categorized by the number of binding constraints at $\bar{\mathsf{x}}$. By randomness, no degeneracy occurred.

4. Conclusion

We provide new copositivity-based optimality conditions for the CDT-problem, thereby reducing the difficulty gap. Table 1 shows that by far the most common scenario is that $\mathsf{H}(\bar{u},\bar{v})$ is psd, but with positive probability it is $\Gamma(\bar{\mathsf{x}})$-copositive but not psd. The second most likely scenario with two binding constraints is that neither condition holds, indicating that there is still scope for further work to close the difficulty gap in characterizing global optimality.

References

[1] Immanuel M. Bomze and Michael L. Overton. Narrowing the difficulty gap for the Celis-Dennis-Tapia problem. Isaac-Newton-Institute Preprint Series NI13063-POP, University of Cambridge, submitted (2013).

[2] Maria Rosa Celis, John E. Dennis Jr., and Richard A. Tapia. A trust region strategy for nonlinear equality constrained optimization. In *Numerical Optimization, 1984 (Boulder, Colo., 1984)*, pages 71–82. SIAM, Philadelphia, PA, 1985.

[3] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *Trust-region Methods*. MPS/SIAM Series on Optimization. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.

[4] José Mario Martínez. Local minimizers of quadratic functions on Euclidean balls and spheres. *SIAM J. Optim.*, 4(1):159–176, 1994.

[5] Ya-Xiang Yuan. On a subproblem of trust region algorithms for constrained optimization. *Math. Program.*, 47(1, (Ser. A)):53–63, 1990.

Parallel Decomposition Methods for Nonconvex Optimization Recent Advances and New Directions

Ivo Nowak

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

Abstract Most industrial optimization problems are sparse, and can be reformulated by smaller sub-problems, which are linked by coupling constraints. Typical examples are planning, control and design problems. In practice, parallel decomposition methods are sometimes the only possibility to compute high-quality solutions of large-scale optimization problems. However, efficient implementations may require expert knowledge and problem-specific tricks. Recently, there is renewed interest in making these methods accessible to general users by developing generic decomposition frameworks and modeling support. These efforts are still early in the development stages, and there is much room for improvements.

The purpose of this paper is two-fold. On one hand we show how nonconvex airline planning and control problems with several hundred millions of variables and constraints can be solved in reasonable time by parallel decomposition methods. On the other hand, we present a novel decomposition approach for nonconvex programming with large duality gaps based on a nonconvex master (global approximation) problem. The method can be applied to general nonconvex optimization problems since an arbitrary sub-solver can be used. In particular, it can be applied to black-box simulation-based design optimization problems if a derivative free optimization method is used as a sub-problem solver.

Keywords: decomposition, nonconvex optimization

1. Introduction

Parallel Decomposition (in short PD) is the process of taking a model and breaking it into smaller sub-problems, which are solved in parallel. Parallel computing is getting more and more important, since clock speed is improving very slowly and in the future computers will have a huge number of cores [4].

PD is a very general approach that can be applied to convex optimization, as well as nonconvex optimization and discrete optimization. It can take place along a number of dimensions, like time-windows, resources or system components. Preliminary experiments with decomposition methods for MINLPs were performed in [6] using the solver `LaGO`, which was partly integrated into `SCIP/MINLP` in [7]. A decomposition algorithm for nonconvex MINLPs is presented in [3].

2. Decomposition algorithms for convex relaxations

We consider a *block-separable* nonconvex optimization problem of the form

$$\min c(x), \;\; h(x) = 0, \;\; x \in X, \tag{1}$$

where $x = (x_k)_{k \in K}$, $x_k \in \mathbb{R}^{n_k}$, $c : \mathbb{R}^n \to \mathbb{R}$, $h : \mathbb{R}^n \to \mathbb{R}^m$ are linear functions, and $X = \prod_{k \in K} X_k$ with $X_k \subset \mathbb{R}^{n_k}$. It is well-known that general sparse optimization problems can be re-formulated as block-separable problems by introducing auxiliary variables and additional constraints. A *convex relaxation* of (1) is defined by

$$\min c(x),\ \ h(x) = 0,\ \ x_k \in \text{conv}\ (X_k),\ \ k \in K, \tag{2}$$

Problem (2) can be solved by four different decomposition algorithms: (i) *column generation*, (ii) *Lagrange decomposition*, (iii) *cutting plane methods* or (iv) *Frank-Wolfe decomposition*. All algorithms solve sub-problems of the form

$$\min s_k^T x_k,\ \ x_k \in X_k, \tag{3}$$

where $s_k \in \mathbb{R}^{n_k}$ is a search direction. In order to be efficient, a fast sub-solver for these sub-problems is necessary. Examples for fast sub-solvers are dynamic programming, shortest path, (truncated) branch-and-cut or local search.

The solution of (2) can be used as a starting point for solving (1). The quality of this starting point depends strongly on the duality gap $\Gamma := \text{val}\,(1) - \text{val}\,(2)$.

3. Decomposition methods for airline planning&control

A simplified arc-based formulation of an airline scheduling problem is given by problem (1), where the feasible set X_k consists of all paths of a sub-network $\mathcal{N}_k$ fulfilling resource constraints. Most of the resource constraints are linear, but some of them are nonlinear. For the crew roster or pairing problem k represent a crew member or a group of crew members and x_k represents a roster or a pairing consisting of duties and transports (typically for one month).

Table 1 shows the size of two crew scheduling instances, which were presented in [5]. The biggest problem is a roster optimization problem with more than 700 million arc-variables, which was solved in less than 24 hours. In this case, a single sub-problem has more than 200.000 arc-variables. In both cases no transports were considered. Scheduling problems with transports are even much bigger.

Table 1: Size of crew scheduling instances

instance	*subproblems*	*nodes*	*arcs*	*vertical constraints*
pairing	2	180.882	1.909.212	30.822
roster	3.012	62.607.432	772.927.392	~500.000

An airline scheduling problem (1) is solved by column generation. The algorithm updates in each iteration a set of trial points $\hat{X}_k \subset X_k$ by alternately solving a restricted (*inner approximation*) master problem and sub-problems (3) regarding reduced cost directions. The master problem can be formulated as (Dantzig-Wolfe reformulation)

$$\min c(x(z)),\ \ h(x(z)) = 0,\ \ z_k \in \Delta_k,\ \ k \in K, \tag{4}$$

where $x(z) = (x_k(z_k))_{k \in K}$ with $x_k(z_k) := \sum_{v \in \hat{X}_k} z_{k,v} \cdot v$ and $\Delta_k \subset \mathbb{R}^{|\hat{X}_k|}$ is the standard simplex. The sub-problems (3) are solved by a dynamic programming based constrained shortest path solver. Because of the huge problem size, the search space is dynamically reduced [5]. The approach is called *reduce-and-generate*, since the main difficulty is the efficient search space reduction. Furthermore, in order to avoid many branching operations, rapid branching is used for solving the integer master problem. The idea of this method is to find a near-optimal solution of (1) by successively solving the *perturbed master problem* (4), in order to move the solution of the convex relaxation (2) towards the feasible set. The costs of columns (or constraints) of the perturbed master problem is reduced by $\texttt{const} \cdot z_{k,v}^2$ regarding to fractional solutions $(z_{k,v})_{v \in \hat{X}_k}$ of (4) such that the objective value is changed as less as possible. Rapid branching works well if the duality gap is small, see [2] for details. A simplified version of the reduce-and-generate approach is described by the following algorithm:

```
Reduce-and-generate:
1. Initialize the search space by reducing the network and adding pruning constraints.
2. Find trial points $\hat{X}_k, k \in K$, in the reduced search space.
3. Compute a dual solution of the master problem (4) using the bundle algorithm.
4. Generate columns by solving (3) regarding reduced cost directions.
5. Add the columns to $\hat{X}_k, k \in K$.
6. If new columns have been generated, goto 3.
7. If the search space is not completely open, increase the search space, and goto 2.
8. For all time-windows:
     Fix some binary variables by rapid branching and
     generate columns by repeating steps 3 - 5, if neccessary.
```

4. A new decomposition approach for design optimization problems with large duality gaps

The reduce-and-generate approach outlined in the previous section can only be applied if the duality gap is small. Experiments with MINLPs in [6] and with MIPs in [1] indicate that the duality gap is often not small, and closing the gap with a branch-cut-and-price algorithm may lead to many branching operations.

In this section we describe some ideas for a new decomposition approach which uses a *nonconvex* piecewise quadratic master problem, instead of a linear master problem. We consider a block-separable reformulation of a *design* optimization problem (with a large duality gap):

$$\min F(x), \;\; h(x) = 0, \;\; x_k \in X_k, \;\; k \in K, \tag{5}$$

where $F(x) := \sum_{k\in K} F_k(x_k)$ and $F_k(x_k) := \min\{f(x_k, y_k) \mid y_k \in Y_k(x_k)\}$, $X_k \subset \mathbb{R}^{n_{X,k}}$ is the feasible set of *design* variables and $Y_k(x_k) \subset \mathbb{R}^{n_{Y,k}}$ is the feasible set of *state* variables. For many engineering design problems defined by partial differential equations we have $n_{Y,k} >> n_{X,k}$.

Let $D_k := \{D_{kj}\}_{j\in J_k}$ be a polyhedral partition of X_k, i.e. $X_k \subseteq \bigcup_{j\in J_k} D_{kj}$. Define a piecewise quadratic *global approximation* (or *response surface*) $F_k^{D_k}$ of F_k which interpolates (or approximates) F_k on a set of trial points $\hat{X}_k \subset X_k, k \in K$, by quadratic functions over $D_{kj}, j \in J_k$.

The following decomposition algorithm computes solutions of (5) by alternately solving sub-problems and computing solutions of the nonconvex master problem

$$\min F^D(x), \;\; h(x) = 0, \;\; x_k \in X_k, \;\; k \in K, \tag{6}$$

where $D := \{D_k\}_{k\in K}$ and $F^D(x) := \sum_{k\in K} F^{D_k}(x_k)$.

```
Approximate-and-generate:
1. Initialize the sets $\hat{X}_k, k \in K$, of trial points by a start heuristic.
2. Compute $D$ and $F^D$ regarding $\hat{X}_k, k \in K$.
3. Compute a set $\mathcal{X}$ of solutions of (6) and (5).
4. Update the sets $\hat{X}_k, k \in K$, using $\mathcal{X}$ and sub-problem solutions.
5. If the stopping criterion is not fulfilled, go to 2.
```

A set $\mathcal{X}$ of solutions of (6) and (5) can be computed by the following dynamic programming approach. Denote by $\mathcal{G}(\mathcal{N}, \mathcal{A})$ the *component graph* (*meta-structure*) of (5) where the nodes $\mathcal{N}$ of the graph are the sub-regions $\{D_{kj}\}_{j\in J_k, k\in K}$ and the arcs $\mathcal{A}$ are defined by the support of the coupling constraints $h(x) = 0$. Let $S = \{D_{kj_k}\}_{k\in K}$ be a path in $\mathcal{G}(\mathcal{N}, \mathcal{A})$ and let $\mathcal{S}^*$ be the set of all non-dominated paths of $\mathcal{G}(\mathcal{N}, \mathcal{A})$, which can be computed by implicit enumeration using dynamic programming. Then for all $S \in \mathcal{S}^*$ a local solution $\hat{x}_S$ of (6) can be computed by minimizing F^D over S. A local solution of (5) can be computed by local search starting from $\hat{x}_S$.

Step 4 of the algorithm can be performed by a deterministic global optimization approach: Generate a polyhedral outer-approximation (OA) of (6) with as few cuts $F_k^{D_k}(x_k) \geq s_{kj}^T x_k + b_{kj}$ over D_{kj} as possible using a branch-and-cut algorithm, as shown in Figure 1. Then correct the cuts of the OA by solving

$$b_{kj}^* = \min\{F_k(x_k) - s_{kj}^T x_k \mid x_k \in X_k \cap D_{kj}\}, \tag{7}$$

using a branch-and-cut algorithm, e.g. by `SCIP/MINLP` [7].

Another possibility for solving sub-problems (7) is to use a global search heuristic, e.g. a derivative free optimization method.

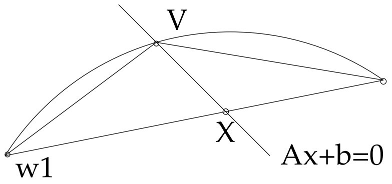


Figure 1: A nonconvex polyhedral outer-approximation conv $(\{w1, v\}) \cup$ conv $(\{v, w2\})$ with zero duality gap, where x is the solution of the convex relaxation (2).

5. Summary

This paper describes a parallel decomposition approach for solving huge nonconvex optimization problems with hundred millions of variables. Furthermore, a new decomposition algorithm for general nonconvex design problems with large duality gaps is proposed.

References

[1] M. Bergner, A. Caprara, F. Furini, M.E. Lübbecke, E. Malaguti, and E. Traversi. Partial convexification of general MIPs by Dantzig-Wolfe reformulation. In O. Günlük and G.J. Woeginger, editors, *Integer Programming and Combinatorial Optimization*, volume 6655 of *Lect. Notes Comput. Sci.*, pages 39–51, Berlin, 2011. Springer.

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

[3] A. Khajavirad and J.J. Michalek. A determinstic lagrangian-based global optimization approach for quasiseparable nonconvex mixed-integer nonlinear programs. *ASME Journal of Mechanical Design*, 131:1–8, 2009.

[4] T. Koch, T. Ralphs, and Y. Shinano. What could a million cores do to solve integer programs? *Mathematical Methods of Operations Research*, 76:67–93, 2012.

[5] I. Nowak. Reducing airline costs via global and robust optimization. China-Germany conference on "Mathematics and Industry", Peking, China, 2010. `https://www.lhsystems.com/solutions-services/airline-solutions-services/research-development/publications.html`

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

[7] S. Vigerske. *Decomposition of Multistage Stochastic Programs and a Constraint Integer Programming Approach to Mixed-Integer Nonlinear Programming*. PhD thesis, Humboldt University Berlin, 2012.

Global Optimization based on Contractor Programming

Jordan Ninin[1] and Gilles Chabert[2]

[1] *ENSTA-Bretagne, LabSTIC, IHSEV team, 2 rue Francois Verny, 29806 Brest, France* jordan.ninin@ensta-bretagne.fr

[2] *Ecole des Mines de Nantes, LINA, TASC team, 4 rue Alfred Kastler, 44300 Nantes, France* gilles.chabert@mines-nantes.fr

Abstract In this paper, we will present a general pattern based on contractor programming for designing a global optimization solver. This approach allows to solve problems with a wide variety of constraints. The complexity and the performance of the algorithm rely on the construction of contractors which characterize the feasible region.

Keywords: Global Optimization, Interval Arithmetic, Contractor Programming

1. Introduction

Considering sets in place of single points is not a common point of view in the Mathematical Programming communities. In contrast, this interpretation is the key concept in global optimization based on interval arithmetic and in constraint programming.

Faced to complexity and diversity of real-life problems, it is hard to find a general pattern or unique algorithm for solving all of them. Some algorithms deal with disjunctive constraints, others with dynamic constraints, others with uncertainties. Furthermore, when real-life problems merge different kinds of constraints, we often need to remove a part of the model just because our solvers do not accept it. Unfortunately, these cases are not rare.

In this talk, we will present a general pattern based on contractor programming that allows to handle a wide variety of problems. In Section 2, we recall the definitions related to interval arithmetic and contractor programming. Then, we present a short subset of some standard contractors and how we can combine and merge them. In Section 4, a global optimization algorithm based on contractors is detailed.

2. Definitions

Since the book of Moore in 1966 [9], many techniques have been developed based on interval arithmetic. More generally, all these techniques can be considered as Set-Membership Methods. These methods and algorithms do not consider single numerical values, or floating-point numbers, but manipulate sets. The interval arithmetic offers a solid theoretical basis to represent and to calculate with subsets of $\mathbb{R}^n$.

An interval is a closed connected subset of $\mathbb{R}$. A non-empty interval $[\mathbf{x}]$ can be represented by its endpoints: $[\mathbf{x}] = [\underline{x}, \overline{x}] = \{x : \underline{x} \leq x \leq \overline{x}\}$ where $\underline{x} \in \mathbb{R} \cup \{-\infty\}$, $\overline{x} \in \mathbb{R} \cup \{+\infty\}$ and $\underline{x} \leq \overline{x}$. The set of intervals will be denoted by $\mathbb{IR}$, and the set of n-dimensional interval vectors, also called *boxes*, will be denoted by $\mathbb{IR}^n$.

Definition 1. *Let $\mathbb{X} \subseteq \mathbb{R}^n$ be a feasible region.*
The operator $\mathcal{C}_{\mathbb{X}} : \mathbb{IR}^n \to \mathbb{IR}^n$ is a contractor for $\mathbb{X}$ if:

$$\forall [\boldsymbol{x}] \in \mathbb{IR}^n, \begin{cases} \mathcal{C}_{\mathbb{X}}([\boldsymbol{x}]) \subseteq [\boldsymbol{x}], & \textit{(contraction)} \\ \mathcal{C}_{\mathbb{X}}([\boldsymbol{x}]) \cap \mathbb{X} \supseteq [\boldsymbol{x}] \cap \mathbb{X}. & \textit{(completeness)} \end{cases}$$

The concept of contractor is very broad for integrating and interfacing mixed techniques [6]. Contractors are directly inspired form constraint programming. A contractor is defined for a feasible region $\mathbb{X}$ and its purpose is to eliminate a part of a domain which is not in $\mathbb{X}$.

Proposition 2. *The operator $\mathcal{C} : \mathbb{IR}^n \to \mathbb{IR}^n$ is a contractor for the equation $f(x) = 0$, if:*

$$\forall [\boldsymbol{x}] \in \mathbb{IR}^n, \begin{cases} \mathcal{C}([\boldsymbol{x}]) \subseteq [\boldsymbol{x}], \\ \forall x \in [\boldsymbol{x}],\ f(x) = 0 \Longrightarrow x \in \mathcal{C}([\boldsymbol{x}]). \end{cases}$$

The basic implementation of a contractor for a numerical constraint is the forward-backward algorithm, also called constraint propagation technique or FBBT or HC4-Revise [4, 8, 11]. This algorithm is the basic block of contractor programming.

All set operators can be extended to contractors. For example, the intersection of two contractors creates a contractor for the intersection of these two sets. In the same way, the hull of two contractors creates a contractor for the disjunction of these constraints.

Definition 3. *Let $\mathbb{X}$ and $\mathbb{Y} \subseteq \mathbb{R}^n$ be two feasible regions.*

$$\begin{aligned} \textit{Intersection:} \quad & (\mathcal{C}_{\mathbb{X}} \cap \mathcal{C}_{\mathbb{Y}})([\boldsymbol{x}]) = \mathcal{C}_{\mathbb{X}}([\boldsymbol{x}]) \cap \mathcal{C}_{\mathbb{Y}}([\boldsymbol{x}]) \\ \textit{Union:} \quad & (\mathcal{C}_{\mathbb{X}} \cup \mathcal{C}_{\mathbb{Y}})([\boldsymbol{x}]) = \mathcal{C}_{\mathbb{X}}([\boldsymbol{x}]) \cup \mathcal{C}_{\mathbb{Y}}([\boldsymbol{x}]) \\ \textit{Composition:} \quad & (\mathcal{C}_{\mathbb{X}} \circ \mathcal{C}_{\mathbb{Y}})([\boldsymbol{x}]) = \mathcal{C}_{\mathbb{X}}(\mathcal{C}_{\mathbb{Y}}([\boldsymbol{x}])) \\ \textit{Fixed Point:} \quad & \mathcal{C}^{\infty} = \mathcal{C} \circ \mathcal{C} \circ \mathcal{C} \circ \ldots \end{aligned}$$

Many known techniques can be cast into contractors. All linear relaxation techniques can be considered as contractors. Such a contractor is constructed by intersecting the input box with the polyhedral hull created by the linear relaxation. This intersection is obtained by solving at most $2n$ linear programs, see [2, 10] for details.

3. Non-conventional Contractors

The main interest of contractors is the ability to deal with constraints that are difficult to combine or to formulate mathematically. For example, if the variable corresponds to a position on a map, it is very simple to make the intersection of a given box with an area of a map, while it would have been cumbersome to describe this area by a mathematical equation.

Another common case is the possibility of corrupted data. If a set of constraints are based on physical data, it is not uncommon that some of this data are wrong. For example, only 80 % of constraints are acceptable, without knowing which ones. In this situation, the q-relaxed intersection of contractors can be applied to this problem:

Definition 4. *The q-relaxed intersection of m subsets $\mathbb{X}_1, \ldots, \mathbb{X}_m$ of $\mathbb{R}^n$ is the set of all $x \in \mathbb{R}^n$ which belong to at least $(m - q)$ $\mathbb{X}_i$. We denote it by $\mathbb{X}^{\{q\}} = \overset{\{q\}}{\bigcap} \mathbb{X}_i$.*

Since the q-relaxed intersection is a set operator, we can extend this notion to contractors: $\left(\bigcap^{\{q\}} \mathcal{C}_{\mathbb{X}_i}\right)([\mathbf{x}]) = \overset{\{q\}}{\bigcap} (\mathcal{C}_{\mathbb{X}_i}([\mathbf{x}]))$. This contractor allows to model the possibility of invalid constraints: it can also be used for robust optimization.

In [5], Carbonnel et al. found an algorithm with a complexity $\theta(nm^2)$ to compute a box which contains the q-relaxed intersection of m boxes of $\mathbb{R}^n$. This algorithm can be interpreted as an implementation of the q-relaxed intersection of m contractors.

Another possibility is to project a subset of $\mathbb{R}^n$ over one or more dimensions. For example, if a constraint needs to be satisfied for all values of a parameter in a given set, such as $\{x \in \mathbb{R}^n : \forall t \in \mathbb{X} \subseteq \mathbb{R}^m, g(x,t) \leq 0\}$, few solvers are available to deal with it. Another example is when a constraint needs to be satisfied for at least one value of the parameter, such as $\{x \in \mathbb{R}^n \ : \ \exists t \in \mathbb{X} \subseteq \mathbb{R}^m, g(x,t) \leq 0\}$.

Two operators are defined on contractors. The first one is the *projection-intersection* and the second one is the *projection-union*.

Definition 5. *Let $\mathbb{X} \subseteq \mathbb{R}^n$, $\mathbb{Y} \subseteq \mathbb{R}^m$, $\mathbb{Z} \subseteq \mathbb{R}^p$, with $\mathbb{Z} = \mathbb{X} \times \mathbb{Y}$. Let $\mathcal{C}$ be a contractor for the set $\mathbb{Z}$. We define $\mathcal{C}^{\cap\mathbb{Y}}$ the* Projection Intersection *of $\mathbb{Z}$ over $\mathbb{X}$ and $\mathcal{C}^{\cup\mathbb{Y}}$ its* Projection Union *by:*

$$\forall x \in \mathbb{R}^n, \ \begin{cases} \mathcal{C}^{\cap\mathbb{Y}}([\mathbf{x}]) = \bigcap_{y \in \mathbb{Y}} \pi_{\mathbb{X}} \left(\mathcal{C}([\mathbf{x}] \times \{y\})\right), \\ \mathcal{C}^{\cup\mathbb{Y}}([\mathbf{x}]) = \bigcup_{y \in \mathbb{Y}} \pi_{\mathbb{X}} \left(\mathcal{C}([\mathbf{x}] \times \{y\})\right). \end{cases}$$

with $\pi_{\mathbb{X}}$ the projection of $\mathbb{Z}$ over $\mathbb{X}$.

Proposition 6. *Let $\mathcal{C}$ be a contractor for a set $\mathbb{Z}$. $\mathcal{C}^{\cap\mathbb{Y}}$ is a contractor for the set $\mathbb{X} = \{x \ : \ \forall y \in \mathbb{Y}, (x,y) \in \mathbb{Z}\}$ and $\mathcal{C}^{\cup\mathbb{Y}}$ is a contractor for the set $\mathbb{X} = \{x \ : \ \exists y \in \mathbb{Y}, (x,y) \in \mathbb{Z}\}$.*

The Projection-Intersection contractor contracts each part of $[\mathbf{x}]$ which are contracted by $\mathcal{C}([\mathbf{x}] \times \{y\})$ for any $y \in \mathbb{Y}$. Indeed, each part $[\mathbf{a}]$ of $[\mathbf{x}]$, such as $\exists y \in \mathbb{Y}, \ ([\mathbf{a}], y) \notin \mathbb{Z}$, can be removed. Thus, each part $[\mathbf{b}]$ of $[\mathbf{x}]$, such as $\forall y \in \mathbb{Y}, \ ([\mathbf{b}], y) \in \mathbb{Z}$, is kept. A similar argument proves Proposition 6 for the Projection-Union contractor.

4. Global Optimization Algorithm

The implementations of all the previous contractors are available in our library IBEX [1, 3]. Thus, given a physical problem, the user can construct a contractor for the feasible region $\mathbb{X}$ of his problem. We denote this contractor $\mathcal{C}_{out}$. Moreover, using the counterparts of set-membership operators for contractors (cf. Definition 3), we can construct in the same way a contractor for the negation of $\mathbb{X}$. This contractor is denoted by $\mathcal{C}_{out}$. The only required mathematical expression is the objective function, f_{cost}.

Given a box $[\mathbf{x}] \in \mathbb{R}^n$, $\mathcal{C}_{out}([\mathbf{x}])$ removes from $[\mathbf{x}]$ a part that does not contain a feasible solution. In the same way, $\mathcal{C}_{in}([\mathbf{x}])$ removes from $[\mathbf{x}]$ parts which are entire feasible; i.e. $([\mathbf{x}]/\mathcal{C}_{in}([\mathbf{x}])) \subseteq \mathbb{X}$. Thus, $([\mathbf{x}]/\mathcal{C}_{in}([\mathbf{x}]))$ is a feasible subset and we can perform a global optimization without constraint on it. If this step succeeds, this set can be discarded: indeed, if a new best current solution is found, we save it and it is proved that this set does not contain a better solution; else it is directly proved that no better solution can be found in this set $([\mathbf{x}]/\mathcal{C}_{in}([\mathbf{x}]))$.

The following algorithm describes a simple implementation pattern for a global optimization solver based on contractors. This algorithm is inspired from the *SIVIA* Algorithm (Set-Inversion Via Interval Analysis), which is used to compute the feasible set in a domain [7].

The inputs are an initial domain $[\mathbf{x}] \in \mathbb{IR}^n$, $\mathcal{C}_{out}$ a contractor for $\mathbb{X}$, $\mathcal{C}_{in}$ a contractor for $\overline{\mathbb{X}}$ and f_{cost} an objective function. The outputs are $\tilde{f}$, the global minimum value found and $\tilde{x}$, a global minimum. A boolean variable b is added for each element of $\mathcal{L}$ to indicate if the element is included in the feasible region.

```
$(\tilde{x}, \tilde{f})$ = OptiCtc ($[\mathbf{x}], \mathcal{C}_{out}, \mathcal{C}_{in}, f_{cost}$):
  $\tilde{f} := +\infty$, denotes the current upper bound for the global minimum;
  $\mathcal{L} := \{([\mathbf{x}], false)\}$, initialization of the data structure of the stored elements;
  Let $\mathcal{C}_f$ a contractor based on the constraint $\{x \ : \ f_{cost}(x) \leq \tilde{f}\}$;
  Repeat until a stopping criterion is fulfilled:
    Extract from $\mathcal{L}$ an element $([\mathbf{y}], b)$,
    Bisect the considered box $[\mathbf{y}]$: $[\mathbf{y}_1], [\mathbf{y}_2]$,
    for $j = 1$ to $2$ :
      if ($b$ = false) then
```

```
        Contract $[\mathbf{y}_i]$ with $\mathcal{C}_{out} \cap \mathcal{C}_f$,
        $[\mathbf{y}_{tmp}] := [\mathbf{y}_i]$,
        Contract $[\mathbf{y}_i]$ with $\mathcal{C}_{in}$,
        Add ($[\mathbf{y}_i]$,false) in $\mathcal{L}$.
        $[\mathbf{y}_{tmp}] := [\mathbf{y}_{tmp}]/[\mathbf{y}_i]$,
      else
        Contract $[\mathbf{y}_i]$ with $\mathcal{C}_f$,
        $[\mathbf{y}_{tmp}] := [\mathbf{y}_i]$,
      Try to find the global optimum without constraint in $[\mathbf{y}_{tmp}]$,
      if the search succeeds in a limited time then
        Update $\tilde{f}$ and $\tilde{x}$.
      else
        Add ($[\mathbf{y}_{tmp}], true$) in $\mathcal{L}$.
  end.
```

Further refinements can improve the behavior of the algorithm, but we must keep in mind that most of time, the performances depend of the problem itself. With this algorithm, the improvements which can have a real and direct impact are in the implementation of the contractors and in the relevance of the model.

5. Summary

This paper introduces the use of the contractors for designing a global optimization algorithm. These concepts will be illustrated to minimize the consumption of two robots following nonlinear parametric trajectories and subject to two constraints: conflict avoidance and validation of at least 80% checkpoints.

References

[1] IBEX : *a C++ numerical library based on interval arithmetic and constraint programming*. http://www.emn.fr/z-info/ibex/.

[2] I. Araya, G. Trombettoni, and B. Neveu. *A contractor based on convex interval taylor*. In Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, Springer, 2012.

[3] I. Araya, G Trombettoni, B. Neveu and G Chabert. *Upper Bounding in Inner Regions for Global Optimization under Inequality Constraints*. Journal of Global Optimization, 2014, to appear.

[4] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Waechter, *Branching and bounds tightening techniques for non-convex MINLP*, Optimization Methods & Software, vol. 24, pp. 597-634, 2009.

[5] C. Carbonnel, G. Trombettoni, P. Vismara and G. Chabert, *Q-intersection algorithms for robust parameter estimation*, in submission.

[6] G. Chabert and L. Jaulin, *Contractor programming*, Artificial Intelligence, vol. 173, n. 11, pp. 1079-1100, 2009.

[7] L. Jaulin and E. Walter. *Set inversion via interval analysis for nonlinear bounded-error estimation*. Automatica, vol. 29, n.4, pp. 1053-1064, 1993.

[8] F. Messine, *Deterministic global optimization using interval constraint propagation techniques*, RAIRO-Operations Research, vol. 38, pp. 277-293, 2004.

[9] R.E. Moore, Interval Analysis, Prentice-Hall Inc., Englewood Cliffs, 1966.

[10] J. Ninin, P. Hansen, and F. Messine, *A reliable affine relaxation method for global optimization*. Cahier du GERAD, in submission.

[11] X.-H. Vu, D. Sam-Haroud, and B. Faltings, *Enhancing numerical constraint propagation using multiple inclusion representations*, Annals of Mathematics and Artificial Intelligence, vol. 55, pp. 295-354, 2009.

A tri-objective model for franchise expansion

J. Fernández[1], A.G. Arrondo[2], J.L. Redondo[3], and P.M. Ortigosa[2]

[1] *University of Murcia, Campus Universitario de Espinardo, 30100, Murcia, Spain* josefdez@um.es

[2] *University of Almería, CeiA3, C/ los Gallardos, s/n, 04120, La Cañada de San Urbano, Almería, Spain* agarrondo@ual.es, ortigosa@ual.es

[3] *University of Granada, ETS Ingenierías Informáticas y de Telecomunicaciones, C/ Periodista Daniel Saucedo Aranda, s/n, E-1807, Granada, Spain* jlredondo@ugr.es

Abstract A franchise wants to enlarge its presence in a given geographical region by opening one new facility. There are already some existing facilities (both of the franchise and not) offering the same service in the region. Both the franchisor (the owner of the franchise) and the franchisee (the actual owner of the new facility to be opened) have the same objective: maximize their own profit. However, the maximization of the profit obtained by the franchisor is in conflict with the maximization of the profit obtained by the franchisee. In addition to this, there is a third player to be taken into account, the existing franchise's facilities, which do not want to lose the market share they already have: they want the cannibalization to be minimized. In the model the demand is supposed to be fixed and concentrated in a discrete set of demand points, which split their buying power among all the facilities proportionally to the attraction they feel for them. The attraction (or utility) function of a customer towards a given facility depends on the distance between the customer and the facility, as well as on other characteristics of the facility which determine its quality. The location and the quality of the new facility are the variables of the problem. An evolutionary algorithm is proposed to obtain a discrete approximation of the efficient set (and the corresponding Pareto-front) of the resulting nonlinear tri-objective optimization problem. Some computational results are reported.

Keywords: Competitive location, Franchise, Cannibalization, Nonlinear multi-objective optimization, 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 to the market in the near future). Many competitive location models are available in the literature, see for instance the survey [4, 5]. However, the literature on multiobjective competitive location models is rather scarce. This is in part due to the fact that single-objective competitive location problems are difficult to solve, and considering more than one objective makes the problem near intractable. In this paper, a continuous nonlinear tri-objective optimization problem is introduced; each objective function alone leads to a single-objective global optimization problem. To our knowledge, this is the first tri-objective model ever proposed for the location of a competitive facility.

In particular we study the case of a franchise which wants to enlarge its presence in a given geographical region by opening one new facility. Both the franchisor (the owner of the franchise) and the franchisee (the actual owner of the new facility to be opened) have the same objective: maximize their own profit. However, those two objectives are in conflict. But there is a third player which complicates things even more. The existing franchise's facilities in the region do not want to lose the market share they capture before the expansion. Notice that the entrance of the new facility may also have a detrimental effect on the market shares of the existing facilities, and this cannibalization should also be minimized. This suggests to use a tri-objective model to obtain the efficient solutions for this problem, so that later on the franchisor, the new franchisee, and the franchisees of the existing facilities can agree in both location and design for the new facility, taking the corresponding economical implications of their selection into account.

2. The tri-objective model

In the model the *demand* is supposed to be fixed and concentrated at n demand points, whose locations p_i and *buying power* w_i are known. The location f_j and quality of the existing facilities is also known. Following Huff [3], we consider that demand points split their buying power among all the facilities proportionally to the *attraction* they feel for them. The attraction function of a customer towards a given facility depends on the distance between the customer and the facility, as well as on other characteristics of the facility which determine its *quality*.

The following notation will be used throughout this paper:

Indices

i index of demand points, $i = 1, \ldots, n$.
j index of existing facilities, $j = 1, \ldots, m$.

Variables

x location of the new facility, $x = (x_1, x_2)$.
α quality of the new facility ($\alpha > 0$).

Data

p_i location of the i-th demand point.
w_i demand (or buying power) at p_i.
f_j location of the j-th existing facility.
d_{ij} distance between p_i and f_j.
α_{ij} quality of f_j as perceived by p_i.
$g_i(\cdot)$ a non-negative non-decreasing function.
$\alpha_{ij}/g_i(d_{ij})$ attraction that p_i feels for f_j.
γ_i weight for the quality of the new facility as perceived by p_i.
k number of existing facilities that are part of the franchise (the first k of the m facilities are assumed in this category, $0 < k < m$).

Miscellaneous

d_{ix} distance between p_i and the new facility x.
$\gamma_i\alpha/g_i(d_{ix})$ attraction that p_i feels for x.

From the previous assumptions, the total market share attracted by the franchisor is

$$M(x,\alpha) = \sum_{i=1}^{n} w_i \frac{\dfrac{\gamma_i \alpha}{g_i(d_{ix})} + \displaystyle\sum_{j=1}^{k} \frac{\alpha_{ij}}{g_i(d_{ij})}}{\dfrac{\gamma_i \alpha}{g_i(d_{ix})} + \displaystyle\sum_{j=1}^{m} \frac{\alpha_{ij}}{g_i(d_{ij})}}.$$

We assume that the operating costs for the franchisor due to the new facility are fixed. In this way, the profit obtained by the franchisor is an increasing function of the market share that it captures. Thus, maximizing the profit obtained by the franchisor is equivalent to maximizing the market share that it captures. This will be the first objective of the problem.

The second objective of the problem is the maximization of the profit obtained by the franchisee, to be understood as the difference between the revenues obtained from the market share captured by the new facility minus its operational costs. The market share captured by the new facility (franchisee) is given by

$$m(x,\alpha) = \sum_{i=1}^{n} w_i \frac{\dfrac{\gamma_i \alpha}{g_i(d_{ix})}}{\dfrac{\gamma_i \alpha}{g_i(d_{ix})} + \displaystyle\sum_{j=1}^{m} \frac{\alpha_{ij}}{g_i(d_{ij})}}$$

and the profit is given by the following expression,

$$\pi(x,\alpha) = F(m(x,\alpha)) - G(x,\alpha)$$

where $F(\cdot)$ is a strictly increasing function which determines the expected sales (i.e., income generated) for a given market share m and $G(x,\alpha)$ is a function which gives the operating cost of a facility located at x with quality α. In our computational studies we have considered F to be linear and G to be separable, of the form $G(x,\alpha) = G_1(x) + G_2(\alpha)$, where $G_1(x) = \sum_{i=1}^{n} \Phi_i(d_{ix})$, with $\Phi_i(d_{ix}) = w_i/((d_{ix})^{\phi_{i0}} + \phi_{i1})$, $\phi_{i0}, \phi_{i1} > 0$ and $G_2(\alpha) = e^{\frac{\alpha}{\alpha_0}+\alpha_1} - e^{\alpha_1}$, with $\alpha_0 > 0$ and α_1 given values (other possible expressions for $G(x,\alpha)$ can be found in [2]).

The owner of the chain should also take into account that some form of competition also exists within the franchise, as expressed by the so-called *cannibalization*. When the new facility enters the market, the existing franchise's facilities might see a decrease of their own market share. To avoid this, the minimization of the cannibalization suffered by those facilities will be considered a third objective of the problem.

The market share captured by the existing facility $\ell \in \{1, \ldots, k\}$, *before* the new facility enters the market is given by

$$msb(\ell) = \sum_{i=1}^{n} w_i \frac{\dfrac{\alpha_{i\ell}}{g_i(d_{i\ell})}}{\displaystyle\sum_{j=1}^{m} \frac{\alpha_{ij}}{g_i(d_{ij})}}$$

which is easily seen to be strictly greater than its market share *after* entry given by

$$msa(\ell,(x,\alpha)) = \sum_{i=1}^{n} w_i \frac{\dfrac{\alpha_{i\ell}}{g_i(d_{i\ell})}}{\dfrac{\gamma_i \alpha}{g_i(d_{ix})} + \displaystyle\sum_{j=1}^{m} \frac{\alpha_{ij}}{g_i(d_{ij})}}.$$

The cannibalization suffered by ℓ is the difference between these market shares

$$Can(\ell,(x,\alpha)) = msb(\ell) - msa(\ell,(x,\alpha))$$

and our third objective is to minimize the sum of the cannibalizations suffered by all existing members of the chain,

$$\min Can(x,\alpha) = \sum_{\ell=1}^{k} Can(\ell,(x,\alpha)).$$

The problem considered is

$$\left\{ \begin{array}{ll} \max & M(x,\alpha) \\ \max & \pi(x,\alpha) \\ \min & Can(x,\alpha) \\ \text{s.t.} & d_{ix} \geq d_i^{\min} \ \forall i \\ & \alpha \in [\alpha_{\min}, \alpha_{\max}] \\ & x \in R \subset \mathbb{R}^2 \end{array} \right. \tag{1}$$

where the parameters $d_i^{\min} > 0$ and $\alpha_{\min} > 0$ are given thresholds, which guarantee that the new facility is not located over a demand point and that it has a minimum level of quality, respectively. The parameter $\alpha_{\max}$ is the maximum value that the quality of a facility may take in practice. By R we denote the region of the plane where the new facility can be located.

3. Obtaining a discrete approximation of the efficient set

For a majority of multi-objective problems, including location problems, it is not easy to obtain an exact description of the efficient set or Pareto-front, since those sets typically include an infinite number of points (usually a continuum set). That is why authors usually propose to present to the decision-maker a good 'representative set' of non-dominated points which suitably represents the whole Pareto-front. By a good representative set we mean a discrete set of points covering the complete Pareto-front and evenly distributed over it.

There is a plethora of metaheuristic methods with that purpose in literature. Nonetheless, the most common approaches utilized in literature is the use of multi-objective evolutionary algorithms (MOEAs). This is due to their ability to find multiple efficient solutions in one single simulation run. The numerous proposed variants have been surveyed, for instance, in [1]. Among them, the algorithms NSGA-II and SPEA2 have been the reference algorithms in the multi-objective evolutionary computation community for years. However, during the last five years, the multi-objective evolutionary algorithm based on decomposition MOEA/D has proved to be superior to other state-of-the-art algorithms (including both NSGA-II and SPEA2) when applied to a wide variety of multi-objective benchmark problems [7].

4. Summary

This paper describes a new tri-objective competitive facility location and design model. The efficiency of the algorithms MOEA/D and FEMOEA [6] (a recent evolutionary algorithm which has been successfully applied to other bi-objective location problems) to generate an effective approximation of the efficient set (and its corresponding Pareto-front) is investigated.

Acknowledgments

This work has been funded by grants from the Spanish Ministry of Economy and Competitiveness (ECO2011-24927 and TIN2012-37483), Junta de Andalucía (P10-TIC-6002, P11-TIC7176 and P12-TIC301), Fundación Séneca (15254/PI/10), in part financed by the European Regional Development Fund (ERDF). Juana López Redondo is fellow of the Spanish "Juan de la Cierva" contract program.

References

[1] C.A.C. Coello. Evolutionary multi-objective optimization: a historical view of the field. *IEEE Computational Intelligence Magazine*, 1(1):28–36, 2006.

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

[3] D.L. Huff. Defining and estimating a trading area. *Journal of Marketing*, 28(3):34–38, 1964.

[4] M. Kilkenny and J.F. Thisse. Economics of location: a selective survey. *Computers and Operations Research*, 26(14):1369–1394, 1999.

[5] F. Plastria. Static competitive facility location: an overview of optimisation approaches. *European Journal of Operational Research*, 129(3):461–470, 2001.

[6] J.L. Redondo, J. Fernández, J.D. Álvarez, A.G. Arrondo, P.M. Ortigosa. Approximating the Pareto-front of a planar bi-objective competitive facility location and design problem. Computers and Operations Research, doi: 10.1016/j.cor.2014.02.013, to appear.

[7] Q. Zhang and H. Li. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 13(2):284–302, 2009.

On longest edge division in simplicial branch and bound*

Juan F. R. Herrera[1], Leocadio G. Casado[1], and Eligius M. T. Hendrix[2]

[1]*Informatics Department, Universidad de Almería (ceiA3), Spain,* juanfrh@ual.es, leo@ual.es

[2]*Department of Computer Architecture, Universidad de Málaga, Spain,* Eligius.Hendrix@wur.nl

Abstract Simplicial partitions are suitable to divide a bounded area in branch and bound. In the iterative refinement process, a popular strategy is to divide simplices by their longest edge, thus avoiding needle-shaped simplices. A range of possibilities arises when the number of longest edges in a simplex is greater than one. The behaviour of the search is different depending on the selected longest edge. In this work, we investigate the importance of the rule to select an edge.

Keywords: longest edge bisection, branching rule, branch and bound, simplex

1. Introduction

Global Optimization pursuits the search of the global optima. Several methods can be used to find the solution. Within deterministic methods, the branch and bound method (B&B) guarantees to find a global minimum point up to a guaranteed accuracy ϵ. This method iteratively divides the search space discards subsets that are proven not to contain an ϵ global solution. Generally, five rules define the method:

- Branching rule: determines how to divide a subproblem into subproblems.
- Bounding rule: defines how to obtain upper and/or lower bounds of the subproblem's solution.
- Selection rule: chooses a subproblem among all subproblems stored in a working set.
- Rejection rule: discards subproblems which are proven not to contain a global solution.
- Termination rule: defines when the given accuracy has been reached. Once a subproblem meets this criterion, it is not further divided. Otherwise, it is stored in the working set.

Every B&B rule plays an important role in the efficiency of the algorithm. Careless decisions in one of the rules may lead to inefficient algorithms. This work focuses in the efficiency of the branching rule using longest edge bisection within simplicial B&B optimization methods.

For some problems like mixture design, the search space is a regular simplex. Here, we focus on box-constrained problems, where the search space is an n-dimensional hyper-rectangle that can be partitioned into a set of non-overlapping n-simplices. An n-simplex is a convex hull of $n + 1$ affinely independent vertices.

A recent study shows how the number of generated sub-simplices varies when different heuristics are applied in the iterative bisection of a regular n-simplex [1]. In that study, the complete binary tree is built by bisecting the heuristically-selected longest edge of a sub-simplex until the width, determined by the length of their longest edge, is smaller or equal to a given accuracy ϵ. A large reduction in the number of generated sub-simplices can be achieved when heuristics, different from bisecting the first longest edge in terms of vertex indexation (the default method), were used.

*This work has been funded by grants from the Spanish Ministry (TIN2008-01117 and TIN2012-37483) and Junta de Andalucía (P11-TIC-7176), in part financed by the European Regional Development Fund (ERDF). J.F.R. Herrera is a fellow of the Spanish FPU program.

In this context, we specifically study whether the reduction factor of the search tree is still large when longest-edge selection heuristics are applied to a simplicial B&B optimization method on box-constrained problems, where the initial search region is not a regular simplex and the termination criterion is based on the bounding rule.

Section 2 briefly explains the main features of the used simplicial B&B algorithm. Section 3 describes the studied division heuristics and Section 4 concludes.

2. Simplicial branch and bound method

In this section we cover both the initial space and the rules that define the simplicial B&B method to solve multidimensional global optimization problems.

Initial space

Most B&B methods use hyper-rectangular partitions. However, other types of partitions may be more suitable for some optimization problems. For the use of simplicial partitions, the feasible region is partitioned into simplices. The most preferable initial covering is face-to-face vertex triangulation. It involves partitioning the feasible region into a finite number of n-dimensional simplices with vertices that are also the vertices of the feasible region. A standard method [5] is triangulation into $n!$ simplices. All simplices share the diagonal of the feasible region and have the same hyper-volume. Figure 1 depicts a hypercube of dimension three partitioned into six irregular simplices.

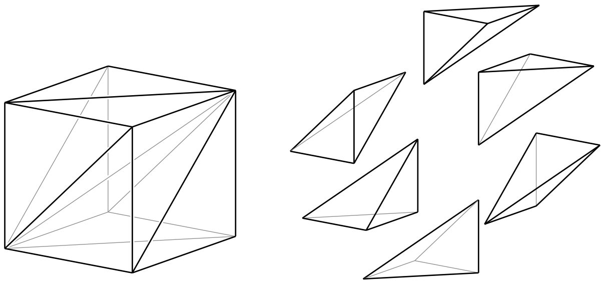


Figure 1: Division of a hypercube into six irregular simplices

Bounding and rejection rules

Consider the objective function f with a global minimum f^* on box-constrained area X. The function f is not required to be differentiable or (Lipschitz) continuous. Given a global minimum point x^*, let scalar K be such that

$$K \geq \max_{x \in X} \frac{|f(x) - f^*|}{\|x - x^*\|}, \tag{1}$$

where $\|\cdot\|$ denotes the Euclidean norm. The function $f^* + K\|x - x^*\|$ is an upper fitting according to [2]. For any $x \in X$ and therefore a subset of evaluated points $x_i \in X$, the area below

$$\varphi(x) = \max_i \{f_i - K\|x - x_i\|\} \tag{2}$$

cannot contain the global minimum (x^*, f^*). Let $f^U = \min_i f_i$ be an upper bound of f^* then the area $\{x \in X : \varphi(x) > f^U\}$ cannot contain the global minimum point x^*. To determine if a simplex S with vertices $v_0, v_1, \ldots, v_n$ contains or not an ϵ-optimal solution, the objective function is evaluated at each vertex v_i, obtaining the cutting cones:

$$\varphi_i(x) := f(v_i) - K\|x - v_i\|. \tag{3}$$

Let Φ be defined by

$$\Phi(S) = \min_{x \in S} \max_i \varphi_i(x). \tag{4}$$

Simplex S cannot contain the global minimum point x^*, and therefore S is rejected, if $f^U < \Phi(S)$. Notice that $\Phi(S)$ is a lower bound of f^* if S contains the minimum point x^*.

Selection and termination rules

The algorithm performs a depth-first search by selecting the sub-simlex with the smallest $\Phi(S)$ value among those generated in the last division, until the final accuracy is reached or both new sub-simlices are rejected. In general, depth-first search minimizes the memory requirement of the algorithm. A simplex S is not divided anymore when $\Phi(S) + \epsilon \geq f^U$.

3. Longest edge bisection

In the literature we can find many methods to divide a simplex [4]. One of them is the Longest Edge Bisection (LEB), which is a popular way of iterative division in the finite element method, since it is very simple and can easily be applied in higher dimensions [3]. This method consists of splitting a simplex using the hyperplane that connects the middle point of the longest edge of a simplex with the opposite vertices. The most common LEB division rule is the following:

LEB$_1$ The natural way to select a longest edge is to take the first one found. The sequence depends on the coding and storing of the vertices and edges, i.e. the index number assigned to each vertex of the simplex. When a simplex is split into two new sub-simplices, the new vertex of each sub-simplex has the same index as the one it substitutes.

Our preliminary studies show the existence of many sub-simplices having more than one longest edge when LEB$_1$ is used as iterative partition rule in a simplicial B&B algorithm.

In order to reduce the search tree size, other heuristics used for selecting the longest edge in the division of a regular n-simplex will be applied here to simplicial B&B algorithms. They are summarized below:

LEB$_\alpha$**:** For each vertex in a longest edge, the sum of the angles between edges ending at that vertex is determined and the longest edge that obtains the smallest sum is selected.

LEB$_C$**:** Bisects the longest edge with the largest distance from its middle point to the centroid of the simplex.

LEB$_W$**:** Selects an edge that has not been involved in many bisections yet via a weight system. A new vertex v_i is initiated with weight $w_i := 1$. Each time vertex v_i is involved in a divided edge, the weight is updated to $w_i := w_i + 1 \mod n$.

LEB$_M$**:** Longest edge with its furthest midpoint from the other vertices is selected.

The research goal is to determine a LEB rule that minimizes the search tree produced by a simplicial B&B algorithm.

4. Summary

This contribution sketches a simplicial branch-and-bound algorithm where the infeasible area is cut away via the concept of an upper fitting. It is illustrated that, in higher dimensional space, an appropriate decision must be taken in the selection of the edge to be bisected. In the talk, results of the investigation of several rules influencing the development of the algorithm are presented.

References

[1] G. Aparicio, L.G. Casado, E M.T. Hendrix, I. Garcia, and B.G. Toth. On computational aspects of a regular n-simplex bisection. In *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2013 Eighth International Conference on*, pages 513–518, Oct 2013. DOI 10.1109/3PGCIC.2013.88.

[2] W.P. Baritompa. Customizing methods for global optimization, a geometric viewpoint. *Journal of Global Optimization*, 3:193–212, 1993. DOI 10.1007/BF01096738.

[3] A. Hannukainen, S. Korotov, and M. Křížek. On numerical regularity of the face-to-face longest-edge bisection algorithm for tetrahedral partitions. *Science of Computer Programming*, 2013. DOI 10.1016/j.scico.2013.05.002.

[4] R. Horst and H. Tuy. *Global Optimization (Deterministic Approaches)*. Springer, Berlin, 1990.

[5] M.J. Todd. *The computation of fixed points and applications*, volume 24 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, 1976.

Proceedings of MAGO 2014, pp. 89 – 92.

Planar facility location with probabilistic outer competition and deterministic inner competition

A.G. Arrondo[1], J.L.Redondo[2], J.Fernández[3], B.G. Tóth[4], and P.M. Ortigosa[1]

[1] *University of Almería, CeiA3, C/ los Gallardos, s/n, 04120, La Cañada de San Urbano, Almería, Spain* agarrondo@ual.es, ortigosa@ual.es

[2] *University of Granada, ETS Ingenierías Informáticas y de Telecomunicaciones, C/ Periodista Daniel Saucedo Aranda, s/n, E-1807, Granada, Spain* jlredondo@ugr.es

[3] *University of Murcia, Campus Universitario de Espinardo, 30100, Murcia, Spain* josefdez@um.es

[4] *Budapest University of Technology and Economics, Faculty of Mathematics, 1111 Budapest, Egry József u. 1, Hungary* bog@math.bme.hu

Abstract The scenario considered in this work is the following. Several firms are present in the market and customers split their demand among the firms by patronizing only one facility from each firm, the one with the highest utility, and the demand is split among those facilities proportionally with their attraction. To solve this location problem, a heuristic evolutionary algorithm has been proposed, which includes a Weiszfeld-like method as local optimization procedure. A comprehensive computational study has been carried out to compare the heuristic algorithm with an exact interval Branch-and-Bound method. Additionally, in order to prove that the solutions offered by the heuristic method are not a consequence of randomness, but of the evolutionary procedures implemented, a multi-start algorithm based on the same Weiszfeld-like local procedure has been also implemented. Results show that the heuristic evolutionary algorithm is a good alternative to deal with this problem.

Keywords: Competitive location, multi-deterministic model, evolutionary algorithm, local optimizer, comparison

1. Introducction

Location science deals with the location of one or more facilities in a way that optimizes a certain objective (minimization of transportation costs, minimization of social costs, maximization of the market share, etc.). All location problems share several components, which leads to different models. The mathematical formulations and methods used to solve the problems vary substantially depending on the type of model [1]. An important question to take into account when modeling a location problem is if customers are free to choose the facility from which they are served. If so, knowing how customers buy goods among the existing facilities helps to estimate the market share captured by each facility. The *patronizing behavior* of the customers is usually considered in literature either as *deterministic*, i.e. the full demand of the customer is served by the facility to which he/she is *attracted* most (leading to Hotelling-type models [5]), or as *probabilistic*, i.e. the customer splits his/her demand among all the existing facilities (leading to Huff-type models [6, 3]).

In order to give a mathematical formulation, consider the following notation. Let i be the index of demand points, $i = 1, \ldots, i_{\max}$; c the index of competing chains, $c = 1, \ldots, c_{\max}$. Chain $c = 1$ is the locating chain. Let j denote the index of existing facilities, $j = 1, \ldots, j_{\max}$. We assume that from $j = j^1_{\min}(= 1)$ to $j^1_{\max}$ the facilities belong to chain $c = 1$ ($0 \leq j^1_{\max} < j_{max}$), from $j = j^2_{\min}(= j^1_{\max}+1)$ to $j^2_{\max}$ belong to chain $c = 2$,..., from $j = j^{c_{\max}}_{\min}(= j^{c_{\max}-1}_{\max}+1)$

to $j_{\max}^{c_{\max}}(=j_{\max})$ to chain $c=c_{\max}$. The variables of the problem are the coordinates giving the location of the new facility, $x=(x_1,x_2)$.

Let p_i be the location of demand point i ($i=1,\ldots,i_{\max}$) and w_i the buying power at p_i. Let f_j denote the location of existing facility j ($j=1,\ldots,j_{\max}$) and d_{ij} the distance between demand point p_i and facility f_j. If the attraction that p_i feels for f_j is denoted by u_{ij}, then the maximum attraction that p_i feels for any of the existing facilities of chain c, u_i^c, is given by $u_i^c=\max\{u_{ij}:j\in\{j_{\min}^c,\ldots,j_{\max}^c\}\}$. Finally, let $d_i(x)$ be to the distance between demand point p_i and the new facility (see [2]), $u_{i0}(x)$ the attraction that p_i feels for the new facility, and $M(x)$ the market share captured by the locating chain.

Based on these assumptions the market share captured by the chain when a deterministic rule is used is

$$M(x)=\sum_{i\in\{1,\ldots,n\}:\max\{u_i^1,u_{i0}(x)\}\geq\max\{u_i^c:c=2,\ldots,c_{\max}\}} w_i.$$

In the previous formula we have assumed that, in case of ties in the attraction, customers choose the locating chain. Notice that in the deterministic rule it is assumed that the attraction of the customers at p_i towards a chain is determined only by the facility to which they are attracted most. The rest of the facilities do not play any role.

When a probabilistic rule is used, the market share captured by the chain is given by

$$M(x)=\sum_{i=1}^{n} w_i\frac{u_{i0}(x)+\sum_{j=j_{\min}^1}^{j_{\max}^1}u_{ij}}{u_{i0}(x)+\sum_{j=1}^{j_{\max}}u_{ij}}.$$

In the probabilistic rule the attraction of the customers at p_i towards a chain is determined by all the facilities belonging to the chain. As we can see, it is assumed that the utility is *additive*: the utility for the first chain is given by $U_i^1(x)=u_{i0}(x)+\sum_{j=j_{\min}^1}^{j_{\max}^1}u_{ij}$.

In some cases, in order to have a better estimation of the market share captured by each facility or chain, new customer choice rules which model customer behaviour closer to reality are needed. In this work, a new rule will be considered, which is described next. This new rule has not been addressed in the literature before, and the aim is to set out and solve a new location model in the plane when this new rule is used.

2. Multi-deterministic rule

The scenario considered in this problem is the following. Several firms are present in the market and customers split their demand among the firms by patronising only one facility from each firm, the one with the highest utility. Then, the demand is split among those facilities proportionally with their attraction. Hakimi already proposed something similar in [4] (see Section 10.4 in that paper; Hakimi named it 'partially binary rule').

In order to give a mathematical formulation, let us consider the case in which there are $c_{\max}$ competing chains in the market. Then, the market share captured by the locating chain (the first one) is

$$M(x)=\sum_{i=1}^{n} w_i\frac{\max\{u_{i0}(x),u_i^1\}}{\max\{u_{i0}(x),u_i^1\}+\sum_{c=2}^{c_{\max}}u_i^c}. \tag{1}$$

As it can be seen in the formula, it is assumed here that the attraction of the customers at p_i towards a chain is determined only by the facility of the chain to which they are attracted most. The rest of the facilities of the chain do not play any role. But unlike the deterministic rule, now all the chains capture part of the demand at p_i.

3. A competitive location problem with multi-deterministic rule

Let α_{ij} be the quality of facility j as perceived by demand point i, α the quality of the new facility and γ_i a weight for the quality of the new facility as perceived demand point i. Let us assume that $u_{ij} = \alpha_{ij}/g_i(d_{ij})$ and $u_{i0}(x) = \gamma_i\alpha/g_i(d_i(x))$, where α is also a variable in the problem and $g_i(\cdot)$ is a given nondecreasing function. The planar competitive facility location and design problem with multi-deterministic patronizing behavior of customer for profit maximization is given by

$$\begin{cases} \max & \Pi(x,\alpha) = F(M(x,\alpha)) - G(x,\alpha) \\ \text{s.t.} & d_{ix} \geq d_{\min}\ \forall i \\ & \alpha \in [\alpha_{\min}, \alpha_{\max}] \\ & x \in S \subset \mathbb{R}^2 \end{cases} \tag{2}$$

where $M(x,\alpha)$ is given by formula (1), $F(\cdot)$ is a strictly increasing differentiable function which transforms the market share into expected sales, $G(x,\alpha)$ is a differentiable function which gives the operating cost of a facility located at x with quality α, and $\Pi(x,\alpha)$ is the profit obtained by the chain. Note that this profit disregards the operating costs of the existing facilities of the locating chain, since these are considered to be constant. The parameters $d_{\min} > 0$ and $\alpha_{\min} > 0$ are given thresholds, which guarantee that the new facility is not located over a demand point and that it has a minimum level of quality, respectively. The parameter $\alpha_{\max}$ is the maximum value that the quality of a facility may take in practice. By S we denote the region of the plane where the new facility can be located.

The function F will often be linear, $F(M(x,\alpha)) = c \cdot M(x,\alpha)$, where c is the income per unit of good sold. Of course, other functions can be more suitable depending on the real problem considered (see [3]).

The function $G(x,\alpha)$ should increase as x approaches to one of the demand points, since it is rather likely that around those locations the operational cost of the facility will be higher (due to the value of land and premises, which will make the cost of buying or renting the location higher). On the other hand, G should be a nondecreasing and convex function in the variable α, since the more quality we require of the facility, the higher the costs will be, at an increasing rate. We will assume G to be separable, i.e. of the form $G(x,\alpha) = G_1(x) + G_2(\alpha)$ (see [3]).

Figure 1 gives the graph of the objective function in the location domain for a problem with setting ($i_{max} = 71, j_{max} = 5, c = 2, j^1_{max} = 2, j^2_{max} = 3$) when the quality is fixed. The white holes in the graphs correspond to the forbidden regions around the demand points. As can be seen, this problem is a highly nonlinear optimization problem which requires global optimization techniques to be solved.

4. Solving the multi-deterministic location problem

In this work, we analyze three approaches to solve the previous problem. The first one is a simple multi-start algorithm in which a local search procedure (namely, a Weiszfeld-like algorithm) is repeated from different starting points. The second one is a evolutionary algorithm, which applies the same Weiszfeld-like algorithm as local optimizer. Finally, an interval branch-and-bound algorithm is also proposed. A comprehensive computational study has been carried out to compare the efficiency and the effectiveness of the proposed methods.

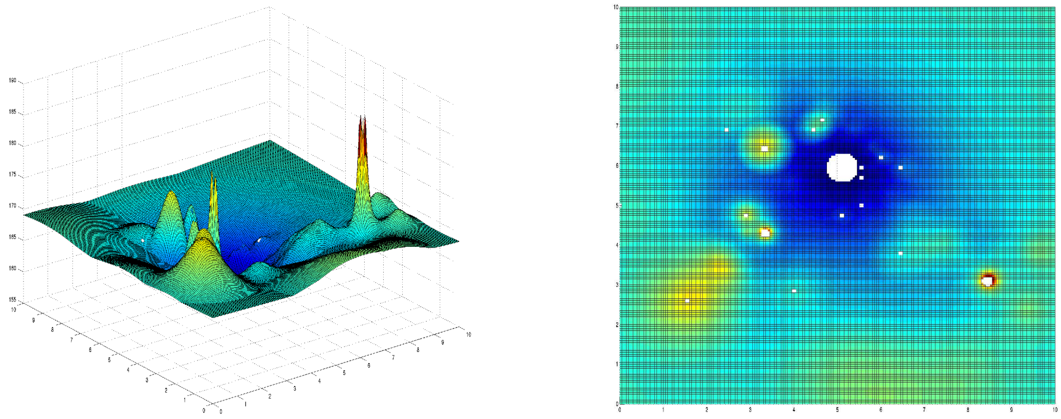


Figure 1: Objective function (on the left) and contour projected in the 2-dimensional location space (on the right) of the instance with setting $(i_{max} = 71, j_{max} = 5, c = 2, j^1_{max} = 2, j^2_{max} = 3)$ when $\alpha = 0.5$.

5. Summary

Analizing the computational results, we can infer that using the same computational resources and CPU time, the solutions obtained by the evolutionary algorithm are better than those obtained by the multistart heuristic, for both the objective value and the solution point. Furthermore, with a suitable parameter setting, the evolutionary algorithm is *always* able to obtain the global optimum, taking less time than the reliable interval branch-and-bound method. Additionally, it can handle much larger problems than the interval method.

Acknowledgments

This work has been funded by grants from the Spanish Ministry of Economy and Competitiveness (ECO2011-24927 and TIN2012-37483), Junta de Andalucía (P10-TIC-6002, P11-TIC7176 and P12-TIC301), Fundación Séneca (The Agency of Science and Technology of the Region of Murcia, 15254/PI/10), in part financed by the European Regional Development Fund (ERDF). Juana López Redondo is fellow of the Spanish "Juan de la Cierva" contract program.

References

[1] Z. Drezner and H.W. Hamacher. Facility location. Applications and theory Springer, Berlin, 2002-

[2] J. Fernández, P. Fernández, and B. Pelegrín. Estimating actual distances by norm functions: a comparison between the $l_{k,p,\theta}$-norm and the $l_{b_1,b_2,\theta}$-norm and a study about the selection of the data set. *Computers and Operations Research*, 29(6):609–623, 2002.

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

[4] S.L. Hakimi. Locations with spatial interactions: competitive locations and games. In R.L. Francis and P.B. Mirchandani, editors, *Discrete location theory*, pages 439–478. Wiley/Interscience, 1990.

[5] H. Hotelling. Stability in competition. *Economic Journal*, 39:41–57, 1929.

[6] D. L. Huff. A Programmed Solution for Approximating an Optimum Retail Location. *Land Economics*, 42:293–303, 1966.

[7] D. L. Huff. Defining and estimating a trading area. *Journal of Marketing*, 28(3):34–38, 1964.

Regular Simplex Refinement by Regular Simplices*

L. G. Casado[1], Boglárka G.-Tóth[2], Eligius M.T. Hendrix[3], and Inmaculada García[3]

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

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

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

Abstract A natural way to define branching in Branch-and-Bound for blending problem is to do bisection. The disadvantage of bisectioning is that partition sets are in general irregular. A regular simplex with fixed orientation can be determined by its center and size, allowing storage savings in a Branch-and-Bound algorithm from computational perspective. Unfortunately for dimension n>3 a regular simplex cannot be covered by regular subsimplices without overlapping. The possible difficulties of the refinement by regular simplices are studied here. The main challenge is to find a refinement with a good convergence ratio which facilitates the discarding of simplices in an overlapped and already evaluated region.

Keywords: unit simplex, subdivision, partition, covering

1. Introduction

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

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

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. As discussed by [3], quadratic quality requirements, minimum dose constraints, looking for robust solutions etc. may complicate the search for the best mixture design. Therefore Branch-and-Bound approaches are described in [3] and specific tests are introduced in [1].

In the Branch-and-Bound method, the initial problem is subsequently partitioned in more and more refined subproblems (branching) over which bounds of an objective function value, or on the constraint functions can be determined (bounding). The search is reduced by eliminating subproblems. One of the elimination rules used is based on defining a global upper bound f^U as the objective function value of the best ϵ-robust solution found so far. Subsets with a lower bound f^L_k of the objective function larger than the upper bound can be discarded, because they cannot contain an optimal solution.

The method starts with a set $C_1 = S$ as the first element of a list Λ of subsets (partition sets) and stops when the list Λ is empty. A generated subset is not stored on Λ, if it can be proved that it is infeasible and/or cannot contain a solution. The size of a simplex ($Size(C)$) is given by the Euclidean length of its largest edge. To force theoretical convergence, the termination rule establishes that the search does not continue on partition sets smaller in size than $\alpha \le \epsilon$.

*This work has been funded by grants from the Spanish Ministry (TIN2008-01117 and TIN2012-37483) and Junta de Andalucía (P11-TIC-7176), in part financed by the European Regional Development Fund (ERDF), and also by the project ICT COST Action TD1207 (EU).

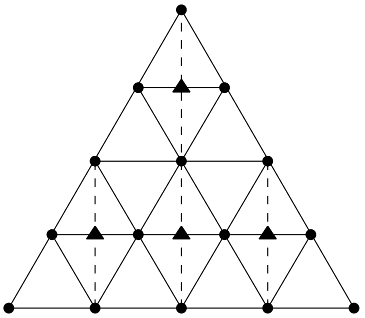


Figure 1: Bisection process. 2-dimensional projection.

The branching concerns the further refinement of the partition. This means that one of the subsets is selected to be split into new subsets. The use of simplicial sets in Branch-and-Bound and several ways of splitting them has been studied extensively in [2, 4]. Bisection of the longest edge of the selected simplex has the advantage that the sets never get a needle shape. Starting with the unit simplex, for all the generated simplices the length of the longest edge is at most twice the size of the shortest edge. Figure 1 sketches the idea of the bisection algorithm. It can be observed that points on a regular grid are generated, but that the bisection also generates edges (dotted lines) in at least one additional direction other than the facets of the unit simplex. The values of all generated points are a multiple of $(1/2)^K$, where K is an integer representing the depth of the search tree.

The storage requirements of the Branch-and-Bound search depends on the used search strategy. Depth-first has the smallest memory requirement. In order to know the feasible region, all final simplices has to be stored. This increases the memory requirements. Nowadays computers consume a lot of time when data do not fit in caches. Therefore it is desirable to use less storage even when it requires more computation, in order to reduce the overall time.

Dividing regular simplices by smaller regular simplices with fixed orientation reduces the stored information in the refinement process, because only the centre and the size of the subsimplices are needed to determine all the other information of a simplex, as the position of its vertices. This saving increases with the dimension. Unfortunately, it is not possible to divide the regular simplex in regular sub-simplices without overlapping for n>3.

The question to answer is if there exists a regular refinement which outperforms other usual partition schemes as the longest edge division, in terms of the number of simplices and vertices evaluations. Here we study a possible candidate.

2. Regular subdivision

In general, a simplicial set can be written as

$$\{x = \sum \lambda_j v_j, \sum \lambda_j = 1, \lambda_j \geq 0, j = 1, \dots n\} = \{x = V\lambda, \lambda \in S\}, \tag{2}$$

where S is the unit simplex defined in (1) and the vertices v_j in matrix V are affine independent. Alternatively, one can write a simplicial set as the convex hull of the vertex set, $conv(V)$. The viewpoint of writing a simplex as in (2) gives that from a subdivision perspective we can focus simply on the unit simplex. A subdivision of S can be translated into partitioning of any simplicial set by multiplication with the vertex matrix V.

Our question is how one can partition set S apart from the usual bisection method. Therefore, we introduce the Uniform Simplex Cover (USC) where the simplex is covered by equally sized, equally oriented overlapping subsimplices and we analyze its characteristics. To express the idea of equally sized and oriented simplices, we introduce the following concepts.

Each simplex C has a base vector b and a radius r where its vertices are described by walking from b a step r in direction $d_j := e_j - e_1$, with e_j the j-th unit vector:

$$C = conv(\{b, b + rd_2, b + rd_3, \dots, b + rd_n\}). \tag{3}$$

A subdivision of the unit simplex S is a collection

$$R = \{C_1, \dots C_p\} \quad \text{such that} \quad S \subset \cup_{i=1}^{p} C_i. \tag{4}$$

In this work we investigate how to choose m base vertices per axis distributed in a regular way, such that we get a subdivision R that covers S. We call this m–Uniform Simplex Cover, shortly mUSC.

2.1 Simplex covering by n simplices, 2USC

We first investigate 2USC and define its base vectors and radius. In 2USC, subdivision R consists of n subsimplices C_i, $i = 1, \dots, n$ with base points

$$b_i = e_1 + \frac{1}{n} d_i. \tag{5}$$

So they have the shape $(1, 0, \dots, 0)^T, ((n-1)/n, 1/n, 0, \dots, 0)^T, ((n-1)/n, 0, 1/n, 0, \dots, 0)^T, \dots, ((n-1)/n, 0, \dots, 0, 1/n)^T$. For the radius we use $(n-1)/n$;

$$C_i = conv(\{b_i + \frac{n-1}{n} d_1, b_i + \frac{n-1}{n} d_2, \dots, b_i + \frac{n-1}{n} d_n\}), \tag{6}$$

where of course $b_i + \frac{n-1}{n} d_1 = b_i$. The simplex is covered by 2USC when using these choices.

As 2USC has only n subsimplices, their vertex set can also be written by turning the orientation:

$$C_i = conv(\{e_i + \frac{n-1}{n}(e_j - e_i)\}). \tag{7}$$

This way of writing shows that covering of S by R is done when each subsimplex C of R covers the centroid of S. Therefore, the smallest reduction ratio allowing the covering of S is r_s=(n-1)/n and the overlap between C_i and S in terms of relative volume is:

$$\frac{Vol(C)}{Vol(S)} = \left(\frac{n-1}{n}\right)^{n-1}. \tag{8}$$

The total overlap between R and S is $n\left(\frac{n-1}{n}\right)^{n-1}$. The following table summarizes the smallest reduction ratio r_s and the relative overlap varying n. Results in terms of convergence are not promising as n increases.

n	2	3	4	5	6	7
r_s	0.5	$0.\dot{6}$	0.75	0.8	$0.8\dot{3}$	0.875
overlap	0	$0.\dot{3}$	0.69	1.05	1.42	1.78

To facilitate discarding of simplices in an overlapped and already evaluated region, the refinement should promote the sharing of vertices as much as possible. What should the reduction ratio be? A possible solution is the Golden ratio. Given a unit edge and a reduction ratio of r for the new sub-simplices, after d divisions the new simplices will have size r^d. They will use the vertices of the first division, if $1-r$, the other vertex on our edge equals to r^d. The following table shows the r_g values as solutions of $(1-r) - r^d = 0$ and the values of n where they can be applied taking into account that r_g cannot be smaller than r_s.

d	1	2	3	4	5	6	7
r_g	0.5	0.618	0.682	0.724	0.755	0.778	0.797
n	2	3	3	3	4	4	4

For instance, to use $r_g = 0.618$ instead of the smallest reduction ratio $r_s = 0.\dot{6}$ for $n = 3$ space seems to be better in terms of sharing vertices, because vertices are shared at every second division levels.

A so-called r_b solution can be to force the vertices of the simplex to be in the regular grid at $(1/2)^k$ pace for some k, with $r_b \geq r_s$.

To avoid redundant computation, a simplex in an already evaluated region can be discarded. This will require the development of appropriate computational routines. From a Branch-and-Bound perspective, three sets of simplices are needed:

- The set of active simplices, those needing further processing.
- The set of final simplices, which already reached the required precision, thus do not need refinement.
- The set of simplices determining the already examined regions, in order to discard a simplex because it is in the rejected or final region, or in a simplex already subdivided further.

In our investigation we also found covers that show less overlap, but where we have to relax the property of having equally oriented subsimplices. The subsimplices, Γ_i have a vertex set $\{e_i, e_i + \frac{n-2}{n-1}(e_j - e_i), j \neq i$. The union of Γ_i leaves a "hole" Γ_{n+1} in S which has an "upside down" orientation and is defined by vertices $w_j = \frac{1}{n-1}\sum_{k \neq j} e_k, j = 1 \ldots n$, which are the centroids of the facets of S. Subdivision $P = \{\Gamma_1, \ldots \Gamma_{n+1}\}$ now just covers S with $n + 1$ simplices of smaller size than C_j. To use this subdivision, one has an upside down orientation. For storage purposes, one should store this orientation in a clever way.

3. Conclusions

This work studies possible ways to refine a regular simplex by regular subsimplices, challenging in the development of efficient algorithms. According to the presented results, the $n = 4$ case seems to be the most promising to study in order to outperform previous partitioning methods. We will first tackle it.

References

[1] 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.

[2] 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.

[3] 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.

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

Computational experience on Distance Geometry Problems 2.0*

Claudia D'Ambrosio[1], Vu Khac Ky[1], Carlile Lavor[2], Leo Liberti[1,3], and Nelson Maculan[4]

[1] *CNRS LIX Ecole Polytechnique, Palaiseau 91128, France* dambrosio,vu,liberti@lix.polytechnique.fr

[2] *IMECC, University of Campinas, Brazil* clavor@ime.unicamp.br

[3] *IBM "T.J. Watson" Research Center, Yorktown Heights, 10598 NY, USA* leoliberti@us.ibm.com

[4] *COPPE, Federal University of Rio de Janeiro, Brazil* maculan@cos.ufrj.br

Abstract We propose a set of formulations and reformulations of the Distance Geometry Problem, which we evaluate computationally with both local and global off-the-shelf solvers. The local solvers are cast in a global optimization metaheuristic (Variable Neighbourhood Search) since the problem is nonconvex and non-global optima are usually of limited practical interest.

Keywords: nonlinear programming, protein conformation, sensor networks

1. Introduction

Roughly every decade, it is useful to assess how generic off-the-shelf solvers perform on mathematical programming formulations of any sufficiently important problem. We commented on computational experiments of the Distance Geometry Problem (DGP) in [7]: although the paper appeared in 2006, the experiments were conducted between 2004 and 2005, so it is high time to re-evaluate the current state of the art.

The DGP requires to "draw" a given weighted graph in $\mathbb{R}^K$ in such a way that the Euclidean distances of the segments between pairs of vertices match the given edge weights. More formally, given a simple undirected graph $G = (V, E)$, an integer $K > 0$, and an edge weight function $d : E \to \mathbb{R}_+$, the DGP asks to establish or deny the existence of a vertex realization function $x : V \to \mathbb{R}^K$ such that:

$$\forall \{u, v\} \in E \qquad \|x_u - x_v\|_2 = d_{uv}. \tag{1}$$

Notationwise, we let $n = |V|$ and $m = |E|$. More information can be found in [10].

In the following, we briefly summarize the results of [7] in Sect. 2, then proceed to list the DGP formulations we shall consider (Sect. 3), the evaluation framework (Sect. 4), and the test set (in Sect. 5). The computational results will be presented at the conference.

2. The computational set-up in [7]

Our testbed for [7] was simple (*way* too simple, in fact): three cubic grid instances taken from [12], and six "protein-like" instances generated randomly according to [6]. We solved these instances using three global optimization solvers: a deterministic, ε-approximate spatial Branch-and-Bound (sBB) algorithm [8], a stochastic Multi-Start (MS) algorithm based on Sobol' sequences [5], and a Variable Neighbourhood Search (VNS) solver for nonconvex Nonlinear Programs (NLP) [9].

*V. Ky is sponsored by a Microsoft Research Ph.D. grant

The solvers were launched in their default configurations to solve the following unconstrained NLP formulation of the DGP:

$$\min_{x\in\mathbb{R}^{Kn}} \sum_{\{u,v\}\in E} (\|x_u - x_v\|_2^2 - d_{uv}^2)^2 \tag{2}$$

on each instance. All cubic grid instances were solved at global optimality by all solvers, as well as four out of six protein-like instances. The remaining two instances were solved by the stochastic solvers (MS and VNS), whereas the sBB was terminated because of the CPU time threshold (1hr of user time). Only one instance failed to be solved to global optimality (i.e. within $\varepsilon = 10^{-3}$), but came nonetheless pretty close. The best overall solver was the VNS.

3. The DGP formulation zoo

First off, all formulations we consider are box-constrained (this makes life simpler for certain solvers) to $x \in [-M, M]^{Kn}$ where $M = \frac{1}{2}\sum_{\{u,v\}\in E} d_{uv}$: we do not write these bounds explicitly below. Every formulation comes with variants; a variant which holds for every formulation is the following: replace $\|x_u - x_v\|_2^2$ by $\|x_u - x_v\|_2$ and d_{uv}^2 by d_{uv}. In such variants, because of floating point issues, $\sqrt{\alpha}$ is implemented as $\sqrt{\alpha+\delta}$, where δ is $O(10^{-10})$. Notationwise, $\mathbf{M} = [-M, M]^m$.

3.1 Exact formulations

1. Eq. (2). Variant: replace $\sum$ with $\max$.

2. This formulation minimizes slacks and tries to satisfy Eq. (1):

$$\left.\begin{array}{rrcl} \min\limits_{x,s} & \sum\limits_{\{u,v\}\in E} s_{uv}^2 & & \\ \forall\{u,v\}\in E & \|x_u - x_v\|_2^2 & = & d_{uv}^2 + s_{uv}. \end{array}\right\} \tag{3}$$

   Variants: (i) replace s_{uv}^2 with $s_{uv}^+ + s_{uv}^-$ and s_{uv} with $s_{uv}^+ - s_{uv}^-$, where $s^+, s^- \geq 0$; (ii) replace $\sum$ with $\max$.

3. This formulation exploits the convexity and concavity of the equations in Eq. (1) separately:

$$\left.\begin{array}{rrcl} \max\limits_{x} & \sum\limits_{\{u,v\}\in E} \|x_u - x_v\|_2^2 & & \\ \forall\{u,v\}\in E & \|x_u - x_v\|_2^2 & \leq & d_{uv}^2. \end{array}\right\} \tag{4}$$

4. This is a Nonlinear Complementarity Problem (NCP) formulation:

$$\left.\begin{array}{rrcl} \max\limits_{x,y\in\mathbf{M},z\in[0,1]^m} & \sum\limits_{\{u,v\}\in E} z_{uv} & & \\ \forall\{u,v\}\in E & \|x_u - x_v\|_2^2 & = & y_{uv} \\ \forall\{u,v\}\in E & (y_{uv} - d_{uv}^2)z_{uv} & = & 0. \end{array}\right\} \tag{5}$$

5. This exploits $\|x_u - x_v\|_2^2 = (x_u - x_v)(x_u - x_v)$:

$$\left.\begin{array}{rrcl} \min\limits_{x,\sigma\in(2\mathbf{M})^K,\tau\in(2\mathbf{M})^K} & \sum\limits_{\{u,v\}\in E}\sum\limits_{k\leq K} (\sigma_{uvk} - \tau_{uvk})^2 & & \\ \forall\{u,v\}\in E, k\leq K & x_{uk} - x_{vk} & = & \sigma_{uv} \\ \forall\{u,v\}\in E & \sum\limits_{k\leq K} \sigma_{uvk}\tau_{uvk} & = & d_{uv}^2. \end{array}\right\} \tag{6}$$

6. This is a nonsmooth version of Eq. (2):

$$\min_x \sum_{\{u,v\}\in E} |\|x_u - x_v\|_2^2 - d_{uv}^2|. \tag{7}$$

   Variant: replace $\sum$ with $\max$.

3.2 Pointwise exact reformulations

These are formulations which are only exact for a specific set of values assigned to certain parameters; they can be used in a stochastic search setting (such as MS or VNS) where the global search occurs over the parameter values. The advantage is that they describe convex problems, so they are solved efficiently.

1. We use formulation 4 and rewrite the norm terms as per Item 5 in Sect. 3.2. This yields:

$$\left.\begin{array}{rcl} \max\limits_x & = & \sum\limits_{\{u,v\}\in E} \sum\limits_{k\le K} \theta_{uvk}(x_{uk} - x_{vk}) \\ \forall\{u,v\}\in E & & \|x_u - x_v\|^2 \le d_{uv}^2 \end{array}\right\} \tag{8}$$

   (it can be shown that there exist values of θ for which (8) is an exact formulation for the DGP).

2. Every formulation in Sect. 3.1 which involves the term $\|x_u - x_v\|_2^2$ gives rise to a pointwise exact convex reformulation, apart from Eq. (5) which gives rise to a Linear Complementarity Problem (LCP).

3. Eq. (7) can itself be interpreted as a pointwise exact convex reformulation if τ are taken as parameters rather than decision variables. It can be further reformulated as a pointwise exact *linear* reformulation by replacing the objective function by $\min\limits_{x,\sigma} \sum\limits_{\substack{\{u,v\}\in E \\ k\le K}} |\sigma_{uvk} - \tau_{uvk}|$.

4. The computational evaluation framework

There is an obvious evaluation framework which consists of gathering computational measures about quality and efficiency of each solver on each formulation for each instance, and compare them on various indices: one may thus answer empirical questions such as, "what is the best solver+formulation combination for a given instance?", or "I need to find the conformation of a set of proteins given some distance data: what solver should I buy and what formulation should I use?" Of course one may also fail to answer such questions, whenever there is no clear winner.

A less trivial evaluation framework is given by the Multiplicative Weights Algorithm (MWA) [1]: each of the N solver+formulation combinations (indexed by $i = (s, f)$) is assigned a weight, which is initially set to 1. We decide an order $<$ for the instance set, and then solve each instance t in the set using every solver+formulation i in the order $<$. At the t-th iteration, we record a non-negative cost μ_{it} of the pair (i, t), which is a convex combination of the solution error and the CPU time (both scaled to the respective maxima). These costs are used to update the weights $\omega_{i,t+1} = \omega_{it}(1 - \frac{1}{2}\mu_{it})$. After all instances have been looked at, the result is a multivariate distribution $p = (\omega_{it}/\Phi)_{i,t}$ where $\Phi = \sum_{i,t}\omega_{it}$. The marginal distributions give an idea of the relative success and failures of the different methods and instances. This whole computation can be repeated for different orders $<$, chosen e.g. to always cluster instances of the same type. An interesting feature of the MWA is that it provides a relative bound on its total error:

$$\sum_t \sum_i \mu_{it} p_{it} \le 2\ln N + \frac{3}{2} \min_i \sum_t \mu_{it}, \tag{9}$$

which is a direct consequence of [1, Thm. 2.1] when the costs are nonnegative. In other words, the total weighted error made by all solvers+formulations over all instances is bounded above by a linear function of the best combination.

For a candidate solution $x' \in \mathbb{R}^{Kn}$, the average and maximum solution error definitions are:

$$\begin{aligned} \eta_{\text{avg}}(x') &= \frac{1}{m} \sum_{\{u,v\}\in E} |\|x'_u - x'_v\|_2 - d_{uv}| \\ \eta_{\max}(x') &= \max_{\{u,v\}\in E} |\|x'_u - x'_v\|_2 - d_{uv}|. \end{aligned}$$

5. The test set

We test the whole formulation zoo (with variants) with the following solvers (for now[1]): SNOPT [4] (local), IPOPT [2] (local), FILTER [3] (local), COUENNE (global). Local solvers and pointwise exact reformulations are tested in a Variable Neighbourhood Search framework as in [9] (i.e. with hyper-rectangular neighbourhoods centered around the current iterate). All solvers will be given the same maximum CPU time.

We shall consider DGP instances for $K = 1$ (application to clock synchronization [13]), $K = 2$ (application to sensor networks [11]), $K = 3$ (application to protein conformation from NMR data [10]). The MWA evaluation framework will be run on every order on $\{1, 2, 3\}$.

References

[1] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8:121–164, 2012.

[2] COIN-OR. *Introduction to IPOPT: A tutorial for downloading, installing, and using IPOPT*, 2006.

[3] R. Fletcher and S. Leyffer. User manual for FILTER. Technical report, University of Dundee, UK, March 1999.

[4] P.E. Gill. *User's guide for SNOPT version 7.2.* Systems Optimization Laboratory, Stanford University, California, 2006.

[5] S. Kucherenko and Yu. Sytsko. Application of deterministic low-discrepancy sequences in global optimization. *Computational Optimization and Applications*, 30(3):297–318, 2004.

[6] C. Lavor. On generating instances for the molecular distance geometry problem. In L. Liberti and N. Maculan, editors, *Global Optimization: from Theory to Implementation*, pages 405–414. Springer, Berlin, 2006.

[7] C. Lavor, L. Liberti, and N. Maculan. Computational experience with the molecular distance geometry problem. In J. Pintér, editor, *Global Optimization: Scientific and Engineering Case Studies*, pages 213–225. Springer, Berlin, 2006.

[8] L. Liberti. *Reformulation and Convex Relaxation Techniques for Global Optimization*. PhD thesis, Imperial College London, UK, March 2004.

[9] L. Liberti and M. Dražic. Variable neighbourhood search for the global optimization of constrained NLPs. In *Proceedings of GO Workshop, Almeria, Spain*, 2005.

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

[11] A. Man-Cho So and Y. Ye. Theory of semidefinite programming for sensor network localization. *Mathematical Programming B*, 109:367–384, 2007.

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

[13] A. Singer. Angular synchronization by eigenvectors and semidefinite programming. *Applied and Computational Harmonic Analysis*, 30:20–36, 2011.

[1] These are preliminary results.

Localization on smart grids*

Pierre-Louis Poirion[1], Sonia Toubaline[1], Claudia D'Ambrosio[1], and Leo Liberti[1,2]

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

[2] *IBM "T.J. Watson" Research Center, Yorktown Heights, 10598 NY, USA* leoliberti@us.ibm.com

Abstract We formalize the problem of localizing monitoring equipment on electrical networks. Each monitoring device can be installed on any link of the network. Various constraints must be taken into account, including topological constraints and Euclidean distance constraints. This yields a Mixed-Integer Nonlinear Program (MINLP) with combinatorial as well as Euclidean distance constraints.

Keywords: MINLP, electrical networks, sensor networks

1. Introduction

An electrical network is a distribution network for the electricity commodity. Some nodes are production nodes, some nodes are demand nodes, there may be intermediate nodes, and the links are usually cables. The technical constraints which regulate the electrical flow involve the physics of alternating currents, and include frequency and phase terms [1, 2, 3]. Although the definition of a *smart grid* is somewhat fuzzy, there is a general agreement that a smart grid should be:

- accountable as regards cost, capacity and resilience down to a very precise detail (e.g. at each second, at each node, and so on);
- robust to failures;
- make use of very different energy sources (hopefully environmentally friendlier than burning coal and gas).

Of course these properties are not independent: the network can be robust *if* it is continuously and precisely monitored, and, in the case of failures, alternative sources of energy are readily available. In this work, we focus on the first of the above properties, i.e. accountability.

For an electrical network to be fully accountable, many monitoring devices have to be installed on its nodes and links (more precisely, the device could be localized anywhere along any link). Also, the information collected by these devices has to be sent to centralization servers which are supposed to store and/or perform computation on these data. Data communication can be achieved by using the power lines or wirelessly. The main objective is to install as few devices as possible subject to the network being satisfactorily monitored.

2. Formulation

2.1 Parameters, variables, objective

Let $G = (V, E)$ be the graph representation of the power network, with node set V and link set E. Each link is a pair $\{i, j\}$ of nodes. The network G is embedded in $\mathbb{R}^2$: for each $i \in V$ let

*Financial support from the ADEME SO-grid project is gratefully acknowledged.

$\nu_i = (\nu_{i1}, \nu_{i2}) \in \mathbb{R}^2$ be the position of node i, and for each $\{i,j\} \in E$ let $\gamma_{ij} : [0,1] \to \mathbb{R}^2$ be the closed-form description of the embedding of the link $\{i,j\}$ in the plane, such that $\gamma_{ij}(0) = \nu_i$ and $\gamma_{ij}(1) = \nu_j$, and $\gamma_{ij}(t) = (\gamma_{ij1}(t), \gamma_{ij2}(t))$ for each $t \in [0,1]$.

For each $i \in V$ let $z_i = 1$ iff a node device is installed at i, and 0 otherwise. For each $\{i,j\} \in E$ let $y_{ij} = 1$ iff a link device is installed on $\{i,j\}$, and 0 otherwise. Let $x_{ij} \in \mathbb{R}^2$ be the position of the link device on $\{i,j\}$, and $t_{ij} \in [0,1]$ be such that $\gamma_{ij}(t_{ij}) = x_{ij}$ if the corresponding link is active:

$$\forall \{i,j\} \in E \quad x_{ij} = y_{ij}\gamma_{ij}(t_{ij}) \tag{1}$$

Cost minimization yields:

$$\min_{x,y,z} \sum_{i \in V} z_i + \sum_{\{i,j\} \in E} y_{ij}. \tag{2}$$

Extending this function to different unit costs for different equipment is very easy.

2.2 Covering constraints

Next, there are covering constraints on nodes and links:

$$\begin{aligned} \forall i \in V \quad z_i + \sum_{\substack{j \in V \\ \{i,j\} \in E}} z_j &\geq 1 && (3) \\ \forall \{i,j\} \in E \quad \sum_{\substack{\{h,k\} \in E \\ h=i \vee h=j}} y_{hk} &\geq 1. && (4) \end{aligned}$$

These constraints ensure that for each node/link neighbourhood at least one monitoring device is installed.

2.3 Communication extent constraints

The communication extent constraints concern the ability of the communication devices to perform their function. If the communication occurs on the power lines, then the constraints are technical (being on either side of a voltage drop barrier, making sure that frequencies and phase overlap nondestructively) and largely depend on the specific properties of the network and the device, so they are difficult to generalize. If the communication is wireless, then it is either anchor-based or point-to-point.

In the first case, it means that every communication device sends its data to a wired hub, commonly located at each node, which has to be within a distance threshold ρ of the device:

$$\forall \{i,j\} \in E \quad y_{ij}\|x_{ij} - z_i\nu_i - z_j(1-z_i)\nu_j\|_2 \leq \rho. \tag{5}$$

Eq. (5) makes sure that every communication device on a link is close enough to a hub on a nearby node.

In the second case, we need to ensure connectivity with additional flow variables on the completion of the line graph G, i.e. the complete graph $\bar{G}$ having E as vertex set: for each $e, f, g, h \in E$ let $w_{ef}^{gh} = 1$ if the communication devices on links g and h use the communication devices on links e, f as intermediate hops because they are within the distance threshold ρ, and 0 otherwise. We use these variables in a multicommodity flow setting, where the sources and targets are the links which have a communication device installed.

$$\forall g \neq h, e \neq f \in E \quad y_e y_f \|x_e - x_f\|_2 \leq w^{gh}_{ef}\rho \tag{6}$$

$$\forall g \neq h \in E \quad \sum_{\substack{e\in E\\ e\neq g}} y_e(w^{gh}_{ge} - w^{gh}_{eg}) = y_g y_h \tag{7}$$

$$\forall g \neq h \in E, e \in E \smallsetminus \{g,h\} \quad \sum_{\substack{f\in E\\ f\neq e}} y_f(w^{gh}_{ef} - w^{gh}_{fe}) = 0. \tag{8}$$

Eq. (6) enforces $w = 1$ on those link pairs having installed communication devices, Eq. (7) are the multicommodity flow constraints at the source nodes, and Eq. (8) are the flow conservation equations.

2.4 Reformulation

The above formulation is a nonconvex MINLP. By assuming box bounds on all continuous variables, however, it can be reformulated exactly to a MINLP where the only nonconvexity is given by Eq. (1) in case the γ functions are nonlinear. First, reformulate Eq. (5) and Eq. (6) by means of the standard "big M" technique. Second, all of the remaining products only involve binary variables, and can therefore be linearized using Fortet's reformulation [4, 5].

2.5 Validation

The hub model (with Eq. (5)) was validated with randomly generated neworks with up to 100 nodes and around 1500 links, using straight line segments as arcs.

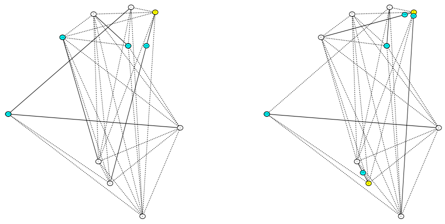


Figure 1: Two 2D instances with 10 vertices and edge probability 0.5: linear segments (left), quadratic curve segments (right).

References

[1] D. Bienstock and A. Verma. The n-k problem in power grids: new models, formulations and computation. *SIAM Journal on Optimization*, to appear.

[2] J.F. Bonnans. Mathematical study of very high voltage power networks i: The optimal dc power flow problem. *SIAM Journal on Optimization*, 7:979–990, 1997.

[3] J.F. Bonnans. Mathematical study of very high voltage power networks ii: The ac power flow problem. *SIAM Journal on Applied Mathematics*, 58:1547–1567, 1998.

[4] R. Fortet. Applications de l'algèbre de Boole en recherche opérationelle. *Revue Française de Recherche Opérationelle*, 4:17–26, 1960.

[5] L. Liberti, S. Cafieri, and F. Tarissan. Reformulations in mathematical programming: A computational approach. In A. Abraham, A.-E. Hassanien, P. Siarry, and A. Engelbrecht, editors, *Foundations of Computational Intelligence Vol. 3*, number 203 in Studies in Computational Intelligence, pages 153–234. Springer, Berlin, 2009.

An Extension of the αBB-type Underestimation to Linear Parametric Hessian Matrices

Milan Hladík[1]

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

Abstract The classical αBB method is a global optimization method the important step of which is to determine a convex underestimator of an objective function on an interval domain. Its particular point is to enclose the range of a Hessian matrix in an interval matrix. To have a tighter estimation of the Hessian matrices, we investigate a linear parametric form enclosure in this paper. We also show that one way to obtain this form is by using a slope extension of the Hessian entries.

Keywords: interval matrix, convex underestimator

1. Introduction

Most of the global optimization methods are based on exhaustive splitting of a search space into smaller parts, usually boxes, and using interval computation to obtain rigorous lower and upper estimations on the global optimal value and to remove boxes certainly not containing any global minimizer. In this context, it is important to be able to compute a tight lower bound of an objective function on a box. The well-known αBB method [2, 3, 12] constructs a convex underestimator, which touches the objective function at the vertices of the box. To describe the method, we have to introduce some notation first.

An interval matrix is defined as

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

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

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

The set of all interval $m \times n$ matrices is denoted by $\mathbb{IR}^{m\times n}$. The magnitue of $\boldsymbol{A} \in \mathbb{IR}^{m\times n}$ is $|\boldsymbol{A}| := \max\{|\underline{A}|, \overline{A}|\}$. Intervals and interval vectors are considered as special cases of interval matrices. For interval arithmetic, see e.g. [10, 11].

Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be a twice-differentiable function and $\boldsymbol{x} \in \mathbb{IR}^n$ an interval vector representing domains of the variables. The problem is to construct an underestimator function $g : \mathbb{R}^n \mapsto \mathbb{R}$ satisfying two conditions:

1. $f(x) \geq g(x)$ for every $x \in \boldsymbol{x}$,
2. $g(x)$ is convex on $x \in \boldsymbol{x}$.

The classical αBB method constructs the convex underestimator in the form of

$$g(x) := f(x) - \sum_{i=1}^{n} \alpha_i(\overline{x}_i - x_i)(x_i - \underline{x}_i),$$

where $\alpha_i \geq 0$, $i = 1, \ldots, n$, are determined such that $g(x)$ is convex.

The Hessian of $g(x)$ reads

$$\nabla^2 g(x) = \nabla^2 f(x) + 2\text{diag}\,(\alpha),$$

where diag (α) is the diagonal matrix with entries $\alpha_1, \dots, \alpha_n$. The parameters α_i-s may be calculated it the following way. Let $\boldsymbol{H}$ be an interval matrix enclosing the image of $\nabla^2 f(x)$ over $x \in \boldsymbol{x}$. Now, to achieve convexity of $g(x)$, it is sufficient to choose α such that each matrix in $\boldsymbol{H} + 2\text{diag}\,(\alpha)$ is positive semidefinite, i.e., its eigenvalues are non-negative. Eigenvalues of interval matrices were investigated e.g. in [2, 6, 9]. For the purpose of the αBB method, it seems that the most convenient method for bounding eigenvalues of interval matrices is the scaled Gerschgorin inclusion [2, 12]. Its benefits are that it is easy to compute and eliminate the unknowns α_i, $i = 1, \dots, n$, and it is also usually sufficiently tight. For any positive $d \in \mathbb{R}^n$, we can put

$$\alpha_i := \max\left\{0, -\tfrac{1}{2}\left(\underline{h}_{ii} - \textstyle\sum_{j \neq i} |\boldsymbol{h}_{ij}| d_j / d_i\right)\right\}, \quad i = 1, \dots, n.$$

To reflect the range of the variable domains, it is frequently used $d := x^{\Delta}$. In [8], it was shown that this choice is optimal under some assumptions. The author also proposed a local improvement method to compute a more convenient scaling vector d in the general case.

The efficiency of symbolic computation of the Hessian matrix was studied in [7]. Symbolic evaluation has a big potential in determining much tighter underestimators, however, it is very difficult to implement symbolic expression simplifications automatically by a computer program.

2. Hessians in linear parametric forms

Traditionally, one encloses the range of the Hessian matrix in an interval matrix $\boldsymbol{H}$ such that

$$F(x) := \nabla^2 f(x) \in \boldsymbol{H}, \quad \forall x \in \boldsymbol{x}. \tag{1}$$

Such an enclosure is, however, often very coarse and overestimates the true range. Herein, we approach to a more gentle enclosure by considering an interval matrix in a parametric form

$$H(p) = \sum_{k=1}^{K} H^{(k)} p_k,$$

where $H^{(1)}, \dots, H^{(K)} \in \mathbb{R}^{n \times n}$ are fixed matrices and $p_1, \dots, p_K$ are parameters varying respectively in $\boldsymbol{p}_1, \dots, \boldsymbol{p}_K \in \mathbb{IR}$. Such parametric forms are nowadays frequently used to model dependencies between parameters in interval linear equation solving [5, 13].

Suppose that we have such a parametric form satisfying the basic enclosure property

$$\forall x \in \boldsymbol{x},\ \exists p \in \boldsymbol{p} : F(x) = H(p).$$

This model is more general than (1), and it enables to derive tighter underestimators. In principle, we can evaluate

$$\boldsymbol{H} := H(\boldsymbol{p}) = \sum_{k=1}^{K} H^{(k)} \boldsymbol{p}_k$$

by using interval arithmetic and apply the standard underestimator on the interval matrix $\boldsymbol{H}$, but this simplification does not utilize the linear parametric form, and the results are unnecessarily overestimated.

Let $d \in \mathbb{R}^n$. In order that $\alpha_1, \dots, \alpha_n$ are admissible, they must be nonnegative and satisfy

$$-2\alpha_i \le H(p)_{ii} - \textstyle\sum_{j \neq i} |H(p)_{ij}| d_j / d_i, \quad \forall p \in \boldsymbol{p},\ \forall i = 1, \dots, n. \tag{2}$$

Fix $i \in \{1, \dots, n\}$ and define

$$\begin{aligned} J^{+} &:= \{j \neq i;\ \underline{H(\boldsymbol{p})}_{ij} \geq 0\}, \\ J^{-} &:= \{j \notin J^{+} \cup \{i\};\ \overline{H(\boldsymbol{p})}_{ij} \leq 0\}, \\ J^{0} &:= \{1, \dots, n\} \setminus (J^{+} \cup J^{-} \cup \{i\}). \end{aligned}$$

Then we can simplify (2) to

$$-2\alpha_i \leq H(p)_{ii} - \sum_{j \in J^{+}} H(p)_{ij} \frac{d_j}{d_i} + \sum_{j \in J^{-}} H(p)_{ij} \frac{d_j}{d_i} \quad - \sum_{j \in J^{0}} |H(p)_{ij}| \frac{d_j}{d_i}. \tag{3}$$

To get rid of the absolute value, we estimate it from above by the linear function [1] and write

$$|H(p)_{ij}| \leq \gamma_{ij} H(p)_{ij} + \beta_{ij},$$

where

$$\gamma_{ij} = \frac{\left|\overline{H(\boldsymbol{p})}_{ij}\right| - \left|\underline{H(\boldsymbol{p})}_{ij}\right|}{\overline{H(\boldsymbol{p})}_{ij} - \underline{H(\boldsymbol{p})}_{ij}} \text{ and } \beta_{ij} = \frac{\overline{H(\boldsymbol{p})}_{ij} \left|\underline{H(\boldsymbol{p})}_{ij}\right| - \underline{H(\boldsymbol{p})}_{ij} \left|\overline{H(\boldsymbol{p})}_{ij}\right|}{\overline{H(\boldsymbol{p})}_{ij} - \underline{H(\boldsymbol{p})}_{ij}}.$$

Using these estimations and (3), we obtain

$$\begin{aligned} -2\alpha_i &\leq H(p)_{ii} - \sum_{j \in J^{+}} H(p)_{ij} \frac{d_j}{d_i} + \sum_{j \in J^{-}} H(p)_{ij} \frac{d_j}{d_i} - \sum_{j \in J^{0}} (\gamma_{ij} H(p)_{ij} + \beta_{ij}) \frac{d_j}{d_i} \\ &= \sum_{k=1}^{K} \left(H_{ii}^{(k)} - \sum_{j \in J^{+}} H_{ij}^{(k)} \frac{d_j}{d_i} + \sum_{j \in J^{-}} H_{ij}^{(k)} \frac{d_j}{d_i} - \sum_{j \in J^{0}} \gamma_{ij} H_{ij}^{(k)} \frac{d_j}{d_i} \right) p_k - \sum_{j \in J^{0}} \beta_{ij} \frac{d_j}{d_i}. \end{aligned}$$

This inequality must be satisfied for each $p \in \boldsymbol{p}$, so to get the smallest α_i, we take

$$\alpha_i := \max\left\{0, -\tfrac{1}{2}\underline{h}_i\right\},$$

where

$$\boldsymbol{h}_i := \sum_{k=1}^{K} \left(H_{ii}^{(k)} - \sum_{j \in J^{+}} H_{ij}^{(k)} \frac{d_j}{d_i} + \sum_{j \in J^{-}} H_{ij}^{(k)} \frac{d_j}{d_i} - \sum_{j \in J^{0}} \gamma_{ij} H_{ij}^{(k)} \frac{d_j}{d_i} \right) \boldsymbol{p}_k - \sum_{j \in J^{0}} \beta_{ij} \frac{d_j}{d_i}.$$

3. Linear parametric Hessians by slope expansion

In the previous section, we generalized the interval Hessian matrix to a linear parametric one. Now, we utilize slope expansion [4, 10, 11] of Hessian matrix entries to get a linear parametric-like form and apply the above method to calculate α_i, $i = 1, \dots, n$.

Let

$$f_{ij}(x) \in f_{ij}(a) + \boldsymbol{S}_{ij}(\boldsymbol{x}, a)^{T}(x - a)$$

be a slope expansion of the Hessian entries, where $\boldsymbol{S}_{ij}(\boldsymbol{x}, a) \in \mathbb{IR}^{n}$, and let us adapt the approach from Section 2. We take $x \in \boldsymbol{x}$ as parameters $p \in \boldsymbol{p}$ and associate the absolute term $f_{ij}(a)$ with the degenerate interval $[1, 1]$. Even though the coefficients $\boldsymbol{S}_{ij}(\boldsymbol{x}, a)$ are intervals now, the results are directly applicable to this case. Therefore, we compute

$$\alpha_i := \max\left\{0, -\tfrac{1}{2}\underline{h}_i\right\},$$

where

$$\boldsymbol{h}_i := \left(\boldsymbol{S}_{ii}(\boldsymbol{x}, a) - \sum_{j \in J^+} \boldsymbol{S}_{ij}(\boldsymbol{x}, a) \frac{d_j}{d_i} + \sum_{j \in J^-} \boldsymbol{S}_{ij}(\boldsymbol{x}, a) \frac{d_j}{d_i} - \sum_{j \in J^0} \gamma_{ij} \boldsymbol{S}_{ij}(\boldsymbol{x}, a) \frac{d_j}{d_i} \right)^T (\boldsymbol{x} - a)$$
$$+ f_{ii}(a) - \sum_{j \in J^+} f_{ij}(a) \frac{d_j}{d_i} + \sum_{j \in J^-} f_{ij}(a) \frac{d_j}{d_i} - \sum_{j \in J^0} (\gamma_{ij} f_{ij}(a) + \beta_{ij}) \frac{d_j}{d_i}.$$

4. Conclusion

We generalized the classical αBB method to utilize linear parametric structure of an enclosure of the Hessian matrix, and we applied this approach by using slope extensions of the Hessian entries. Since the slope form depends on the center of linearization a, we plan to discuss different choices and their effect on the underestimator. We will also present some numerical examples showing that the linear parametric form can sometimes significantly reduce the overestimation of the lower bound of the objective function.

Acknowledgments

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

References

[1] O. Beaumont. Solving interval linear systems with linear programming techniques. *Linear Algebra Appl.*, 281(1-3):293–309, 1998.

[2] C.A. Floudas. *Deterministic global optimization. Theory, methods and applications*, volume 37 of *Nonconvex Optimization and its Applications*. Kluwer, Dordrecht, 2000.

[3] C.A. Floudas and P.M. Pardalos, editors. *Encyclopedia of optimization. 2nd ed.* Springer, New York, 2009.

[4] E.R. Hansen and G.W. Walster. *Global optimization using interval analysis*. Marcel Dekker, New York, second edition, 2004.

[5] M. Hladík. Enclosures for the solution set of parametric interval linear systems. *Int. J. Appl. Math. Comput. Sci.*, 22(3):561–574, 2012.

[6] M. Hladík. Bounds on eigenvalues of real and complex interval matrices. *Appl. Math. Comput.*, 219(10):5584–5591, 2013.

[7] M. Hladík. The effect of Hessian evaluations in the global optimization αBB method. preprint, http://arxiv.org/abs/1307.2791, 2013.

[8] M. Hladík. On the efficient Gerschgorin inclusion usage in the global optimization αBB method. *J. Glob. Optim.*, 2014. DOI 10.1007/s10898-014-0161-7.

[9] M. Hladík, D. Daney, and E.P. Tsigaridas. Bounds on real eigenvalues and singular values of interval matrices. *SIAM J. Matrix Anal. Appl.*, 31(4):2116–2129, 2010.

[10] R.E. Moore, R.B Kearfott, and M.J. Cloud. *Introduction to interval analysis*. SIAM, Philadelphia, PA, 2009.

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

[12] A. Skjäl and T. Westerlund. New methods for calculating αBB-type underestimators. *J. Glob. Optim.*, 58(3):411–427, 2014.

[13] M. Zimmer, W. Krämer, and E.D. Popova. Solvers for the verified solution of parametric linear systems. *Comput.*, 94(2-4):109–123, 2012.

A probabilistic algorithm for L-infinity norm solution of under-determined algebraic systems of linear equation

M. M. Ali[1,2], Adam Earle[1], and Dario Fanucchi[1]

[1] *School of Computational and Applied Mathematics, University of the Witwatersrand, South Africa*

[2] *TCSE, Faculty of Engineering and Built Environment, University of the Witwatersrand*

Abstract We propose a new algorithm for the solution of minimum infinity solution of under-determined systems. It is a primal method like the one exists in the current literature but it is decidedly more in the spirit of a dual method. It is geometrically and conceptually clear and provides important new insight into the nature of the problem. The method is premised on the observation that at minimum there are at least $n - (m - 1)$ elements equal in absolute value and that these elements are maximal. The algorithm is thus logically divided into two parts; firstly a solution with $n - (m - 1)$ elements equal is absolute value and maximal is obtained, and secondly the location of those elements is changed in such a way as to reduce the infinity norm at each step. Heuristics are used to identify an index set corresponding to the initial solution.

Keywords: Path following algorithm, Directed acyclic graph, primal formulation, Index set, Line search procedure, Heuristics

1. Introduction

It goes without saying that the solutions to system of linear equations are of paramount importance in almost every field of science, engineering and management. Throughout the text we will consider the following linear system

$$Ax = b,\, x \in R^n,\; A \in R^{m\times n},\; b \in R^m. \tag{1}$$

In practice, it is common to encounter systems that do not admit a unique solution, rather the system is consistent where the number of variables exceeds the number of equations - in which case an infinite number of solutions exist, or the system is inconsistent and no solutions exist. In the former case the system is called under-determined and in the latter the system is called over-determined. When the system is under-determined we have in general $m < n$, and practitioners seek to pick the solution that is best suited to their needs. Minimum infinity norm solutions are often chosen for various practical problems. For example, control theoreticians have found application for the minimum infinity norm solution in terms of kinematically redundant robots. This describes the situation in which robotic manipulators have an excess number of degrees of freedom and thus an infinity number of possible solution paths. The path that the control theoreticians would like to choose is the one that minimizes the maximum dependence on any one joint. This is formulated mathematically as

$$\text{minimize} \quad \|x\|_\infty \quad \text{subject to} \quad Ax = b. \tag{2}$$

The problem is to minimize the maximum value of elements in the solution vector. Such a solution is chosen when one seeks to minimize the maximum load on any node of a given system. In particular, such solutions are sought in control theory when the limitations of any individual component of a system cannot be breached.

The current best algorithms for the solution to problem (2) are given in [1, 2, 3, 4]. These are the path following algorithm was proposed by Cadzow [1] in which the polyhedral structure of the objective function of the dual was exploited. The algorithm is premised on the observation that an optimum solution to the augmented dual problem must contain a minimum number of element equal to zero. Abdelmalek [2] some years later proposed a linear programming formulation of the dual problem in which a modified simplex method was applied to a reduced tableu – this being made possible by the strong symmetries present in the constraint matrix. Shim and Yoon [3] almost two decades later proposed a primal method which they claim is conceptually and geometrically clear at the cost of computational inferiority to both of the methods already mentioned. Their method is based on geometrical consideration of when the level curves of the objective function first touch the solution space [3]. The polyhedral nature of the objective function is exploited in the above methods. The fundamental ideas of the path following method were later replicated in a 2002 paper by Ha and Lee [4] in which a computationally inferior but geometrically clearer algorithm was presented. We propose a primal path following method to deal with the problem (2). Unlike the dual approach suggested in [1], our method is based on the primal formulation and yet conceptually different from the primal method in [3].

2. Solution procedure of the primal method

The basic features of the proposed solution method for problem (2) will be now presented. It is a primal method like that of the Shim-Yoon method [3] but it is decidedly more in the spirit of Cadzow's path following algorithm [1]. It is geometrically and conceptually clear and provides important new insight into the nature of the problem. The method is premised on the observation that at the minimum at least $n-(m-1)$ elements of the solution vector are equal in absolute value and that these elements are also maximal. This assertion may be deduced from the alignment criteria between the solution to the primal and the dual problem, as previously described in Cadzow's method, and the fact that an optimal solution to the dual problem contains at least $m-1$ zeros [1]. An alternative proof is given in [4], although this result has long been known.

The algorithm is thus logically divided into two parts; in Part 1 a solution with $n-(m-1)$ elements equal is absolute value and maximal is obtained, and in Part 2 the location of those elements is changed in such a way as to reduce the infinity norm at each step. We shall frequently refer to the set of equal maximal elements, the following definition establishes some useful notation in this regard. The index set, which we will denote as Λ, is the set of all elements of a vector that are maximal in absolute value i.e. $\Lambda = \{i \,:\, |x_i| = \|x\|_\infty\}$. Also then we have the compliment as, $\Lambda^c = \{i \,:\, i \notin \Lambda\}$.

Part 1 iteratively generates the required solution using a number of heuristics procedures. The second part of the method iteratively minimizes the infinity norm in such a way that $\|x^{k+1}\|_\infty < \|x^k\|_\infty$ by bringing new element(s) into the index set Λ taking out element(s) from the index set into its complement Λ^c. Part 2 of the method also uses two heuristics procedure to reach the desired optimal.

An element previously in Λ^c is brought into the index set by means of a *line search* procedure; $x^{k+1} = x^k + \alpha^k d^k$ where d^k denotes the direction vector at iteration k and α^k the step length. The direction vector is chosen so as to maintain the elements in Λ while reducing their absolute value, $|x_i^{k+1}| < |x_i^k|\ \forall i \in \Lambda$. We shall henceforth understand x_Λ to the set of all element of x in the index set Λ, $x_\Lambda = \{x_i : |x_i| = \|x\|_\infty\}$, similarly for x_{Λ^c}. The process by which elements in Λ^c are brought into Λ is fundamentally the same regardless of whether we are in Part 1 or Part 2 of the algorithm.

We have proposed criteria for the descent condition and hence the stopping condition of the algorithm. A probabilistic scheme is suggested which allows more than one heuristic to be employed in an iteration of the algorithm.

In particular, our method employs novel heuristics coupled with ideas that have been exploited for the over-determined system. Heuristics at both parts are integrated probabilistically showing much improved results against when individual heuristics are employed. Results and comparisons with the existing method will also be shown.

References

[1] Cadzow, J.A. "An efficient algorithmic procedure for obtaining a minimum L-infinity norm solution to a system of consistent linear equations". SIAM Journal on Numerical Analysis, 11(6): pp. 1151–1165, 1974.

[2] Abdelmalek, N.N. "Minimum l-infinity solution of underdetermined systems of linear equations. Journal of Approximation Theory", 20(1): pp. 57–69, 1977.

[3] Shim, I.C. and Yoon, Y.S. "Stabilized minimum infinity norm torque solution for redundant manipulators". Robotica, 16(2):pp. 193–205, 1998.

[4] Insso Ha, I. and Lee, J. "Analysis on a minimum infinity norm solution for kinematically redundant manipulators". ICASE, 4(2): pp. 130–139, 2002.

On Fractional Quadratic Problems*

Paula Amaral[1]

[1]*FCT UNL, Portugal,* paca@fct.unl.pt

Abstract Constrained Fractional Quadratic Problems consists in the minimization of the ratio of two quadratic functions, over a set of constrains. In this paper we address the particular case of linear constraint and so the problem we are dealing with is the minimization of a fractional quadratic function over a polytope. This problem has important applications such as the the correction of inconsistent linear systems and the eigenvalue complementarity problems. The nonconvexity of the objective function poses difficulties in finding global optimal solutions. In this paper we will present some results that contribute to the development of global optimization methods, including Completely Positive and Copositive formulations. Some preliminary ideas for addressing the Standard Fractional Quadratic Problem are also presented.

Keywords: Fractional Quadratic Problems, Copositive, Inconsistent systems

1. Introduction

Consider

$$\begin{aligned} \text{CFQP:} \quad \min \quad & \frac{x^T C x}{x^T B x} = f(x) \\ \text{s.t.} \quad & Ax = a \\ & x \geq 0 \end{aligned} \tag{1}$$

The difficulty in the minimization CFQP arises from the nonconvexity of the objective function. The interest is due to the many applications, such as the Constrained Total Least Squares Problem (CTLSP) in which context some valid approaches have been presented.

Data fitting, the adjustment of a function to a set of points minimizing the squares of the errors, and approximation solutions to overdetermined linear systems are defined as Least Squares Problem (LSP). These problems are well studied and efficient methods are known. However, if additional assumptions or constraints are introduced then solving these problems is more difficult. Total Least Squares Problem (TLSP) is a LSP with additional assumption of corruptness of the data as well as the output. This problem is still manageable but the introduction, in addition, of further constraints such non negativity constraints, turns the problem into a constrained TLSP (CTLSP) which is a difficult optimization problem.

There are some important subclasses of the CTLSP, such as the Regularized Total Least Squares Problem (RTLSP), where an additional quadratic constraint (Tikhonov regularization) is considered to ensure solution stability, [1], [2], [3], [4], [5].

Another generalization of the TLSP is related with the minimal correction of inconsistent linear systems, in presence of inequality constraints, and the minimal correction is defined by the minimization of the Frobenius norm of the perturbations of the matrix of coefficients and the independent term. In [6] the proof that this problem can be formulated as a fractional quadratic program (FQP) is presented. If we consider only equality constraints then the problem is equivalent to the TLSP [7]. The introduction of inequalities in the linear system

[*] CMA FCT UNL

makes the problem much harder [8], and in [6] a branch-and-bound approach based on the Reformulation Linearization Technique (RLT) for finding lower bounds was used.

Another interesting application of the FQP is the Eigenvalue Complementary Problem (EiCP) with symmetric real matrices. Finding a complementary eigenvalue reduces to finding a stationary point of the Rayleigh quotient on the simplex [9]. This problem has several applications in engineering and physics, as for instance, in the study of resonance frequency of structures and stability of dynamical systems [10].

For the general quadratic fractional problem for small-scale problems a method that combines the classical Dinkelbach method and a branch-and-bound approach for the nonconvex quadratic problem is presented in Konno [11] and Yamamoto and Konno [12] proposed an exact algorithm combining the classical Dinkelbach approach and an integer optimization formulation.

Quadratically constrained quadratic problems [13], [14], [15], [16] are equivalent to a particular subclass of constrained fractional quadratic problems [13], when the linear constraints are homogeneous. However, it seems that departure from homogeneity yields more complications, at least if there are more than just one constraints.

2. Copositivity and constrained fractional quadratic problems

Let

$$\mathcal{M}_n = \left\{ A \text{ an } n \times n \text{ matrix} : A^\top = A \right\}$$

be the cone of symmetric matrices. With respect to duality, the dual cone of the copositive matrices

$$\mathcal{C}_n = \left\{ C \in \mathcal{M}_n : \mathbf{x}^\top C \mathbf{x} \geq 0 \text{ for all } \mathbf{x} \in \mathbb{R}^n_+ \right\}$$

is the cone of completely positive matrices

$$\mathcal{C}^*_n = \left\{ D \in \mathcal{M}_n : D = YY^\top, Y \text{ an } n \times k \text{ matrix with } Y \geq O \right\} .$$

Let $\mathcal{P}_n \subset \mathcal{M}_n$ be cone of symmetric psd $n \times n$ matrices and $\mathcal{N}_n \subset \mathcal{M}_n$ be the cone of nonnegative symmetric matrices. It is known that $\mathcal{K}_0 = \mathcal{P}_n + \mathcal{N}_n$ provides a approximation of the copositive cone $\mathcal{C}_n$ in the sense of $\mathcal{K}_0 \subseteq \mathcal{C}_n$. Since $\mathcal{P}_n$ and $\mathcal{N}_n$ are self-dual cones we have

$$\mathcal{C}^*_n \subseteq \mathcal{K}^*_0 = (\mathcal{P}_n + \mathcal{N}_n)^* = \mathcal{P}_n \cap \mathcal{N}_n \,.$$

The latter matrix cone is also called the cone of doubly nonnegative matrices, and sometimes denoted by $\mathcal{D}_n$.

Preisig's article [13], in to the best of our knowledge, is the only reference where copositivity is explicitly used for finding the global solution to the FQP, where an iterative procedure for which convergence to a KKT point of the StFQP can be proved, provided that B is both *positive-semidefinite* (psd) and strictly copositive. However, no information was provided on the quality of the solution found by this algorithm, and thus even for StFQP this method cannot be considered complete from a global optimization perspective.

In [17] strong lower bounds based on SDP relaxations for the CFQP were presented. First constructing matrices

$$\overline{A} = \begin{bmatrix} \mathbf{a}^\top \mathbf{a} & -\mathbf{a}^\top A \\ -A^\top \mathbf{a} & A^\top A \end{bmatrix}, \quad \overline{B} = \begin{bmatrix} 0 & 0^\top \\ 0 & B \end{bmatrix}, \quad \overline{C} = \begin{bmatrix} 0 & 0^\top \\ 0 & C \end{bmatrix}. \tag{2}$$

and considering that

$$A\mathbf{x} = \mathbf{a} \quad \Longleftrightarrow \quad \overline{A}\mathbf{z} = [-\mathbf{a}\,,\, A]\mathbf{z} = \quad \Longleftrightarrow \quad \mathbf{z}^\top \overline{A} \mathbf{z} = 0\,,$$

where $\mathbf{z} = [1\,,\,\mathbf{x}^\top]^\top \in \mathbb{R}^{n+1}$, (1) was rephrased as

$$\psi = \min\left\{\frac{\mathbf{z}^\top\overline{C}\mathbf{z}}{\mathbf{z}^\top\overline{B}\mathbf{z}} : \mathbf{z} \in \mathbb{R}^{n+1}_+,\ \mathbf{z}_1 = 1,\ \mathbf{z}^\top\overline{A}\mathbf{z} = 0\right\}. \tag{3}$$

Then putting $Z = \mathbf{z}\mathbf{z}^\top$, rewriting $\mathbf{z}^\top\overline{A}\mathbf{z} = \overline{A} \bullet Z$, with $\overline{A}$ psd and observing that $Z_{11} = z_1^2$ and $\mathbf{z} \in \mathbb{R}^{n+1}_+$, we have

$$\psi = \min\left\{\frac{\overline{C} \bullet Z}{\overline{B} \bullet Z} : Z_{11} = 1\,,\ \overline{A} \bullet Z = 0\,,\ \operatorname{rank}(Z) = 1\,,\ Z \in \mathcal{C}^*_{n+1}\right\}. \tag{4}$$

By homogeneity, for any Z feasible to (4), constraint $Z_{11} = 1$ can be replace the by $Z_{11} > 0$. Defining $X = \frac{1}{\overline{B}\bullet Z} Z \in \mathcal{C}^*_{n+1}$ which also has rank one with $X_{11} > 0$ and satisfies $\overline{B} \bullet X = 1$, in [17], the following equivalent problem was obtained:

$$\psi = \min \quad \left\{\overline{C} \bullet X : \overline{B} \bullet X = 1,\ \overline{A} \bullet X = 0,\ \operatorname{rank}(X) = 1, X_{11} > 0,\ X \in \mathcal{C}^*_{n+1}\right\}. \tag{5}$$

It was also proved that the strict linear inequality and the (non-convex) rank-one constraint could be dropped to obtain the equivalent problem

$$\min\left\{\overline{C} \bullet X : \overline{B} \bullet X = 1, \overline{A} \bullet X = 0, X \in \mathcal{C}^*_{n+1}\right\}. \tag{6}$$

In conclusion, in [17] it was proved that,

$$\begin{aligned} \psi &= \min\left\{f(\mathbf{x}) = \frac{\mathbf{x}^\top C\mathbf{x}}{\mathbf{x}^\top B\mathbf{x}} : A\mathbf{x} = \mathbf{a}, \mathbf{x} \in \mathbb{R}^n_+\right\} && (7)\\ &= \min\left\{\overline{C} \bullet X : \overline{B} \bullet X = 1, \overline{A} \bullet X = 0, X \in \mathcal{C}^*_{n+1}\right\}. && (8)\end{aligned}$$

Checking condition $X \in \mathcal{C}^*_{n+1}$ is (co-)NP-hard but exploiting (7) and using the inclusion $\mathcal{C}^*_{n+1} \subseteq \mathcal{D}_{n+1} = \mathcal{P}_{n+1} \cap \mathcal{N}_{n+1}$ a lower bound for the CFQP was proposed by solving

$$\psi_{\text{cop}} = \min\left\{\overline{C} \bullet X : \overline{B} \bullet X = 1, \overline{A} \bullet X = 0, X \in \mathcal{D}_{n+1}\right\}. \tag{9}$$

3. Standard Fractional Quadratic Program

In the section we will present some preliminary ideas for the construction of method for the global optimization of the Standard Fractional Quadratic Program

$$\begin{aligned} \text{SFQP:}\quad \min\quad & \frac{x^T Cx}{x^T Bx}\\ \text{s.t.}\quad & e^T x = 1\\ & x \geq 0. \end{aligned} \tag{10}$$

Let $\Delta = \left\{x \in \mathbb{R}^n : e^T x = 1, x \geq 0\right\}$ be the ordinary simplex. Then the global minimum of SFQP is the complementary eigenvector $\bar{x}$ associated with maximum complementary eigenvalue $\bar{\lambda}$ of $EiCP(B, -A)$. So SFQP can be solved by a modification of an algorithm for finding all the complementarity eigenvalues with increasing values. Based on this property it is possible to develop methods for the global optimization of (10) but this is currently under research.

4. Conclusions

In this paper we present copositive exact formulations for the CFQP. The practical interest in these problems is discussed, with emphasis on the correction of inconsistent linear systems. Based on these formulations SDP relaxations are proposed providing good lower bounds. Theoretical results presented in this paper have important implications in the computation of lower bounds for the CFQP and in the development global optimization approaches, for instance Branch and Bound methods.

References

[1] G. H. Golub, P. C. Hansen, and D. P. O'Leary. Tikhonov regularization and total least squares. *SIAM Journal of Matrix Analysis and Applications*, 21:185–194, 1999.

[2] D.M. Sima, S. Van Huffel, and G.H. Golub. Regularized total least squares based on quadratic eigenvalue problem solvers. *Bit Numerical Mathematics*, 44(4):793–812, 2004.

[3] R. Renaut and A.H. Guo. A regularized total least squares algorithm. In S. Van Huffel and P. Lemmerling, editors, *Total Least Squares and Errors-in-Variables Modeling: Analysis, Algorithms and Applications*, pages 57–66. Kluwer Academic, 2002.

[4] R. Renaut and A.H. Guo. Efficient algorithms for solution of regularized total least squares. *SIAM Journal on Matrix Analysis and Applications*, 26(2):457–476, 2005.

[5] A. Beck, A. Ben-Tal, and M. Teboulle. Finding global optimal solution for a quadratically constrained fractional quadratic problem with applications to the regularized total least squares. *SIAM Journal on Matrix Analysis and Applications*, 28:425–445, 2006.

[6] P. Amaral, J. Júdice, and H. D. Sherali. A reformulation–linearization–convexification algorithm for optimal correction of an inconsistent system of linear constraints. *Computers and Operations Research*, 35:1494–1509, 2008.

[7] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

[8] P. Amaral and P. Barahona. Connections between the total least squares and the correction of an infeasible system of linear inequalities. *Linear Algebra and Applications*, 395:191–210, 2005.

[9] M.G. Queiroz, J.J. Júdice, and C. Humes, Jr. The symmetric eigenvalue complementarity problem. *Mathematics of Computation*, 73:1849–1863, 2004.

[10] A.P. Costa, I.N. Figueiredo, J. Júdice, and J.A.C. Martins. The directional instability problem in systems with frictional contacts. *Computer Methods in Applied Mechanics and Engineering*, 193:357ï£¡384, 2004.

[11] J-Y. Gotoh and H. Konno. Maximization of the ratio of two convex quadratic functions over a polytope. *Computational Optimization and Applications*, 20(1):43–60, 2001.

[12] R. Yamamoto and H. Konno. An efficient algorithm for solving convex-convex quadratic fractional programs. *Journal of Optimization Theory and Applications*, 133(2):241–255, 2007.

[13] J.C. Preisig. Copositivity and the minimization of quadratic functions with nonnegativity and quadratic equality constraints. *SIAM Journal on Control and Optimization*, 34(4):1135–1150, 1996.

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

[15] J. Linderoth. A simplicial branch-and-bound algorithm for solving quadratically constrained quadratic programs. *Mathematical Programming*, 103:251–282, 2005.

[16] K. M. Anstreicher. Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming. *Journal of Global Optimization*, 43:471–484, 2009.

[17] P. Amaral, I.M. Bomze, and J. Júdice. Copositivity and constrained fractional quadratic problems. *Mathematical Programming (to appear)*.

A mixed integer linear programming heuristic for computing nonstationary (s,S) policy parameters

Roberto Rossi[1], Onur A. Kilic[2], and S. Armagan Tarim[2]

[1] *Business School, University of Edinburgh, Edinburgh, United Kingdom,* roberto.rossi@ed.ac.uk

[2] *Institute of Population Studies, Hacettepe University, Ankara, Turkey* {onuralp,armagan.tarim}@hacettepe.edu.tr

Abstract In this work we present a novel MILP based heuristic for computing nonstationary (s,S) policy parameters. This approach presents advantages with respect to other existing methods, since it is easy to implement and features narrower optimality gaps.

Keywords: stochastic lot sizing, (s,S) policy, nonstationary demand, mixed integer linear programming, heuristic

1. Introduction

The stochastic lot sizing problem consists in controlling an inventory system facing random demand over a given planning horizon. The decision maker faces inventory holding costs, if she orders too much; and backorder penalty costs, if she orders too little and demand fulfilment is delayed until the next replenishment arrives. Each time production runs there are fixed and variable production/ordering costs that must be accounted for while controlling the system. The structure of the optimal control policy to this problem has been characterised — under very mild assumptions — over fifty years ago [8]. This control policy, named (s,S), is surprisingly simple; this policy monitors the inventory position, i.e. on hand stock minus backorders plus incoming orders, and issues an order to bring the inventory position up to S whenever the inventory position falls below s.

As pointed out by [4] incorporating more realistic assumptions about product demand constitutes an important research direction in inventory theory. The ability to model and control a nonstationary demand process is essential in practical settings, since only very few businesses actually face stationary demand, while most products are subject to demand processes that evolve over time with frequent changes in their directions and rates of growth or decline.

When demand is nonstationary an (s,S) policy is still cost optimal. However, computing optimal control parameters for this policy constitutes a hard combinatorial task. Standard pseudo-polynomial dynamic programming (DP) algorithms can only tackle small instances. This motivates the investigation of effective heuristics. To date, there are only two established heuristics for computing optimal control policy parameters under nonstationary demand [1, 2]. Unfortunately, these heuristics present a number of drawbacks. They are not easy to implement, since they require dedicated code. Furthermore, in a recent study [3], their respective optimality gap on a large test bed has been found to average 4% to 5%. The same work also demonstrated that approaches such as [9, 6], despite implementing heuristics for control policies that are theoretically inferior to a nonstationary (s,S) policy, feature much lower optimality gaps, i.e. around 1.5%, on the same test bed. This demonstrates that further research is needed to develop more effective heuristics for computing nonstationary (s,S) control policy parameters.

In this work we develop an MILP based heuristic for computing nonstationary (s,S) policy parameters. The key insight upon which sour approach is based comes from the study in [3], which showed that a nonstationary (R,S) policy often performs very close to optimal. The idea is then to use an existing MILP model for computing nonstationary (R,S) policy parameters [5] as a proxy to determine near optimal (s,S) policy parameters. Our heuristic is easy to implement, since it is based solely on a standard MILP model and on a simple binary search procedure. It performs better than other existing approaches, featuring an average optimality gap of 0.2% on our preliminary tests.

2. The stochastic lot sizing problem

The finite-horizon single-item single-stocking location nonstationary stochastic lot sizing problem as introduced in [8] can be formalised as follows. We consider a finite planning horizon of n periods. Customer demand d_t in each period $t = 1, \dots, n$ is a random variable with known probability distribution. There are three types of costs: a nonlinear purchasing or ordering cost c(z), where z is the amount purchased, which takes the general form

$$c(z) = \begin{cases} K + vz & \text{if } z > 0 \\ 0 & \text{otherwise} \end{cases}$$

where K and v denote the fixed and variable purchasing/ordering cost components, respectively; a holding cost of h is paid of each unit of inventory carried from one period to the next; and a shortage cost p which is paid for each unit of demand backordered at the end of a period. Holding and shortage costs are charged at the end of a period. Ordering costs are charged when a purchase is made. Without loss of generality, see [8], delivery of an order is immediate.

Let y denote the stock level immediately after purchases are delivered, the expected holding and shortage cost for a generic period are given by

$$L(y) = \begin{cases} \int_0^y h(y-\omega)g(\omega)\mathrm{d}\omega + \int_y^\infty p(\omega - y)g(\omega)\mathrm{d}\omega & y \geq 0 \\ \int_0^\infty p(\omega - y)g(\omega)\mathrm{d}\omega & y < 0 \end{cases}$$

where $g_t(\cdot)$ denotes the probability density function of the demand in period t. If the initial inventory at the beginning of the planning horizon is x and $C_n(x)$ represents the expected total cost over the n-periods planning horizon if provisioning is done optimally then $C_n(x)$ satisfies

$$C_n(x) = \min_{y \geq x} \left\{ c(y-x) + L_n(y) + \int_0^\infty C_{n-1}(y-\omega)g_n(\omega)\mathrm{d}\omega \right\}$$

If $y_n(x)$ is the argument minimising the above functional equation, then $y_n(x) - x$ denotes the optimal initial purchase.

3. (s,S) policy

As shown in [8], the optimal control policy for the problem introduced in Section 2 takes a surprisingly simple form. The result stems from the study of the following function

$$G_n(y) = cy + L_n(y) + \int_0^\infty C_{n-1}(y-\omega)g_n(\omega)\mathrm{d}\omega$$

More specifically, Scarf proved that $G_n(y)$ is K-convex.

Definition 1. *Let $K \geq 0$, and let $f(x)$ be a differentiable function, $f(x)$ is K-convex if*

$$K + f(a+x) - f(x) - af'(x) \geq 0$$

for all positive a and all x. This definition can be extended to a non differentiable function.

It follows that, under general nonstationary settings, the optimal policy can be described via n pairs (s_i,S_i), where s_i denotes the reorder point and S_i the order-up-to-level for period i. In practice, S_n denotes the absolute minimum of $G_n(y)$ and $s_n < S_n$ is the unique value such that $K + G_n(S_n) = G_n(s_n)$. The fact that $G_n(y)$ is K-convex ensures that ripples in the above nonlinear function do not affect the existence of a unique reorder point $s_n \leq S_n$, since their height will never exceed K.

We shall now illustrate graphically the notion of K-convexity on a simple numerical example. Consider a planning horizon of $n = 4$ period and a demand d_t normally distributed in each period t with mean $\mu_t \in \{20, 40, 60, 40\}$, for period $t = 1, \dots, n$ respectively. The standard deviation σ_t of the demand in period t is equal to $0.25\mu_t$. Other problem parameters are $K = 100$, $h = 1$ and $p = 10$; to better conceptualise the example we let $v = 0$. In Fig. 1 we plot $G_n(y)$ for an initial inventory $y \in (0, 200)$. In period one, when the opening inventory level y falls below 14 it is convenient to pay the fixed ordering cost K to increase available inventory to S_n, i.e. $K + G_n(S_n) = G_n(s_n) \leq G_n(y)$. Comparable graphs can be produced for all other periods.

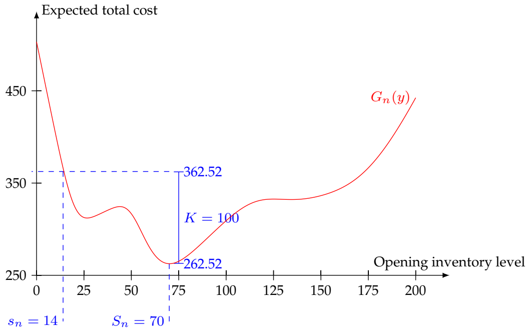


Figure 1: Plot of $G_n(y)$

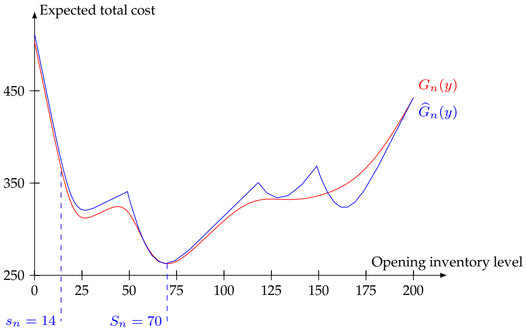


Figure 2: $G_n(y)$ vs $\widehat{G}_n(y)$

4. (R,S) policy

A widely adopted control policy, alternative to the (s,S) policy, is the (R,S) policy. In this policy, all replenishment periods must be fixed at the beginning of the planning horizon; however, the decision maker can decide upon the actual order quantity just before issuing a replenishment. Under a nonstationary settings this policy takes the form (δ_i,S_i), where δ_i is a binary variable that is set to 1 if a replenishment is scheduled in period i and S_i denotes the order-up-to-level associated with a replenishment that occurs in period i. [5] developed a mixed integer linear programming model to compute near-optimal (R,S) policy parameters. To model nonlinear expected holding and shortage costs the authors exploit piecewise linear upper and lower bounds of the first order loss function [7]. An interesting feature of the model in [5] is the fact that, despited being explicitly developed for the (R,S) policy, it can nevertheless be used as a "proxy" to the expected total cost of an (s,S) policy, i.e. $G_n(y)$. In fact, we can first observe that the optimal expected total cost and the order-up-to-level for period one returned by the model for an initial stock level of x units are tight approximation to $C_n(x)$ and $y_n(x) = S_n$, respectively. Furthermore, if we set $\delta_1 = 0$ — i.e. we do not schedule any replenishment at the beginning of the planning horizon — since $G_n(y)$ is K-convex, there is a unique reorder point $s_n < S_n$ such that $K + G_n(S_n) = G_n(s_n)$. We can therefore exploit a binary search on $y < S_n$ to find s_n. In the binary search procedure, the cost associated with a given opening inventory level y can be approximated using $\widehat{G}_n(y)$, the solution of the

Table 1: Optimal policy for the numerical example; expected total cost (ETC) estimated at 95% confidence.

	SDP — ETC: (362.2,362.9)		MILP — ETC: (363.0,363.1)	
t	S_t	s_t	S_t	s_t
1	70.0	14.0	70.2	15.0
2	141	29.5	53.9	29.0
3	113	58.0	116	58.1
4	53.5	28.5	53.9	29.0

MILP model. We then repeat this procedure to find S_i and s_i for each period $i = 1, \ldots, n$, by analysing $\widehat{G}_n(y), \widehat{G}_{n-1}(y), \ldots$.

In Fig. 2, for the numerical example previously discussed, we plot $\widehat{G}_n(y)$, obtained via the MILP model in [5], when we vary y, i.e. the opening inventory level at the beginning of the planning horizon. We also compare it to the plot of $G_n(y)$ obtained via a standard DP approach. The optimal policy found via DP for the above example is contrasted in Table 1 against the policy obtained via the MILP heuristic.

5. Summary

In this paper we presented an MILP based heuristic for computing nonstationary (s,S) policy parameters. Preliminary computational results over a large set of instances reveal that the average optimality gap observed for our approach is 0.2%. In contrast, on the same test bed [1, 2] feature optimality gaps of 2.09% and 3.52%, respectively; however, these latter heuristics are faster than ours. A complete analysis of these results will be discussed in the full paper.

References

[1] R.G. Askin. A procedure for production lot sizing with probabilistic dynamic demand. *AIIE Transactions*, 13(2):132–137, June 1981.

[2] S. Bollapragada and T.E. Morton. A simple heuristic for computing non-stationary (s, S) policies. *Operations Research*, 47(4):576–584, 1999.

[3] G. Dural-Selcuk, O.A. Kilic, S.A. Tarim, and R. Rossi. A comparison of lot sizing methods for non-stationary stochastic demands, December 2013.

[4] S.C. Graves. A Single-Item Inventory Model for a Non-Stationary Demand Process. *Manufacturing & Service Operations Management*, 1:50–61, 1999.

[5] R. Rossi, O.A. Kilic, and S.A. Tarim. Piecewise linear approximations for the static-dynamic uncertainty strategy in stochastic lot-sizing, December 2013.

[6] R. Rossi, S.A. Tarim, B. Hnich, and S. Prestwich. Constraint programming for stochastic inventory systems under shortage cost. *Annals of Operations Research*, 195(1):49–71, July 2012.

[7] R. Rossi, S.A. Tarim, S. Prestwich, and B. Hnich. Piecewise linear lower and upper bounds for the standard normal first order loss function. *Applied Mathematics and Computation*, 231:489–502, March 2014.

[8] H.E. Scarf. Optimality of (s, S) policies in the dynamic inventory problem. In K. J. Arrow, S. Karlin, and P. Suppes, editors, *Mathematical Methods in the Social Sciences*, pages 196–202. Stanford University Press, Stanford, CA, 1960.

[9] S.A. Tarim and B.G. Kingsman. Modelling and computing (Rn,Sn) policies for inventory systems with non-stationary stochastic demand. *European Journal of Operational Research*, 174(1):581–599, October 2006.

Design of Space Thrusters: a Topology Optimization Problem solved via a Branch and Bound Method

Satafa Sanogo and Frédéric Messine

Université de Toulouse, Laplace (CNRS UMR5213, Toulouse INP), F- 31071 Toulouse,
{ sanogo, messine}@laplace.univ-tlse.fr

Abstract In this work, an exact Branch and Bound algorithm has been applied to a practical problem. This optimization problem arising in the design of space thrusters, is hard to solve mainly because the objective function to be minimized is implicit and must be computed by using a finite element code. In a previous paper, we implement a method based on local search algorithms and we then proved that this problem is non convex yielding a strong dependence on the starting points. In this paper by posing an hypothesis of monotonicity that we validated numerically, we provided properties making it possible the computation of lower bounds and some improvements of the convergence of such a Branch and Bound code. Two numerical examples show the efficiency of the approach.

Keywords: Maxwell's PDE, Topology Optimization, SIMP method, Branch and Bound.

1. Introduction

In the field of space propulsion, the electric propulsion constitutes an interesting technology compared to the chemical one. Indeed, the weight and volume of the total system including the thruster and its corresponding fuel is considerably decreased. Among the electric propulsion systems, the Hall effect thrusters are more and more used on board of telecommunication satellites, mainly for keeping some geostationary positions. This technology seems to be not studied so far since the seventies in Russian laboratories.

In this work, we try to find the structure of some zones which can provide an imposed magnetic field inside an objective zone; in Figure 1, Ω_v corresponds to the variable area and Ω_T is the zone where the magnetic field must be approximate, yielding a least square optimization problem. Thus, the design of spatial plasma thrusters can be understood and formulated as a topology optimization problem where the variable areas will be discretized in small subdomains where the value will take 0 for void and 1 for iron providing a large scale non linear discrete optimization problem.

The difficulty of this problem is that the objective function is not explicit but as to be computed via the resolution of Maxwell's partial derivative equations (PDE). And so, that constitutes one of the main difficulty of our optimization problem. In [1], we first solved this topology optimization problem by associating a penalization method (SIMP approach [2]) with local optimization based algorithms. This first code was developed in `MatLab` using `fmincon` subroutine and `FEMM` software to solve the PDE. We validate this approach on numerous examples reaching problems with 800 variables.

Nevertheless, as we shown in [1] this optimization problem is non-convex and therefore the optimal solutions depend strongly to the starting point given to the local solver `fmincon`. Thus, it becomes interesting to study this global optimization problem. The idea of this work is to develop a Branch and Bound code to solve exactly this least square problem. The main difficulty here is to deal with an implicit objective function which has to be computed via a finite element code to solve the Maxwell's PDE. This approach is based on an hypothesis which seems to be verified in our examples.

In Section 2, the problem formulation is detailed. The Maxwell's PDE are then presented. In Section 3, some properties are discussed in order to make it possible the use of a standard Branch and Bound code. In Section 4, some numerical results are presented by using a Branch and Bound code that we developed in `MatLab`. These results validate our approach and the hypothesis that we provide. Section 5 concludes.

2. Problem formulations

We consider the design domain depicted in Figure 1, the purpose is to minimize the discrepancy between the *expected magnetic flux distribution* B_0 and a computed value B in the target region Ω_{T}. The subsets Ω_{v1} and Ω_{v2} of the considered domain Ω are the variable areas. The design goal is then to distribute optimally the ferromagnetic material inside them. In order to impose more specifications on the expected results, a limited material quantity is fixed. This constraint is formulated in term of allowed volume V_0 of the design variable region. The design parameter is the magnetic permeability (μ) of the considered ferromagnetic material (here the iron). The computed magnetic field induction B is the curl of the vector potential A. This vector potential is called the state variable indeed it is solution of a Maxwell equation considered in the literature as state equation. The power source of the device to be manufactured is supplied from a fixed current density J. The density current J is provided by coil1 and coil2 (see Figure 1). Thus, our topology optimization problem can be formulated as follows:

$$(\wp)\left\{\begin{array}{l}\min\limits_{\mu,\,A} F(\mu, A) = \|B(A) - B_0\|^2,\\ \text{s.t. :}\\ \qquad -\frac{1}{\mu}\Delta A = J, \text{ in } \Omega, \text{ and } A \in H_0^1(\Omega),\ (\mathtt{I})\\ \qquad \mu \in \mathcal{P} := \{\mu \in L^\infty(\Omega) : \mu_{\min} \leq \mu \leq \mu_{\max},\ \int_\Omega \mu d\Omega = V_0\},\ (\mathtt{II})\end{array}\right.$$

where: $B(A) = \left(\frac{\partial A}{\partial y}, -\frac{\partial A}{\partial x}\right), \quad \forall\, A \in H_0^1(\Omega).$

For designing a structure, in particular a magnetic circuit with topological optimization method, we are interested in the determination of the optimal placement of a given isotropic material in space; i.e., we should determine which points of space should be material points and which points should remain void (no material). It follows that the problem becomes a "0-1problem" indeed we can set 1 for material points and 0 for void ones. A new variable denoted by ρ and called material density function in the literature is introduced to parameterize the distribution of the material in the design domain such that ρ equals to 1 for material points and 0 elsewhere. Then an interpolation scheme is used to express the magnetic permeability μ in function of the density function by the relation below:

$$\mu(\rho) = \mu_{\min} + (\mu_{\max} - \mu_{\min})\rho, \text{ with } \rho \in \{0, 1\}, \tag{1}$$

where $\mu_{\min}$ and $\mu_{\max}$ are the permeability of void and the predefined ferromagnetic material respectively.

A typical approach to solve numerically problem $(\wp)$, is to discretize the problem using finite element. We use a finite element method software FEMM to solve the PDE $(\mathtt{I})$ in $(\wp)$ to have the values of A in function of the density ρ. Thus, problem $(\wp)$ can be formulated depending on ρ thanks to equation (1) (note that μ depends also on ρ). The variable areas Ω_{v1} and Ω_{v2} are meshed coarsely. This mesh of the variable areas remains fixed throughout all the optimization process. Each cell of that mesh grid is associated to a design variable and corresponds to a component of ρ , thus inside each cell one must determine the material properties (ferromagnetic or void). Finally our topology optimization problem can be rewritten depending on the material density function under the following form:

$$(\wp_\rho)\begin{cases} \min\limits_{\rho} F(\rho) = \|\mathcal{B}(\rho) - B_0\|^2, \\ \quad \text{s.t. :} \\ \qquad h_A(\rho) = 0, \qquad (i) \\ \qquad \sum_{i=1}^{N} \rho_i = v_0, \quad (ii) \\ \qquad \rho \in \{0,1\}^N, \quad (iii) \end{cases}$$

where: equation (i) is the equivalent of the state equation by using material density function; equation (ii) is the volume constraint. N is the number of cells provided by discretizing the design variable domain. Note that since the mesh is regular and uniform each cell has the same volume and we put v that elementary volume and v_0= V_0/v. And $\mathcal{B}(\rho) := B(A)$, with A solution of Equation (I) for a given ρ.

3. A Branch and Bound Algorithm for Designing a Space Thruster

We solve the problem $(\wp_\rho)$ with a global optimization technique based on Branch and Bound method. But it is well known that the complexity of that method is 2^N (N is the number of variables see problem $(\wp_\rho)$). Hence, it is very difficult to deal with large scale problems. In our study, we have to use a hypothesis by observing some monotonicity of the magnetic flux distribution in the design domain.

Hypothesis 1. *Let X be a subset of $[0,1]^N$. For all ρ, we have:*

$$\rho \in X \Longrightarrow \|\mathcal{B}(\rho)\| \in [\|\mathcal{B}(X.inf)\|, \|\mathcal{B}(X.sup)\|], \tag{2}$$

The standard vectorial interval arithmetic notations are used for $.inf$ and $.sup$.

Remark 1. *This hypothesis owns a physical sense. Indeed it means that in the design domain the density of the module of the magnetic flux $\mathcal{B}$ increase with the presence of the ferromagnetic material in the variable areas. Moreover, a lot of numerical tests were performed and they confirmed that hypothesis 1 holds (at least for all the configurations that we studied so far). Nevertheless, actually we cannot provide an entire proof of hypothesis 1.*

Using hypothesis 1, it is possible to construct efficient lower bounds, as follows:

Proposition 2. *Let X be a subset of $[0,1]^N$. If hypothesis 1 holds, we have:*

$$F(\rho) \geq \|\mathcal{B}(X.inf)\|^2 - 2\|\mathcal{B}(X.sup)\|\|B_0\| + \|B_0\|^2, \forall \rho \in X. \tag{3}$$

Proof. With hypothesis 1 the proof is direct by expanding the expression of $F(\rho)$. □

By using again hypothesis 1, we obtain the following two following properties:

Proposition 3. $\|\mathcal{B}(X.inf)\| \geq \|B_0\| + \sqrt{\tilde{f}}$ *and* $\|\mathcal{B}(X.sup)\| \leq \|B_0\| - \sqrt{\tilde{f}}$*, where $\tilde{f}$ is a current solution obtained during the iterations of the Branch and Bound algorithm.*

Proof. By considering the objective function and $\tilde{f}$ and by denoting ρ^* a global minimizer, we have that $f(\rho^*) \leq \tilde{f}$. This yields that we are only interested by points $\rho \in X$ such that $f(\rho) = \|\mathcal{B}(\rho) - B_0\|^2 \leq \tilde{f}$. By remarking that $\|\mathcal{B}(\rho) - B_0\|^2 \geq (\|\mathcal{B}(\rho)\| - \|B_0\|)^2$, the result follows. □

Remark 4. *Proposition 3 yields two constraints that can be used inside our Branch and Bound algorithm. These particular added constraints make much more efficient the Branch and Bound code that we developed here.*

4. Numerical Results

Our approach was tested with success on some examples. We present here two results: the first one owns 6 variables where the global solution is obtained in 15 iterations ($< 2^6 = 64$) and the second one with 20 variables is much more difficult and the global optimum is proveded in 4750 iterations (instead of 1 048 576 iterations), see Figures 2 and 3. These numerical results were performed with: $J_1 = -2 \cdot 10^6$ A/m^2 (in coil1), $J_2 = +2 \cdot 10^6$ A/m^2 (in coil2), $\mu_{\text{min}} = \mu_0 = 4\pi \cdot 10^{-7}$ H/m and $\mu_{\text{max}} = 1000\mu_0$. In Figures 2 and 3, we just plot the design variable areas (Ω_{V1} on the left and Ω_{V2} on the right) where the blue cells are for void regions and the red ones are for iron parts.

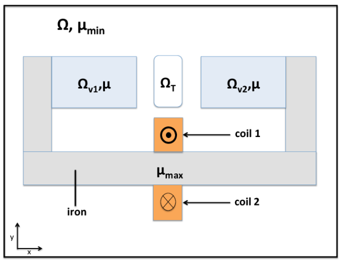


Figure 1: Design domain subdivision for topological optimization.

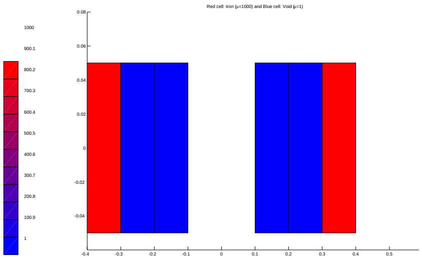


Figure 2: Global optimal design for the problem with 6 variables (in 15 iterations and about 2.5 seconds).

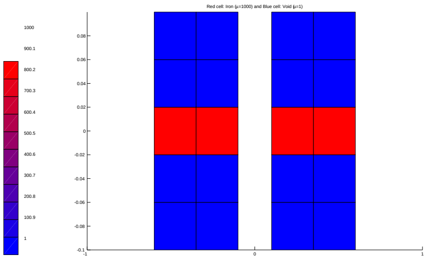


Figure 3: Global optimal design for the problem with 20 variables (in 4676 iterations and about 17 hours).

5. Conclusion

In this paper, we present a way to solve a difficult optimal design problem where the objective function has to be computed via a finite element code. Remarking that hypothesis 1 holds in our cases, we derive properties which makes it possible the computation of bounds of the objective function as well as the addition of constraints. Thus, a Branch and Bound code is provided and validated on two examples. This will be not possible to use this exact global optimization method to solve large scale topology optimization such as those encounter in real-life applications. However, this method has two main interests: (i) it permits to construct small difficult problems with a known solution which makes it possible to validate some other local approaches; (ii) it permits to construct starting points for a local solver which will work on a more discretized domain.

Acknowledgments

The authors wants to thank Carole Hénaux and Raphaël Vilamot for their indirect helps in this work.

References

[1] S. Sanogo, F. Messine, C. Henaux and R. Vilamot. Topology Optimization for Magnetic Circuits dedicated to Electric Propulsion: Optimization Online, http://www.optimization-online.org/index.html, 2014.

[2] M. P. Bendsøe, and O. Sigmund. Material interpolation schemes in topology optimization: Archive of Applied Mechanics 69, 1999, pp 635-654 ©Springer-Verlag.

Solving Integer Programs with Dense Conflict Graphs

Austin Buchanan[1], Jose L. Walteros[2], Sergiy Butenko[1] and Panos M. Pardalos[2]

[1] *Texas A&M University MS-3131, College Station, TX 77843-3131 USA* {albucha,butenko}@tamu.edu

[2] *University of Florida, Gainesville, FL 32611, USA* {jwalteros,pardalos}@ufl.edu

Abstract This paper describes branching strategies for n-variable m-constraint 0-1 programs with dense conflict graphs. The main idea is to branch on the variable that causes the most conflicts in the conflict graph. It leads to an algorithm whose runtime is parameterized by compatibility-degeneracy d (defined in the paper). The algorithm runs in time $O(n^2 \log n + n^2 m + nT_{d,m})$ where $T_{d,m}$ is the time to solve a subproblem of the 0-1 program that has d unfixed variables and m constraints. A simple extension lists all feasible solutions in time $O\left(n^2(\log n + m2^d)\right)$ which is worst-case optimal to within a polynomial factor. A second approach, which branches on pairs of variables, achieves a smaller exponent at the cost of a larger polynomial factor. Both approaches are easy to parallelize. The branching strategies apply to a variety of 0-1 programs—not necessarily integer linear programs. The analysis can provide one explanation for why integer programs can be extremely difficult to solve in the worst-case, but can be tractable in practice.

Keywords: Integer Programming, Conflict Graphs, Fixed-parameter Tractability

1. Introduction

Many important problems are $\mathcal{NP}$-hard and do not even admit good approximation algorithms under reasonable complexity assumptions. For example it is $\mathcal{NP}$-hard to approximate the maximum clique problem within $n^{1-\epsilon}$ for any $\epsilon > 0$ [15, 25], yet picking any single vertex gives an n-approximation. Surely this is not satisfactory. Difficulties such as $\mathcal{NP}$-completeness and inapproximability lead to a growing interest in the study of fixed-parameter tractability and exact exponential algorithms. The reader is referred to the texts of [10] and [12] for more information about these growing fields.

A parameterized problem is said to be fixed-parameter tractable if instances with size n and parameter p admit an algorithm running in time $f(p)n^{O(1)}$, where $f(\cdot)$ is a function that may grow exponentially in p (or worse) yet is independent of n [10]. For example, our previous paper [7] provides an algorithm for solving the maximum clique problem in d-degenerate n-vertex graphs in time $O^*(2^{d/4})$, thus replacing n with d in the $O(2^{n/4})$ algorithm of [22]. (The O^* notation suppresses polynomial factors.) Our work in this paper generalizes this approach for 0-1 programs using properties of an associated conflict graph to parameterize the runtime. Conflict graphs are well-studied in the integer programming literature [4] and are used in commercial solvers.

Fixed-parameter tractability provides one explanation for why problems can be extremely difficult in theory, yet tractable in practice. When problems are parameterized only by the size n of the problem, a running time of $\Theta(2^n)$ is prohibitive for all but the smallest instances.

However, if the instances encountered in practice exhibit relatively small values of the parameter p, an algorithm running time $2^p n^{O(1)}$ can be useful. In this sense, our approach can provide one explanation for why integer programming can be extremely hard in the worst-case, yet can appear much easier in practice.

We consider the canonical 0-1 programming problem:

$$\text{maximize } c^T x \tag{1}$$
$$\text{subject to } Ax \leq b \tag{2}$$
$$x \in \{0,1\}^n. \tag{3}$$

We assume that a conflict graph $G = (V, E)$ is given. Conflict graphs represent logical implications between the variables. For each binary variable x_i there is a vertex x_{i1} representing $x_i = 1$ and a vertex x_{i0} representing $x_i = 0$. We denote the vertex set by $V = \{x_{it} : i \in \{1, \ldots, n\}, t \in \{0,1\}\}$. Thus G has $2n$ vertices. An edge $\{x_{is}, x_{jt}\}$ in G represents a conflict encountered when simultaneously setting $x_i = s$ and $x_j = t$. This conflict either gives rise to an infeasibility or to a suboptimal solution. For more information about conflict graphs and on typical procedures to create them, see [4]. Note that this type of a conflict graph expresses the same information as an implication graph [3], but it is different from the conflict graphs [1].

In this paper, it is often easier to work with the complement of the conflict graph. This graph, which we will call a *compatibility* graph, will be sparse whenever the conflict graph is dense. An optimal solution to the 0-1 program corresponds to a clique in the compatibility graph. The converse is not true.

2. Degeneracy, Its Extensions, and the Algorithms

The notion of compatibility-degeneracy is inspired by degeneracy. The degeneracy of a graph is a common measure of its sparsity and is within a constant factor of other measures of sparsity such as arboricity and thickness and is bounded above by pathwidth, treewidth, and the h-index [11]. It has been used to parameterize a variety of algorithms [7, 11, 2]. Degeneracy is also known as width and linkage, and an algorithm with some similarities to our approach was defined in these terms [13, 14]. However this approach uses a *constraint* graph, which has two major differences with conflict graphs. First, a constraint graph has only one vertex per variable, and variables need not be binary. Second, an edge between two vertices in a constraint graph denotes that the two associated variables appear in the same constraint. Thus, constraint graphs convey different information. They also have fewer vertices and typically have more edges than conflict graphs.

Definition 1 (degeneracy). *A graph is said to be d-degenerate if every (non-empty) subgraph has a vertex of degree at most d. The degeneracy of a graph is the smallest value of d such that it is d-degenerate.*

By [19], every d-degenerate n-vertex graph admits an ordering of its vertices $(v_1, \ldots, v_n)$ such that each vertex v_i has at most d neighbors after it in the ordering, i.e., $|N(v_i) \cap \{v_i, \ldots, v_n\}| \leq d$. In fact, admitting such an ordering is equivalent to being d-degenerate [19]. The degeneracy, as well as such an ordering, can be found in linear time by iteratively removing a vertex of minimum degree [21].

Given a compatibility graph one may ask if there is a binary assignment to the variables that avoids all conflicts. This is an instance of 2SAT, which can be solved in linear time [3] (possibly quadratic with respect to n). We will say that a compatibility graph is feasible if such an assignment exists. This can be generalized for subgraphs of compatibility graphs, and feasibility can still be determined in linear time.

Definition 2 (feasible subgraph). *A subgraph $G' = (V', E')$ of a compatibility graph $G = (V, E)$ is said to be feasible if and only if there exists a mutually compatible subset $S \subseteq V'$ of n vertices, i.e., for every pair of distinct vertices $u, v \in S$, we have $\{u, v\} \in E'$. Otherwise, G' is said to be infeasible.*

We remark that the class of infeasible subgraphs of a compatibility graph is closed under taking induced subgraphs. This is easy to see because the n "mutually compatible" vertices are a clique. Note that a feasible 0-1 program implies a feasible subgraph, but the converse may not be true.

For simple undirected graphs the open (closed) neighborhood of a vertex is denoted by $N(\cdot)$ $(N[\cdot])$. A similar notion, specifically for compatibility graphs, is denoted by $\mathcal{N}(\cdot)$.

Definition 3 (compatibility-neighborhood)**.** *Given a feasible subgraph $G' = (V', E')$ of a compatibility graph G, the compatibility-neighborhood of a vertex $x_{it} \in V'$ in G' is denoted by $\mathcal{N}_{G'}(x_{it}) = \{j : \{x_{it}, x_{j0}\} \in E'$ and $\{x_{it}, x_{j1}\} \in E'\}$, which is the set of indices of variables that remain free when fixing $x_i = t$.*

Definition 4 (compatibility-degeneracy)**.** *A compatibility graph is said to be d-compatibility-degenerate if for every feasible subgraph $G' = (V', E')$ there exists a vertex $x_{it} \in V'$ with compatibility-degree $|\mathcal{N}_{G'}(x_{it})| \leq d$. The compatibility-degeneracy is the smallest value of d such that the graph is d-compatibility-degenerate.*

Just as a d-degenerate graph admits a degeneracy-ordering, a d-compatibility-degenerate graph admits a compatibility-degeneracy ordering. We will see that a compatibility-degeneracy ordering can also be computed in polynomial time.

Proposition 5. *A compatibility graph G is d-compatibility-degenerate if and only if it admits an ordering $(x_{i_1t_1}, \ldots, x_{i_{2n}t_{2n}})$ of its vertices such that for every vertex $x_{i_jt_j}$ either $G[S_j]$ is infeasible or $|\mathcal{N}_{G[S_j]}(x_{i_jt_j})| \leq d$, where $S_j = \{x_{i_jt_j}, \ldots, x_{i_{2n}t_{2n}}\}$.*

Thus, if we have fixed $x_{i_j} = t_j$, there are at most d variables whose vertices occur later in the ordering, i.e., at most d variables remain undetermined. This is the main idea that will be exploited in the algorithm that is based on compatibility-degeneracy.

Now we describe a different parameter related to the compatibility graph that is based on pairs of adjacent vertices. This results in the notion of bicompatibility-degeneracy d', which is a generalization of community-degeneracy introduced in our previous paper [7]. We note that the bicompatibility-degeneracy can be found in polynomial time, and provide such an algorithm.

Definition 6 (bicompatibility-degeneracy)**.** *A compatibility graph is said to be d-bicompatibility-degenerate if for every feasible subgraph $G' = (V', E')$ there exists an edge $\{x_{is}, x_{jt}\} \in E'$ with $|\mathcal{N}_{G'}(x_{is}) \cap \mathcal{N}_{G'}(x_{jt})| \leq d$. The bicompatibility-degeneracy d' is the smallest value of d such that the graph is d-bicompatibility-degenerate.*

Given a graph $G = (V, E)$, the *edge-induced* subgraph of E', denoted $G[E'] = (V', E')$, includes those vertices V' that are an endpoint of an edge in E'.

Proposition 7. *A compatibility graph G is d-bicompatibility-degenerate if and only if it admits an ordering $(e_1, \ldots, e_m)$ of its edges such that for every edge $e_k = \{u, v\}$ either $G[E_k]$ is infeasible or $|\mathcal{N}_{G[E_k]}(u) \cap \mathcal{N}_{G[E_k]}(v)| \leq d$, where $E_k = \{e_k, \ldots, e_m\}$.*

We propose polynomial-time algorithms for computing compatibility-degeneracy d and bicompatibility-degeneracy d'. Two algorithms for solving 0-1 programs are developed whose runtimes are parameterized by d or d'. The first algorithm solves 0-1 programs with d-compatibility-degenerate graphs in time $O(n^2 \log n + n^2 m + nT_{d,m}) = O^*(2^d)$. Using $q = O(n)$ processors, this reduces to time $O(n^2 \log n + nm + T_{d,m})$. The following result holds.

Theorem 8. *All feasible solutions of d-compatibility-degenerate 0-1 programs can be listed in time $O(n^2(\log n + m2^d)) = O^*(2^d)$. Any such algorithm requires $\Omega(n(n-d)2^d)$ time, so the approach is optimal to within a polynomial factor.*

The second algorithm solves 0-1 programs with d'-bicompatibility-degenerate graphs $G = (V, E)$ in time $O\left(n^2 \log n + |E|(n|E| + nm + T_{d',m})\right) = O^*(2^{d'})$. Using $q = O(|E|)$ processors, this reduces to time $O\left(n|E|^2 + n^2 \log n + nm + T_{d',m})\right)$. The following result holds.

Theorem 9. *All feasible solutions of d-compatibility-degenerate 0-1 programs can be listed in time $O^*(2^{d'})$. Any such algorithm requires $\Omega(n(n-d')^2 2^{d'})$ time, so the approach is optimal to within a polynomial factor.*

In the talk, we will also report some sample values of d for instances in literature.

References

[1] T. Achterberg. Conflict analysis in mixed integer programming. *Discrete Optimization*, 4(1):4–20, 2007.

[2] N. Alon and S. Gutner. Linear time algorithms for finding a dominating set of fixed size in degenerated graphs. *Algorithmica*, 54(4):544–556, 2009.

[3] B. Aspvall, M.F. Plass, and R.E. Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters*, 8(3):121–123, 1979.

[4] A. Atamtürk, G.L. Nemhauser, and M.W.P. Savelsbergh. Conflict graphs in solving integer programming problems. *European Journal of Operational Research*, 121(1):40–55, 2000.

[5] D.A. Bader, H. Meyerhenke, P. Sanders, and D. Wagner, editors. *Graph partitioning and graph clustering. 10th DIMACS Implementation Challenge Workshop*, volume 588 of *Contemporary Mathematics*. American Mathematical Society and Center for Discrete Mathematics and Theoretical Computer Science, 2013.

[6] R. Bixby, S. Ceria, C. McZeal, and M. Savelsbergh. An updated mixed integer programming library: MIPLIB 3.0, 1996.

[7] A. Buchanan, J.L. Walteros, S. Butenko, and P. M. Pardalos. Solving maximum clique in sparse graphs: an $O(nm + n2^{d/4})$ algorithm for d-degenerate graphs. *Optimization Letters*, 2013. To appear.

[8] C. Calabro, R. Impagliazzo, and R. Paturi. The complexity of satisfiability of small depth circuits. In *Parameterized and Exact Computation*, pages 75–85. Springer, 2009.

[9] N. Creignou and J.-J. Hébrard. On generating all solutions of generalized satisfiability problems. *Informatique théorique et applications*, 31(6):499–511, 1997.

[10] R.G. Downey and M.R. Fellows. *Parameterized Complexity*. Springer, 1999.

[11] D. Eppstein, M. Löffler, and D. Strash. Listing all maximal cliques in sparse graphs in near-optimal time. *Algorithms and Computation*, pages 403–414, 2010.

[12] F.V. Fomin and D. Kratsch. *Exact exponential algorithms*. Springer, 2010.

[13] E.C. Freuder. A sufficient condition for backtrack-free search. *Journal of the ACM*, 29(1):24–32, 1982.

[14] E.C. Freuder. A sufficient condition for backtrack-bounded search. *Journal of the ACM*, 32(4):755–761, 1985.

[15] J. Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, 182(1):105–142, 1999.

[16] R. Impagliazzo and R. Paturi. Complexity of k-sat. In *Proceedings of the Fourteenth Annual IEEE Conference on Computational Complexity*, pages 237–240. IEEE, 1999.

[17] R. Impagliazzo, R. Paturi, and S. Schneider. A satisfiability algorithm for sparse depth two threshold circuits, 2013. http://arxiv.org/abs/1212.4548.

[18] D.S. Johnson and M.A. Trick, editors. *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, Workshop, October 11-13, 1993*, volume 26 of *Discrete Mathematics and Theoretical Computer Science*. AMS, 1996.

[19] D.R. Lick and A.T. White. k-degenerate graphs. *Canad. J. Math*, 22:1082–1096, 1970.

[20] A.O. Makhorin. GLPK (GNU linear programming kit) version 4.52, 2013. http://www.gnu.org/software/glpk/.

[21] D.W. Matula and L.L. Beck. Smallest-last ordering and clustering and graph coloring algorithms. *Journal of the ACM*, 30(3):417–427, 1983.

[22] J.M. Robson. Finding a maximum independent set in time $O(2^{n/4})$. Technical report, LaBRI, Université de Bordeaux I, 2001. http://www.labri.fr/perso/robson/mis/techrep.html.

[23] M. Rospocher. *On the computational complexity of enumerating certificates of NP problems*. PhD thesis, University of Trento, 2006.

[24] D.R. Wood. On the maximum number of cliques in a graph. *Graphs and Combinatorics*, 23(3):337–352, 2007.

[25] D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 681–690. ACM, 2006.

Maximizing the number of solved aircraft conflicts through velocity regulation [*]

Sonia Cafieri

ENAC, MAIAA, F-31055 Toulouse, France,
Université de Toulouse, IMT, F-31400 Toulouse, France,
sonia.cafieri@enac.fr

Abstract We propose a model for the maximization of the number of aircraft conflicts that can be solved by performing velocity regulation. The model is mixed-integer as binary variables are used to count solved conflicts and to model alternative choices, while nonlinearities appear in the aircraft separation conditions. The main nonlinearities can however be relaxed by standard reformulations. Numerical results show that the model can be satisfactorly applied at least as a preprocessing step in a conflict avoidance procedure in a given airspace.

Keywords: Aircraft conflict avoidance, mixed-integer nonlinear optimization (MINLO), mathematical modelling, Air Traffic Management applications

1. Introduction

The growth of air traffic on the world scale leads to an increasing need for automatic decision-support tools able to integrate the work of air traffic controllers to guarantee flight safety. In this context, we focus on a crucial problem arising in Air Traffic Management, that of aircraft conflict detection and resolution.

Aircraft are said to be potentially *in conflict* when their horizontal or altitude distances are less than given standard separation distances (5NM, where 1 NM (nautical mile)= 1852 m, and 1000 ft). Thus, when a loss of separation occurs, aircraft have to be separated. Aircraft conflict avoidance can be performed by different strategies aimed to separate aircraft, including aircraft trajectory (heading angle) changes, flight level changes or aircraft velocity regulation. Corresponding mathematical models can be developed, leading to optimization or optimal control problems. A review is provided in [5]. In recent years, mixed-integer linear and nonlinear optimization (MILO, MINLO) have been proposed for aircraft conflict detection and resolution, with interesting results. See for example [1], [4], [6], [7]. In previous work [4], we proposed MINLO models for conflict avoidance based on velocity regulation, aiming at solving all the conflicts occurring in a given air sector observed during a time horizon (in a tactical flight phase). These models are quite complex and computationally challenging.

In this paper, we propose a mixed-integer nonlinear programming problem for maximizing the number of conflicts that are solved, in a time horizon, when only a velocity regulation is performed. The model can be easily relaxed using standard reformulations. The interest of the proposed model is twofold. First, it allows to easily discriminate between conflicts that can be solved by velocity changes and those that require the application of another separation strategy. Second, it can provide a starting point, and a corresponding feasible solution, for more complex models like those in [4], eventually simplifying the branch-and-bound procedure for their solution.

[*] Financial support under grant ANR 12-JS02-009-01 “ATOMIC“ is gratefully acknowledged.

The paper is organized as follows. In Sect. 2 we present the proposed mixed-integer optimization model. In Sect. 3 we discuss the results of some numerical experiments. Sect. 4 concludes the paper.

2. Model: maximizing the number of solved conflicts

We model conflict avoidance in such a way to achieve aircraft separation by performing a speed change maneuver. This means that a conflict involving a pair of aircraft is solved by aircraft acceleration or deceleration, so that aircraft pass through the points of potential conflict at a different time with respect to what would occur if no maneuvers were performed. There are however a few situations where velocity regulation cannot solve all conflicts of a given aircraft configuration, that corresponds to infeasible optimization problems. In such a case, speed change maneuvers can be performed, leaving potentially some conflicts unsolved and needing the application of another separation maneuver, like heading angle changes.

In the present work we propose an optimization problem where the number of aircraft conflicts that can be solved by speed changes is maximized. The proposed model can then be used as a preprocessing step in a conflict resolution procedure in a target airspace.

Let A be the set of n aircraft. For all $i, j \in A$, let z_{ij} be binary decision variables defined as

$$z_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are separated (no conflict)} \\ 0 & \text{otherwise} \end{cases}$$

The other decision (continuous) variables of the problem are represented by the aircraft velocities, which are eventually modified with respect to the original ones (that are data of the problem) to solve conflicts:

$$v_{min} \leq \bar{v}_i \leq v_{max} \quad \forall i \in A,$$

where the bounds v_{min} and v_{max} are imposed to allow aircraft only small speed changes, following the idea of *subliminal control* of velocities suggested in the context of the aeronautic project ERASMUS [3], such that speeds can vary between -6% and +3% of the original speed. We obtain a mixed-integer model because of the presence of binary as well as continuous variables.

The objective function, to be maximized, is the sum of solved conflicts:

$$\sum_{i,j \in A,\ i<j} z_{ij}.$$

The constraints are given by the integrality constraints on z variables, the bounds on $\bar{v}$ variables, and the separation constraint on pairs of aircraft.

Let us assume that aircraft fly at the same flight level and are identified by 2-dimensional points on a plane. We know their initial position, their trajectory (heading) and their velocity. The aircraft separation between two aircraft i and j at time t is expressed by the condition

$$||x^r_{ij}(t)|| \geq d, \tag{1}$$

where d is the minimum required separation distance (usually, 5 NM) and $x^r_{ij}(t) = x_i(t) - x_j(t)$ is a vector representing the relative distance between aircraft i and j.

We assume that uniform motion laws apply, so the relative distance of aircraft i and j is expressed as the sum of their relative initial position and the product of their relative speed $\bar{v}^r_{ij}$ by the time:

$$x^r(t) = x^{r0}_{ij} + \bar{v}^r_{ij} t \qquad \forall t,$$

that, substituting into (1) and squaring, gives

$$(v^r_{ij})^2\, t^2 + 2(x^{r0}_{ij} \bar{v}^r_{ij})\, t + ((x^{r0}_{ij})^2 - d^2) \geq 0. \tag{2}$$

Notice that the associated equation is an equation of second degree in one unknown t (its graph is a parabola that, as $(\bar{v}^r_{ij})^2 > 0$, has a minimum point and opens upward), that has no solutions if the discriminant $\Delta = (x^{r0}_{ij}\bar{v}^r_{ij})^2 - (\bar{v}^r_{ij})^2((x^{r0}_{ij})^2 - d^2)$ is negative. The solutions of this equation, if they exist, are the times at which the aircraft are not separated. So, we consider $\Delta < 0$ as a first condition of separation of aircraft i and j. In the case when this condition is not satisfied, and so aircraft i and j can potentially be in conflict, we look at the form of trajectories. In this work, we assume that trajectories are straight lines intersecting in one point. As per the geometric interpretation of the scalar product, we can look at the sign of the scalar product $x^{r0}_{ij}\bar{v}^r_{ij}$ to infer if the vectors form an acute or an obtuse angle. In particular, when the scalar product $x^{r0}_{ij}\bar{v}^r_{ij}$ is negative, then the aircraft are converging, potentially generating a conflict, while they are separated when the product is positive.
Finally, we impose aircraft separation imposing that $\Delta < 0$ or $x^{r0}_{ij}\bar{v}^r_{ij} > 0$ for all $i, j,\ i < j$. Using again the z binary variables, the two constraints are written as

$$\left((x^{r0}_{ij}\bar{v}^r_{ij})^2 - (\bar{v}^r_{ij})^2((x^{r0}_{ij})^2 - d^2)\right)(2z_{ij} - 1) \leq 0$$

i.e.

$$(x^{r0}_{ij}\bar{v}^r_{ij})^2(2z_{ij} - 1) \leq (\bar{v}^r_{ij})^2((x^{r0}_{ij})^2 - d^2)(2z_{ij} - 1) \tag{3}$$

and, respectively,

$$(x^{r0}_{ij}\bar{v}^r_{ij})(2z_{ij} - 1) \geq 0. \tag{4}$$

Notice that the left hand sides of the two conditions differ only for a square. The same binary variable z_{ij} can be used to model the *or* condition:

$$\begin{aligned} (x^{r0}_{ij}\bar{v}^r_{ij})^2(2z_{ij} - 1) &\leq (\bar{v}^r_{ij})^2((x^{r0}_{ij})^2 - d^2)(2z_{ij} - 1) && (5)\\ (x^{r0}_{ij}\bar{v}^r_{ij})(1 - z_{ij}) &\geq 0 && (6)\end{aligned}$$

then using an additional variable to account for a separated pair of aircraft when the second condition is satisfied.

The nonlinear terms appearing in the constraints come mainly from the products between continuous and binary variables, that can be easily relaxed using the Fortet linearization. This is commonly implemented in the most of the MINLO solvers.

3. Numerical experiments

We tested our model on instances built placing n aircraft on a circle of a given radius r, in 2-dimensional space, with speed v and a heading angle such that their trajectory is toward the center of the circle (or slightly deviated with respect to such direction). The zone of conflict is around the center of the circle where aircraft are placed, and each aircraft is in conflict with each other. We solve the problem using `COUENNE` [2], which implements a spatial Branch-and-Bound based on convex relaxations and provides the global optimal solution.

As an example of solution, let us consider an instance of the conflict avoidance problem with $n = 5$ aircraft having speed v = 400 NM/h (equal for all aircraft). There are 10 potential conflicts, that are all solved.
The ratio of the new speeds over the original ones for the 5 aircraft is shown in Table 1.
We see that 2 aircraft are accelerated and 3 of them are decelerated. The speed variation are in the small range [-6%, +3%] around the original velocity for a subliminal control as suggested by ERASMUS.
The global optimal solution is obtained in 0.16 seconds.

Table 1: Ratio of the aircraft velocities in the optimal solution over the original ones.

aircraft	v_{ratio}
1	1.00814
2	1.02809
3	0.941877
4	0.981939
5	0.962551

4. Summary

We proposed a mathematical model for the maximization of the number of aircraft conflicts that can be solved by velocity changes. The model gives a mixed-integer nonlinear optimization problem that can be efficiently solved by standard solvers for MINLO.

References

[1] A. Alonso-Ayuso, L.F. Escudero, and F.J. Martín-Campo. A mixed 0-1 nonlinear optimization model and algorithmic approach for the collision avoidance in ATM: Velocity changes through a time horizon. *Computers and Operations Research*, 39(12):3136–3146, 2012

[2] 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.

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

[4] S. Cafieri, N. Durand. Aircraft deconfliction with speed regulation: new models from mixed-integer optimization. *Journal of Global Optimization*, 58:613–629, 2014.

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

[6] 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.

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

Falsification of Hybrid Dynamical Systems Using Global Optimization Techniques*

Jan Kuřátko[1] and Stefan Ratschan[2]

[1] *Institute of Computer Science, Czech Academy of Sciences,* kuratko@cs.cas.cz

[2] *Institute of Computer Science, Czech Academy of Sciences, ORCID: 0000-0003-1710-1513,* ratschan@cs.cas.cz

Abstract A hybrid dynamical system is a dynamical system that shows both continuous and discrete state and behavior. In our talk we will discuss the usage of global optimization techniques for solving the problem of finding a trajectory that leaves a given set of states considered to be safe. Unlike other known methods, we do not restrict our search to trajectories of a certain bounded length. The algorithm combines local with global search for achieving both efficiency and global convergence, and exploits derivatives for efficient computation.

Keywords:

hybrid dynamical systems, global optimization

1. Introduction

A hybrid dynamical system combines the continuous state and behavior of ordinary differential equations with some discrete state and behavior. The importance of such systems stems from their suitability for modeling embedded systems, where a discrete controller interacts with its continuous environment. In our talk we will discuss the application of global optimization techniques to the problem of falsification of hybrid systems, that is, to the following problem:

Given:

- A hybrid system H,
- a set of states I (considered to be initial),
- and a set of states U (considered to be unsafe).

Find: A trajectory of H that starts in the set of initial states I and reaches the set of unsafe states U (we will call such a trajectory an *error trajectory*).

Existing methods for falsification of hybrid systems roughly fall into the following two categories:

- Local search [1, 9]: Such methods use local optimization to incrementally bring a starting trajectory closer to an error trajectory. The advantage of local search is its relative efficiency. The disadvantage is that for convergence it needs to be started close enough to an error trajectory. Here the partially discrete nature of hybrid system brings an additional difficulty, since there is no natural concept of ”discrete closeness” which makes the search for starting trajectories for local search a difficult problem.

*This work was supported by the Czech Science Foundation (GAČR) grant number P202/12/J060 with institutional support RVO:67985807.

- Black-box global search [2, 6]: Such methods search for error trajectories globally, but use black box optimization techniques [4, 7] that do not exploit the structure specific to hybrid systems. This may be desired in some applications (e.g., for being able to handle Simulink models), but this may also result in a loss of efficiency. Moreover, it only searches for error-trajectories up to a user-specified upper bound on their length.

In our contribution we will present an algorithm that combines the advantage of both approaches:

- It aims at quickly finding starting points for local search that can then directly converge to an error trajectory.
- Unlike methods based on black-box global search, the algorithm will be able to handle the fact that trajectories of dynamical systems can have arbitrary length. It will not assume a user-provided upper bound on the length of error trajectories, but will search for error trajectories of arbitrary length.
- The algorithm will fulfill some global convergence properties.

However, the current goal is not to come up with an algorithm that is—on its own—as efficient as possible. Instead, we want to present an algorithm that fulfills the properties above and, in addition, gives an as-simple-as-possible starting point for studying properties of the algorithm, for incorporating more sophisticated global optimization techniques [5], and for experimenting with implementations of specific, efficient variants of the algorithm.

The starting point for our approach are standard techniques from global optimization for combining local with global search, so called two-phase methods [8]. But we adapt those methods to the situation that we have here: A direct application of two-phase methods would use a search space that is spanned by variables of two kinds:

- the initial point of trajectories, and
- the trajectory length (wrt. time).

However, trajectory length is special, since it

- is unbounded, and
- computing a trajectory of the given length from a given initial point also computes all trajectories from that initial point with shorter length.

Moreover, hybrid systems combine continuous with discrete behavior and local search can exploit derivatives for searching the continuous part of the states space, but no such derivatives are available for discrete search which is another obstacle to the direct application of two-phase methods.

Hence, our approach modifies two-phase methods in such a way that —instead of treating trajectory length as a problem variable— they build trajectories incrementally from trajectory segments, and use derivative based continuous local search to glue together those segments based on continuous search (cf. the notion of ”multiple shooting” [3, 9] in the literature on numerical algorithms for solving boundary value problems).

2. Problem Formulation

We will now formalize the problem under consideration:

Definition 1. *A* hybrid system *is a quintuple $H = (Q, \Omega, F, G, R)$, where*

- *Q is a finite set (often called the* modes *of H)*
- *$\Omega \subseteq Q \times \mathbb{R}^n$ (often called the* state space *of H)*
- *F is a function that assigns to each mode $q \in Q$ a system of ordinary differential equations $F_q(x, \dot{x}) = 0$, where $(q, x) \in \Omega$*
- *$G \subseteq \Omega$ (often called the* guards *of H)*
- *$R : \Omega \mapsto \Omega$ (often called the* reset function *of H)*

Definition 2. *A* trajectory *of a hybrid dynamical system H is a sequence of the form*

$$((q_1, x_1), (q_2, x_2), \ldots (q_k, x_k)),$$

where for all $i = 1, \ldots, k$, $q_i \in Q$ and $x_i : [0, t_i] \mapsto \mathbb{R}^n$ is a continuous trajectory of the system of differential equations given by $F(q_i)$, such that for all $t \in [0, t_i]$, $(q_i, x_i(t)) \in \Omega$. We call $t_i \in \mathbb{R}^{\geq 0}$ the length *of x_i.*

We denote by $(q_i, x_i^s) \in \Omega$ the starting point of a trajectory (q_i, x_i) and $(q_i, x_i^e) \in \Omega$ its endpoint. Whenever $(q_i, x_i^e) \in G(q_i)$, a discrete jump between two consecutive segments occurs and $(q_{i+1}, x_{i+1}^s) = R((q_i, x_i^e))$, $i = 1, \ldots, k-1$.

Note that —up to now— we did not fix a certain set of initial states. Instead, we allow trajectories to start in arbitrary states of the state space. Note also, that this definition results in trajectories that —from a given initial state— have a unique evolution (this is usually called a *deterministic* hybrid system).

Now we are ready to formulate the problem of *falsification of hybrid dynamical systems*.

Problem 1. *Let H be a hybrid dynamical system and* $\mathrm{Init} \subseteq \Omega$, $\mathrm{Unsafe} \subseteq \Omega$ *be two sets. The set* Init *is called the set of initial states and the set* Unsafe *is called the set of unsafe states. The problem of falsification of H is to find any trajectory $((q_1, x_1), (q_2, x_2), \ldots (q_k, x_k))$ of H such that $(q_1, x_1^s) \in \mathrm{Init}$ and $(q_k, x_k^e) \in \mathrm{Unsafe}$. Such a trajectory is called an error trajectory of H.*

3. Key Ideas of Algorithm

The algorithm that we will present, is based on the following key ideas:

- Instead of immediately computing candidates for error trajectories in one piece, we compute trajectories that are only intended to be pieces of the final error trajectory (we will call them *trajectory segments* in the following). By forming sequences of such trajectory segments we will get candidates for error trajectories.
- We reformulate the hybrid systems falsification problem into an optimization problem using an objective function that measures the distance of a sequence of trajectory segments to being an error trajectory.
- We allow various strategies for computing the trajectory segments (e.g., random global search). In order to handle the lack of a natural distance measure that can be used for local optimization over the discrete modes we connect the discrete modes by trajectory segments.

- We use continuous local search to minimize the distance of a sequence of trajectory segments to being an error trajectory. This glues together the trajectory segments and moves the first and last point toward the set of initial and unsafe states. Note that here we do not only move the trajectory segments, but we also optimize their length.
- We initialize the set of computed trajectory segments in such a way that we can use continuous local search from the beginning.

The algorithm can be instantiated with a variety of heuristics and strategies. In the talk we will present some variants of the algorithm and the results of computational experiments with them. We will also discuss conditions under which we succeeded to prove global convergence of the algorithm.

4. Conclusion

We use techniques from global optimization to solve the problem of falsification of hybrid dynamical systems. In general, the application of global optimization techniques to hybrid systems promises to be highly fruitful.

References

[1] H. Abbas and G. Fainekos. Linear hybrid system falsification through local search. In *Automated Technology for Verification and Analysis*, volume 6996, pages 503–510. Springer, 2011.

[2] Y. Annpureddy, C. Liu, G. Fainekos, and S. Sankaranarayanan. S-TaLiRo: A tool for temporal logic falsification for hybrid systems. In Parosh Aziz Abdulla and K.Rustan M. Leino, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, volume 6605 of *Lecture Notes in Computer Science*, pages 254–257. Springer Berlin Heidelberg, 2011.

[3] U.M. Ascher and L.R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1st edition, 1998.

[4] M. Gendreau and J.Y. Potvin, editors. *Handbook of Metaheuristics.* Springer, 2nd edition, 2010.

[5] M. Locatelli and F. Schoen. *Global Optimization–Theory, Algorithms, and Applications.* SIAM, 2013.

[6] T. Nghiem, S. Sankaranarayanan, G. Fainekos, F. Ivančić, Aarti Gupta, and George J. Pappas. Monte-Carlo techniques for falsification of temporal properties of non-linear hybrid systems. In *HSCC'10*, pages 211–220, New York, NY, USA, 2010. ACM.

[7] L.M. Rios and N.V. Sahinidis. Derivative-free optimization: A review of algorithms and comparison of software implementations. *Journal of Global Optimization*, pages 1–47, 2012.

[8] F. Schoen. Two-phase methods for global optimization. In P.M. Pardalos and H.E. Romeijn, editors, *Handbook of Global Optimization*, volume 62 of *Nonconvex Optimization and Its Applications*, pages 151–177. Springer US, 2002.

[9] A. Zutshi, S. Sankaranarayanan, J.V. Deshmukh, and J. Kapinski. A trajectory splicing approach to concretizing counterexamples for hybrid systems. In *CDC'13*, 2013.

Proceedings of MAGO 2014, pp. 137 – 140.

MINLPLib 2

Stefan Vigerske

GAMS Development Corp., svigerske@gams.com

Abstract Since 2001, the Mixed-Integer Nonlinear Programming Library (MINLPLib)[1] [1] and the GLOBAL Library (GLOBALLib)[2] have provided algorithm developers with a varied set of both theoretical and practical (MI)NLP test models. In this presentation, we report on recent progress on extending, updating, and categorizing MINLPLib and GLOBALLib. We hope that the updated library can be a starting point to define a widely accepted test set to evaluate the performance of NLP and MINLP solving software.

Keywords: mixed-integer nonlinear programming, nonlinear programming, global optimization, instance library, benchmarking

1. Introduction

Collection of instantiations of mathematical programming models play an important rule for solver software developers. The task of such collections is to provide access to a wide set of interesting problem instances with different characteristics. Especially commercial solver vendors test their solver on thousands of test problems before releasing a new software version. Additionally, the evaluation of algorithmic improvements (in terms of robustness and efficiency) requires well-balanced test sets of significantly many real-world instances.

The Netlib collection[3] [3] had an important impact on the field of Linear Programming. Still today, developers of LP solvers test and compare their implementations on this collection. By allowing for independent comparisons of Linear Programming solvers, this collection contributed to make Linear Programming solvers as reliable and efficient at they are today. Later, the MIPLIB collection[4] with its regular updates [5] has become the standard test set to compare the performance of Mixed-Integer Linear Programming solvers.

In the area of Global Optimization for (Mixed-Integer) Nonlinear Programming, several collections of model instances have been made available in various formats. In 2001, the MINLPLib collection was released [1], which integrated instances from the GAMS Model Library[5], MacMINLP[6], the MINOPT library[7], and [2][8] in a single collection using one common format (GAMS). Additionally, the CONVERT tool to translate GAMS models into other formats, including AMPL, BARON, and LINGO, was created. Initially, MINLPLib consisted of 136 instances, most of them originating from different applications. The size of the instances varied from tiny (e.g., 1 equation and 5 variables) to huge (e.g., 24972 equations and 23827 variables of which 10920 are binary). Over the years, additional instances were contributed from various sources, so that the MINLPLib consisted of ≈ 270 instances by the beginning of 2013. Similarly, the GLOBALLib collection of nonlinear programming (NLP) instances was

[1] http://www.gamsworld.org/minlp/minlplib.htm
[2] http://www.gamsworld.org/global/globallib.htm
[3] http://www.netlib.org/lp/index.html
[4] http://miplib.zib.de/
[5] http://www.gams.com/modlib/modlib.htm
[6] http://www-unix.mcs.anl.gov/ leyffer/MacMINLP/
[7] http://titan.princeton.edu/MINOPT/modlib/Tests/
[8] http://titan.princeton.edu/TestProblems/

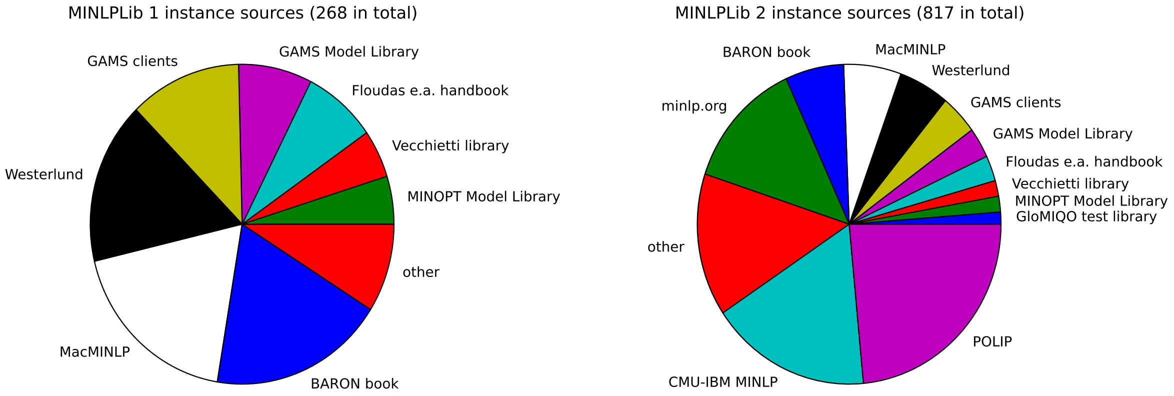


Figure 1: Source of Instances in MINLPLib at the beginning of 2013 (left) and now (right).

released in 2001. It originally consisted of 256 instances from [2] and the GAMS Model library. Today, it comprises 392 instances.

Even though MINLPLib and GLOBALLib have been extremely useful as test sets for solver developers, the inclusion of many very easy instances, many very hard instances, and large homogeneous test sets makes these collections unsuited as a benchmark set to compare global or local solvers. On the other hand, the success of Netlib and the MIPLIB collections raises the hope that a commonly accepted benchmark set of NLP and MINLP instances could enormously contribute to the development of efficient and reliable global NLP and MINLP solvers.

As a first step towards this direction, we have started in 2013 with a renovation of the MINLPLib infrastructure and instance collection. In the following, we highlight some of these developments.

2. New Instances

New MINLP Instances have been harvested from several sources. The largest portion are instances taken from the CMU-IBM Cyber-Infrastructure for MINLP[9] [4], the CMU-IBM Open source MINLP Project[10], and the library for polynomially constrained mixed-integer programming POLIP[11]. Figure 1 shows the number of instances from various sources at the beginning of 2013 and now. Figure 2a shows the size of all MINLPLib instances in a scatter plot.

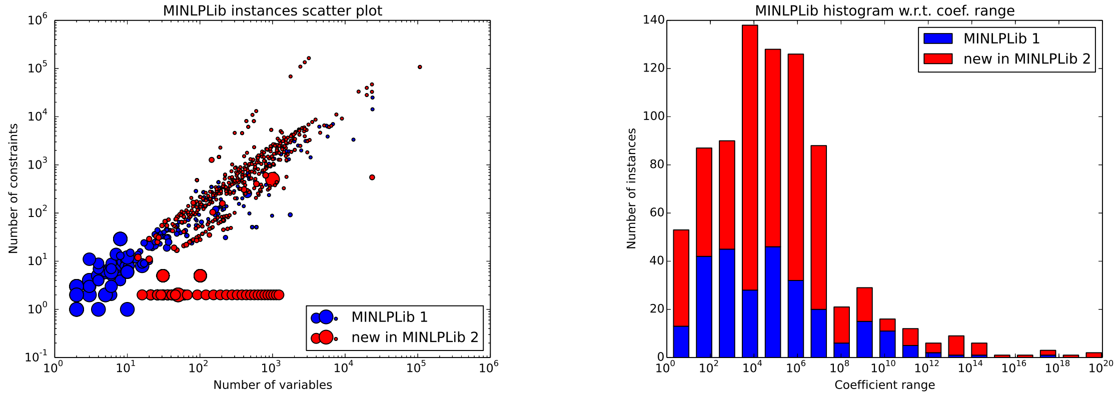
As there exists no generally accepted free format for nonlinear programs, most instances are available in various formats now. While all instances are available in GAMS format, most of them are also available in AMPL and Optimization Services Instance Language (OSiL) format[12]. Additionally, instances are available in PIP format, if nonlinearities are at most polynomial, and in CPLEX LP format, if nonlinearities are at most quadratic (≈ half of the instances). The formats OSiL, LP, and PIP have the advantage that they can be processed without a commercial modeling system like AMPL or GAMS.

[9] http://www.minlp.org

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

[11] http://polip.zib.de/

[12] In a few cases, the presence of certain nonlinear functions like errorf or signpower in the GAMS formulation prohibits a conversion into AMPL or OSiL.



(a) Scatter plot of instance sizes. Area of bubble corresponds to instance density, which is calculated as max(0.05, (#nonzeros in objective and jacobian) / (#vars (#cons + 1))).

(b) Histogram of coefficient range for all instances. The coefficient range of an instance is computed as the quotient of maximal and minimal absolute non-zero coefficient in the objective gradient and Jacobian w.r.t. the instance' initial point and known feasible solutions.

Figure 2: Instance Properties.

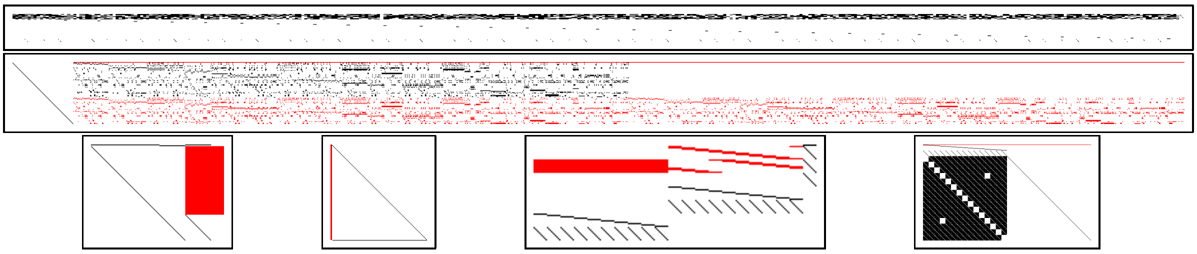


Figure 3: Sparsity pattern of Objective Gradient (first row) and Jacobian for instances `densitymod` (Density modification based on single-crystal X-ray diffraction data) (top), `lop97ic` (Rail Line Optimization) (middle), `dosemin2d` (Radiation Therapy) (bottom left), `johnall` (Asset Management) (bottom, middle left), `mbtd` (bottom, middle right), and `qapw` (Quadratic Assignment) (bottom right). Linear terms are black, nonlinear are red.

3. Instance Properties

While for the original MINLPLib, only statistics on number of variables, equations, and nonzeros were available, we have now started to include many more information on the instances itself. These include sparsity pattern (see Figure 3), coefficient ranges (as an indicator for numerical difficulty) (see Figure 2b), and distinction into linear, quadratic, polynomial, signomial, and generally nonlinear functions.

For the original MINLPLib, convexity information of instances was frequently requested. Therefor, we now attempt to automatically prove or disprove convexity and concavity of objective and constraint functions. For quadratic functions, we use LAPACK to compute minimal and maximal eigenvalues of the Hessian Matrix. For general nonlinear functions, we also investigate the spectrum of the Hessian Matrix in one random point, which allows to disprove convexity in many cases. For the remaining instances, we apply the symbolic convexity checker that is implemented in the solver SCIP. This method applies some simple rules for the propagation of curvature information in an expression graph [7, Chapter 7.3.3] to prove "evident" convexity or concavity of functions. Finally, we applied an approach based on quantifier elimination to prove convexity of rational functions [6]. With these methods, we can disprove convexity for 57% of the instances and prove convexity for 31%.

4. Solution Information

Traditionally, the MINLPLib includes also solution points, i.e., known feasible solutions. For the current renovation, we corrected small infeasibilities in the existing points and added new points that improve incumbent solutions for some difficult instances. For each point, we now make the maximal constraint violation and coefficient ranges of the instance in this point available. Next to a representation in GAMS Data Exchange (GDX) binary format, solutions are now also available in text files.

While an incumbent solution provides a verifiable primal bound on the optimal value of an instance, providing a dual bound[13] in verifiable form (e.g., as branching tree with LP relaxations and their dual solution) is not feasible. To be still able to provide somewhat reliable information on the quality of a known primal bound, we collect the dual bounds that are provided by various MINLP solvers. If the bound that is reported by one solver is verified by at least two other solvers, we decide to trust this bound. Using this information, we can verify optimality of the incumbent solution for approximately half of the current instances.

5. Future Work

The here reported status constitutes a snap-shot (taken on 15th March 2014) of ongoing work on extending and updating the MINLPLib. We are currently in the process of adding further instances from publicly available sources and transmitted contributions. While so far only feasible (or not yet proven to be infeasible) instances were added, we hope to add also infeasible instances in the future. Additionally, we want to improve our tools to decide convexity and to recognize duplicates. Finally, we plan to extend our efforts to nonlinear programming problems (i.e., GLOBALLib) soon.

References

[1] M. R. Bussieck, A. S. Drud, and A. Meeraus. MINLPLib – a collection of test models for mixed-integer nonlinear programming. *INFORMS Journal on Computing*, 15(1):114–119, 2003.

[2] C. A. Floudas, P. M. Pardalos, C. S. Adjiman, W. R. Esposito, Z. H. Gumus, S. T. Harding, J. L. Klepeis, C. A. Meyer, and C. A. Schweiger. *Handbook of Test Problems in Local and Global Optimization*, volume 33 of *Nonconvex Optimization and Its Applications*. Kluwer Academic Publishers, 1999.

[3] D. M. Gay. Electronic mail distribution of linear programming test problems. Mathematical Programming Society COAL Newsletter, 1985.

[4] I. E. Grossmann and J. Lee. Cyberinfrastructure for mixed-integer nonlinear programming. *SIAG/OPT Views-and-News*, 22(1):8–12, 2011.

[5] T. Koch, T. Achterberg, E. Andersen, O. Bastert, T. Berthold, R. E. Bixby, E. Danna, G. Gamrath, A. M. Gleixner, S. Heinz, A. Lodi, H. Mittelmann, T. Ralphs, D. Salvagnin, D. E. Steffy, and K. Wolter. MIPLIB 2010 – mixed integer programming library version 5. *Mathematical Programming Computation*, 3(2):103–163, 2011.

[6] W. Neun, T. Sturm, and S. Vigerske. Supporting global numerical optimization of rational functions by generic symbolic convexity tests. In Vladimir P. Gerdt, Wolfram Koepf, Ernst W. Mayr, and Evgenii H. Vorozhtsov, editors, *Computer Algebra in Scientific Computing*, volume 6244 of *Lecture Notes in Computer Science*, pages 205–219. Springer, 2010.

[7] S. Vigerske. *Decomposition of Multistage Stochastic Programs and a Constraint Integer Programming Approach to Mixed-Integer Nonlinear Programming*. PhD thesis, Humboldt Universität zu Berlin, 2013.

[13] A dual bound is a lower (upper) bound on the optimal value of a minimization (maximization, resp.) problem.

Networks of Optimization Methods and Problems

Tamás Vinkó[1] and Kitti Gelle[2]

[1]*University of Szeged, Szeged, Hungary* tvinko@inf.u-szeged.hu

[2]*University of Szeged, Szeged, Hungary* Gelle.Kitti.Erzsebet@stud.u-szeged.hu

Abstract Benchmarking optimization methods and meaningful characterization of optimization problems have been the focal points of many research projects done in the field of global optimization. Our approach aims at investigating this topic with the usage of the computational and mathematical tools of network science. For a particular test problem a network formed by all the minima found by an optimization method can be constructed. Given these networks the analysis of their particular properties (e.g. degree distribution, path lengths, centrality measures, etc.) can lead to novel characterization of optimization problems and methods.

Keywords: benchmarking, network science

1. Introduction

Let $f : D \subset \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable. This work deals with optimization *problems* of the type

$$\min_{x \in D} f(x)$$

together with optimization *methods* belonging to the class of incomplete and asymptotically complete methods [1]. Several benchmarking techniques have been proposed already (see, e.g. [2, 3]) with the goal of giving hints on which optimization methods should be used in order to solve certain type of optimization problems in an efficient way. Our method complements these works with the help of the emerging field of network science.

2. Methodology

The proposed methodology takes inspirations from the early work of Stillinger and Weber [4], in which potential energy landscapes of some atomic clusters were formed into networks. The idea was that these landscapes can be divided into basins of attractions surrounding each locally minimal energy level. This approach was later successfully applied to the analysis of network topology of the potential function of small Lennard-Jones clusters [5]. In this case the so-called inherent structure network can be built in which nodes correspond to the minima and the edges link those minimum which are directly connected by a transition state. The same idea can be used for combinatorial optimization problems [6]. We give here a possible extension of these ideas to the space of continuous optimization problems and methods.

Given an optimization test problem, define a reasonable fine grid in its search space. Let x_S be a point on this grid. We take x_S as the starting point of the investigated optimization method. For each and every starting point the results of the optimization methods (i.e. the stationary points found from that starting point) is recorded. Now the stationary point network (SPN) can be constructed: the vertices of this graph are the stationary points found by the optimization method, and two vertices are connected if they were found from the same starting point. Note that in case of a deterministic solver this definition would never produce any edge, so in that case the definition needs to be modified. A simple example is given in

Section 3. Similar construction is used in [5] (and called inherent structure network) and in [6] (called local optima network).

Once these SPN graphs are constructed for each method and for each test functions, their properties could be used as comparison of the methods and problems in question.

Graph measures In the following we give an incomplete list of graph measures, taken from network science, together with their interpretations in the local optima networks context.

Size of the network is defined as the number of nodes. Clearly, this represents the number of local minima found by the optimizer.

Node degree is the number of edges a node has to other nodes. In our case this measures the number of adjacent stationary points. Related to this, it is worth considering the nodes degree distribution.

Average path length is defined as the average value of all shortest paths in the network. This measure indicates that how many non-local jumps should be taken, on average, from one basin to another to reach the one representing the global optimum value.

Diameter is the size of the longest of all shortest paths. This gives a worst-case scenario regarding the number of non-local jumps to reach the global minimum.

Betweenness centrality for a given node is calculated as the fraction of paths connecting all pairs of nodes and containing the node of interest. We hypothesize that the global optima have the highest betweenness centrality value.

3. Preliminary results

Currently we have a prototype framework in which two methods (a simple steepest decent (SD), and Differential Evolution (DE) [7]) are implemented along with some standard optimization tests from the classical Dixon-Szegő problem sets. We choose these two methods because their application in the proposed methodology must be clarified.

Firstly, SD is a simple example of the deterministic methods, i.e. it always produces the same result if it is started from a single point. Thus, the corresponding SPN does not contain any edge. In order to override this issue we propose here that upon starting from x_S it is checked if $\nabla f(x_S) = 0$, e.g. whether we start from a stationary point. (Practically, this is tested by checking if $\|\nabla f(x_S)\| < \epsilon$.) If x_S is a stationary point then the following 'multistart' type procedure is applied: give a small perturbation to x_S and start SD from there; repeat this for, say, 10 times. The resulting graph is shown on Figure 1. Note that for better visualization we do not show all the different points found by SD, only with those with positive degree.

Secondly, DE is a population based method, i.e. it uses more than one point during its run. Our proposed solution here is that the starting point x_S is always included in the first population (and obviously the other points in the population are selected randomly, as it is done in the standard DE). Regarding the result, the connected component of the graph produced by DE contains 665 nodes and 3848 edges. This case shows that the resulting graph contains much more vertices than the number of local/global minima of the function, which indicates the need for the introduction of further properties in the SPN, for example node weights.

Finally, we notice that the whole approach has particular relevance for problems with multiple local minima. In that case the resulting graph is expected to be large enough for the analysis by the network science tools. Detailed results will be given in the full version of the paper.

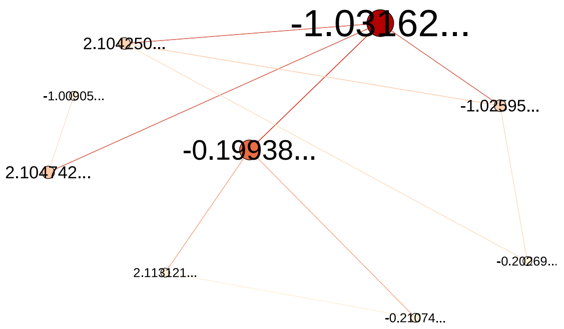


Figure 1: Stationary point network of SHCB using Steepest Decent method. Larger nodes represents higher betweenness centrality value.

Acknowledgments

This work was partially supported by the European Union and the European Social Fund through project FuturICT.hu (grant no.: TAMOP-4.2.2.C-11/1/KONV-2012-0013) and by the grant Aktion Österreich-Ungarn 87öu3. T. Vinkó was supported by the Bolyai Scholarship of the Hungarian Academy of Sciences.

References

[1] A. Neumaier, Complete search in continuous global optimization and constraint satisfaction, Acta Numerica 13(2004), 271–369.

[2] L. M. Rios and N. V. Sahinidis, Derivative-free optimization: A review of algorithms and comparison of software implementations, Journal of Global Optimization, 56(2013), 1247–1293.

[3] A. Neumaier, O. Shcherbina, W. Huyer, and T. Vinkó, A comparison of complete global optimization solvers, Mathematical Programming 103(2005), 335–356.

[4] F.H. Stillinger and T.A. Weber, Packing structures and transitions in liquids and solids. Science 225 (1984), 983–989.

[5] J.P.K. Doye, The network topology of a potential energy landscape: A static scale-free network, Phys. Rev. Lett. 88, 238701 (2002)

[6] M. Tomassini, S. Verel, and G. Ochoa, Complex-network analysis of combinatorial spaces: The NK landscape case. Physical Review E 78.6 (2008): 066114.

[7] 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.

Proceedings of MAGO 2014, pp. 145 – 148.

Optimization in Surgical Operation Design

Tibor Csendes[1] and István Bársony[2]

[1] *University of Szeged, Institute of Informatics, P.O. Box 652, Szeged, Hungary,* csendes@inf.u-szeged.hu

[2] *Kecskemét College, Kecskemét, Hungary,* barsony.istvan@gamf.kefo.hu

Abstract A new treatment of oncological diseases is brachytherapy that means the insertion of low level radiation isotopes into the organ to be healed. This cure has much less side effects than traditional radiation therapy, while being as much effective. The problem is to determine how to position the 40-90 capsules in such a way that the tissue to be healed should obtain at least a given level of dose, while the surrounding other organs should absorb a dose less than a prescribed level. The related nonlinear optimization problem is of medium dimensionality (120-270). The global optimization problem is very redundant, and it has several forms of symmetries as well. The present work aims to speed up the optimization, to allow different intensity radiation capsules, and to decrease the cost of the treatment. The first test results obtained for artificial models are reported.

Keywords: Operation design, Sphere packing

1. Introduction

Some cancers can be treated by a new method, by inserting low radiation level material into the given organ. This technique, called brachytherapy has been accepted by the Hungarian authorities a few years ago, and is used in the developed world since a decade.

Usually 40-80 pieces of small capsules (called seeds, see Figure 1) are inserted that contain the radiating material. The operation requires local anesthesia, and can be carried out in an ambulant way. Brachytherapy has minimal side effects compared to traditional radiation therapy, while it is at least as effective.

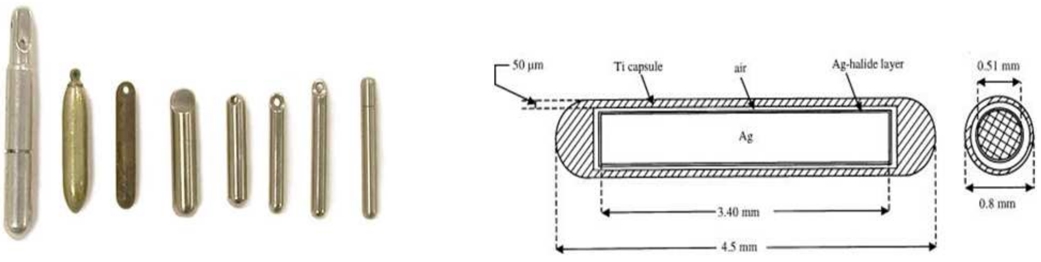


Figure 1: Seeds to be inserted containing the isotopes, and the structure of the seeds.

Brachytherapy costs around 5,000 Euros per patient. The remained pieces of the ordered set of seeds are now handled as dangerous waste requiring expensive handling.

Brachytherapy is used for cancers in the brain, thyroid gland, neck, breast, and prostate. The healing effect is based on the fact that tumor cells are less effective in repairing errors in the genes. The obtained radiation dose is comparable with that of the diagnostic imaging procedures such as CT and xray.

The optimization method to be presented can help in answering the following questions:

- How to place a given number of seeds with known common radiation intensity in such a way, that gives the required radiation dose to the tissues with tumors, while the surrounding other tissues receive minimal radiation only?
- Which setting of the seeds allows to reach the therapy aims with a minimal number of seeds?
- The present surgical operation design method requires 0.5-1.5 hours of computation, while the patient waits anesthetized. Can this part of the therapy be speeded up?
- Is it possible to provide an acceptable solution for seeds of different radiation intensity (saving much money)?

We applied the following simple model for the computational tests: The tumorous tissue to be treated is a sphere with center at the origin, and radius 3. The tissues to be saved from the radiation are outside the sphere, and a vertical cylinder around the axe z with a diameter of 1. The radiation dose is assumed to be the same within each cube of side length 0.3, and the dose is calculated in its center. Only those cubes were considered that belong exclusively to one of the mentioned two tissues. A surgical operation design is regarded to be acceptable, if at least 90% the tumorous tissue obtains radiation dose of 110 units, and at least 90% of the tissues to be saved gets at most 90 units of radiation.

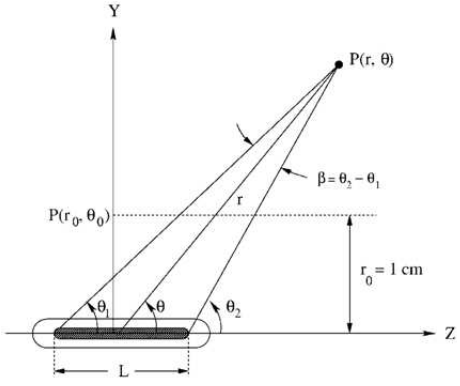


Figure 2: Dose calculation for a not point-like source.

Dose calculation details are from the paper of Tibor Major and Jenő Julow [3] (Figure 2). The half life of the isotope iodine-125 is 60 days, the average energy density is 35,5 keV, and it emits gamma-radiation. The half life of the isotope iridium-192 is 74 days, the average energy density is 370 keV, and it emits beta-radiation. In our simplified model we do not calculate with the anisotropy (see Figure 3) of the radiation sources.

2. Optimization model

Let $S_{in} = \{(x, y, z) \mid x^2+y^2+z^2 \leq 3^2, z^2 \geq 1\}$ be the tissue to be treated, and $S_{out} = [-3, 3]^3 \backslash S_{in}$ that to be saved. Then the optimization problem is

$$\max_{x_i, y_i, z_i \in [-3,3]^3} \frac{(\mathrm{vol}(\{s \in S_{in} \mid D(s) \geq 110\}) + \mathrm{vol}(\{s \in S_{out} \mid D(s) \leq 90\}))}{(\mathrm{vol}(S_{in}) + \mathrm{vol}(S_{out}))},$$

where $D(s)$ is the cumulative dose obtained at point s from all seeds, and $\mathrm{vol}(S)$ is the volume of the set S. The function $D(s)$ that calculates the summed up dose from the point-like radiation sources, is proportional with the reciprocal of the square of the distance to the seed.

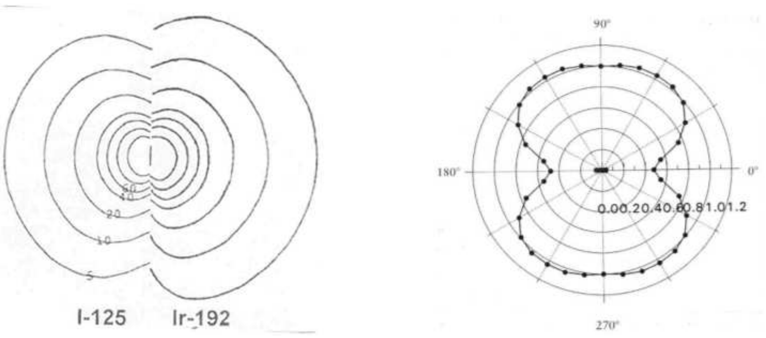


Figure 3: Curves with equivalent dose values for the isotopes iodine-125 and iridium-192, and illustration of anisotrophy.

The cumulative dose constraints mean basically a kind of generalization of sphere covering of bounded bodies. In this way the optimal surgery operation design problem belongs to discrete geometrical optimization.

The algorithm development was made in a Matlab environment. The final implementation is planed to be in a high level programming language like C. The GLOBAL algorithm [1], a multistart method was used for optimization. Due to the high dimensionality of the problem, the random walk type local search technique, UNIRANDI was applied, instead of the BFGS quasi-Newton algorithm. Since the problem is highly symmetric, such as the circle packing problem, similar tricks should be used to have an efficient and effective algorithm.

3. First results

A representative Matlab setting of the computational testing without the trics utilizing the symmetries and redundancies:

```
>> LB = [-3; -3; ...  -3];
>> UB = [3; 3; ...  3];
>> OPTS.N100 = 50;
>> OPTS.NG0 = 2;
>> OPTS.NSIG = 6;
>> OPTS.MAXNLM = 3;
>> OPTS.METHOD = 'unirandi';
>> FUN =@brachy;
>> [X0,F0,NC,NFE] = GLOBAL(FUN, LB, UB, OPTS);
```

A typical result when the algorithm settings were: resolution 0.2, 10 seeds, number of sample points: 500, tolerance value 10^{-8}. At most 3 local minima was allowed, the local search method was UNIRAND, and the running time up to 25 seconds.

```
>> [X0,F0,NC,NFE] = GLOBAL(FUN, LB, UB, OPTS);
*** TOO MANY CLUSTERS ***
NORMAL TERMINATION AFTER 1414 FUNCTION EVALUATIONS

  LOCAL MINIMA FOUND: 3
  F0 =
-0.8023
-0.6991
-0.6976

  X0 =
0.2743 -0.8013 2.3628
1.2200 0.7336 0.3030
-0.4384 1.2329 -0.1015
```

The next step of algorithm development is to utilize the known sparsest covering structure of spheres in the space, reduce the dimensionality of the optimization problem, and force a unique identification of the seeds. This would be in accordance with what we have learned in the determination of optimal circle packing [4, 5]. Also other global optimization methods can help, such as the Multilevel Coordinate Search [2].

4. Summary

We have achieved an 80% quality solution for the oversimplified model (10 seeds, point-like radiation source) with minimal computation time. The solution quality got worse with increasing the dimensionality – possibly due to the decreasing relative size of the set of feasible solutions. The computation times experienced are promising (taking into account that the interpreter mode functioning of Matlab results in a ca. 500 times slower execution than a high level language implementation). The resolution of the search space influences the CPU time in a cubic way.

Acknowledgments

The research was supported by the European Union and the European Social Fund through the project National Excellence in the Convergence Region (grant TAMOP-4.2.4-A/2) in the frame of a János Szentágothai Fellowship. Special thanks to Tibor Major (Hungarian National Institute of Oncology) for his help in the doses calculations.

References

[1] T. Csendes, L. Pál, J.O.H. Sendín, and J.R. Banga: The GLOBAL Optimization Method Revisited. Optimization Letters 2(2008) 445-454

[2] W. Huyer and A. Neumaier: Global optimization by multilevel coordinate search. J. Global Optimization 14(1999) 331-355

[3] T. Major and J. Julow: Dosimetry for the stereoaxial intersticial brachytherapy of inoperable brain tumors, in Hungarian, Magyar Onkológia 42(1998) 39-44

[4] M.Cs. Markót and T. Csendes: A new verified optimization technique for the "packing circles in a unit square" problems. SIAM J. on Optimization 16(2005) 193-219

[5] P.G. Szabó, M.Cs. Markót, T. Csendes, E. Specht, L.G. Casado, and I. García: New Approaches to Circle Packing in a Square – With Program Codes Springer, Berlin, 2007

An Efficient Approach for Solving Uncapacitated Facility Location Models with Concave Operating Costs

Robert Aboolian,[1] Emilio Carrizosa [2] and Vanesa Guerrero[2]

[1] *College of Business Administration, California State University San Marcos, San Marcos, California,* raboolia@csusm.edu

[2] *Instituto de Matemáticas de la Universidad de Sevilla (IMUS), University of Seville, Spain,* ecarrizosa@us.es vguerrero@us.es

Abstract We consider a nonlinear version of the Uncapacitated Facility Location Problem (UFLP). The total cost, to be minimized, has two parts: the transportation costs, supposed to be linear in the (client,plant) allocations, and the operation costs, which are here assumed to be given by a concave nondecreasing function of the demand served by each open facility. Thus we call the problem Uncapacitated Facility Location Problem with Concave Operating Cost (UFLPCOC). The problem is modeled and an exact solution approach is presented. This approach is mainly based on obtaining efficient lower and upper bounds for UFLPCOC. Lower bounds are obtained by solving a UFLP with extra linear constraints. To find an upper bound, we present a heuristic which is based on a neighborhood search over the location set from the solution to the previous Integer Linear Program. The exact approach is based on successive lower and upper bound improvements for UFLPCOC until convergence is obtained. Computational results are presented.

Keywords: Nonlinear Uncapacitated Facility Location Problem, Cutting planes, Integer Concave Programming

1. Introduction

In this work we consider a nonlinear version of the Uncapacitated Facility Location Problem (UFLP), see [2], in which the total cost, to be minimized, has two parts: the transportation costs, supposed to be linear in the (client,plant) allocations, and the operation costs, which are here assumed to be given by a concave nondecreasing function of the demand served by each open facility. Thus we call the problem Uncapacitated Facility Location Problem with Concave Operating Cost (UFLPCOC). The problem is modeled in Section 2 and an exact solution approach is presented in Section 3. This approach is mainly based on obtaining efficient lower and upper bounds for UFLPCOC. Lower bounds are obtained by solving a UFLP with extra linear constraints. To find an upper bound, we present a heuristic which is based on a neighborhood search over the location set from the solution to the previous Integer Linear Program, which follows the methodology used in [1]. The exact approach is based on successive lower and upper bound improvements for UFLPCOC until convergence is obtained. Computational results are presented.

2. Problem statement

Let $N(|N| = n)$ be the set of customer demand aggregation points and $M(|M| = m)$ be the set of candidate locations for the facilities. We denote demand rate at node i as λ_i for each $i \in N$. For each facility at node $j \in M$, denote by $f_j \geq 0$ the fixed cost to locate a facility at node j and by $F(\Lambda_j)$ the variable cost to allocate a capacity of Λ_j at facility at node j. Assume that $F(\Lambda_j)$ is a concave function in Λ_j. Let c_{ij} be the access cost of one unit of demand from node i at a facility located at node j. We will use $S \subseteq M$ to denote the set of facilities selected. If a facility is located at site j, we call it facility j. Like UFLP each customer is assumed to be served by the closest open facility.

Let us write the problem as an Integer Nonlinear Program. Let x_j be a binary variable which is one if we open a facility at j and zero otherwise. Let y_{ij} be a binary variable which is one if facility j is assigned to customers at i and zero otherwise. Given the above definitions, the Uncapacitated Facility Location Problem with Concave Operating Cost problem ($UFLPCOC$) is formulated as:

$$\min \sum_{j \in M} \left(f_j x_j + \sum_{i \in N} F(\lambda_i y_{ij}) \right) + \sum_{i \in N} \sum_{j \in M} \lambda_i c_{ij} y_{ij} \tag{1}$$

s.t.

$$\sum_{j \in M} y_{ij} = 1, \ \forall i \in N, \tag{2}$$

$$y_{ij} \le x_j \ \forall i \in N, \ j \in M, \tag{3}$$

$$\sum_{k \in M} c_{ik} y_{ik} \le (c_{ij} - L) x_j + L, \ \forall i \in N, \ j \in M, \tag{4}$$

$$x_j, y_{ij} \in \{0, \ 1\}, \ \forall i \in N, \ j \in M. \tag{5}$$

The objective function (1) is the sum of the transportation costs and the facilty fixed and variable costs. Constraints (2) ensure that each customer is assigned to just one facility, while constraints (3) enforce the customers to be only assigned to the open facilities. Constraints (4) are to ensure that each customer is served by the least access cost open facility. In constraints (4) L is a large-enough positive number (e.g. $L = \max_{j \in M, i \in N}\{c_{ij}\}$).

Note that $UFLPCOC$ is an integer nonlinear program which is large-scale in nature. To efficiently solve this problem we propose an exact solution approach.

3. Algorithms for UFLPCOC

Our exact solution approach is mainly based on obtaining efficient lower and upper bounds for $UFLPCOC$. In Section 3.1, we develop an MIP, for which the objective function value of its optimal solution provides a lower bound for $UFLPCOC$. In Section 3.2, we present a heuristic, which is based on a neighborhood search over the location set from the solution to the MIP. It is used to find an upper bound for $UFLPCOC$. The exact approach presented in 3.3 is based on successive lower and upper bound improvements for $UFLPCOC$.

3.1 A lower bound for $UFLPCOC$

In order to determine a lower bound for $UFLPCOC$, the concave function $F(\Lambda_j)$ is replaced by a linear function $F^L(\Lambda_j)$ such that $F^L(\Lambda_j) \le F(\Lambda_j)$ for $\Lambda_j^{\min} \le \Lambda_j \le \Lambda_j^{\max} \ \forall j \in M$. Since $F(\Lambda_j)$ is concave, every chord lies below the graph of F. Thus, the condition required above is satisfied if a chord is taken as F^L.

Thus, being $F^L(\Lambda_j) = a_j \Lambda_j + b_j$ a chord of $F(\Lambda_j)$, $j = 1, \ldots, M$ and by replacing $F(\Lambda_j)$ with $F^L(\Lambda_j)$ in $UFLPCOC$ will result the following mixed integer program, which we call $LBMIP$:

$$\begin{aligned}
&\min \sum_{j\in M}(f_j+b_j)x_j+\sum_{i\in N}\sum_{j\in M}\lambda_i(a_j+c_{ij})y_{ij}\\
&\text{s.t.}\\
&\quad (2)-(3)\\
&\quad x_j\in\{0,\ 1\},\ y_{ij}\geq 0,\ \forall i\in N,\ j\in M.
\end{aligned}$$

3.2 An upper bound for $UFLPCOC$

We note that any feasible location vector $\mathbf{x}$ including the one produced by solving $LBMIP$ generates a feasible solution to $UFLPCOC$. This is achieved by first defining the assignment vector $\mathbf{y}(\mathbf{x})$ using the assumption that customers are assigned to open facilities level by level in an increasing order of shipping cost. Then, the resulting value of the objective function (1), provides an upper bound for $UFLPCOC$.

Denote by $S_{\mathbf{x}}$ the set of facility locations, which are open given location vector $\mathbf{x}$. To find an improved upper bound, the heuristic uses a descent approach in a neighborhood search for $S_{\mathbf{x}}-$ the location set produced by solving $LBMIP$. The distance-k neighborhood of $S\subseteq M$ is defined as

$$N_k(S)=\{S'\subseteq M:|S-S'|+|S'-S|\leq k\}$$

i.e., S' is in the distance-k neighborhood of S if the number of non-overlapping elements in the two sets does not exceed k.

Once the neighborhood is well defined, the descent algorithm is straightforward: use the solution to $LBMIP$ as a starting subset $S_{\mathbf{x}}$; evaluate the change in the value of the objective function (1) for all the subsets in the neighborhood; if an improved subset exists in the neighborhood, move the search to the best vector in the neighborhood. Repeat the process with the new subset until no improved vector exists in the neighborhood. The last subset is the solution.

Denote by $S_{\overline{\mathbf{x}}}$ (the set of facility locations under vector $\overline{\mathbf{x}}$) the solution subset to the descent approach, and let $\mathbf{y}(\overline{\mathbf{x}})$ be the assignment vector where customers are assigned to least access cost open facilities at $S_{\overline{\mathbf{x}}}$. The resulting value of the objective function (1) is our new upper bound and the solution to the descent approach $\overline{\mathbf{x}}$ and $\mathbf{y}(\overline{\mathbf{x}})$ is the solution to our heuristic.

3.3 An exact approach for $UFLPCOC$

The exact approach presented is based on successive improvements on lower and upper bounds on $UFLPCOC$ in each step of the algorithm. In this approach, we first find initial lower and upper bounds for $UFLPCOC$ by solving the heuristic proposed in Section 3.2. In the next step, we find an improved lower bound by solving an improved $LBMIP$. An improved $LBMIP$ is $LBMIP$ with additional cuts, which exclude the pre-examined location vectors from the feasible region (at the first step $LBMIP$ is solved without any cuts), and with a tighten objective function, which is found by solving $2M$ integer programs. After finding an improved lower bound, the location set produced by solving the improved $LBMIP$ is used as a starting location set in a neighborhood search to find an improved upper bound using the descent approach named in Section 3.2. The procedure continues until the lower bound is greater than the upper bound, so that it is evident that the unexamined location sets are unable to improve the current upper bound.

4. Conclusions

An efficient approach for solving uncapacitated facility location models with concave operating costs is presented. Preliminary computational results are being carried out on a PC Intel®

Core$^{\text{TM}}$ i7-2600K, 16GB of RAM. We use the optimization engine CPLEX v12.4 (CPLEX 2012) for solving all optimization problems.

Acknowledgments

This research is funded in part by projects MTM2012-36163 (Ministerio de Economía y Competitividad, Spain), P11-FQM-7603 (Junta de Andalucía), both supported by EU ERD funds, and by Fundación Cámara.

References

[1] R. Aboolian, T. Cui and Z.J.M. Shen. *An efficient Approach for Solving Reliable Facility Location Models. Addison Wesley, Massachusetts, INFORMS Journal on Computing, 25(4), 720-729, 2012.*

[2] G. Cornuéjols, G.L. Nemhauser and L.A. Wolsey The uncapacitated facility location problem, in: Discrete Location Theory, eds. P:B: Mirchandani and R.L. Francis, Wiley, New York, 119-171, 1990.

An Introduction to Lipschitz Global Optimization[*]

Yaroslav D. Sergeyev[1,2]

[1] *DIMES, University of Calabria, Via P. Bucci, 42C – 87036, Rende (CS), Italy*

[2] *Software Department, N. I. Lobachevsky State University, Nizhni Novgorod, Russia*
yaro@si.dimes.unical.it

Abstract This lecture deals with the global optimization problems where the objective function can be "black box", multiextremal, and possibly non-differentiable. It is also assumed that evaluation of the objective function at a point is a time-consuming operation. Two statements of the problem are taken into consideration: (i) the objective function satisfies the Lipschitz condition; (ii) the gradient of the objective function satisfies the Lipschitz condition. Two cases are considered for both problems: the Lipschitz constant is either known a priori or unknown (in this case it should be estimated). Local tuning on the behavior of the objective function and a new technique, named *local improvement*, are used in order to accelerate the search. Convergence condition are given and extensive numerical experiments are presented.

Keywords: Lipschitz global optimization, Numerical methods, Partition strategies, Peano-Hilbert space-filling curves

1. Introduction

Global optimization is a thriving branch of applied mathematics and an extensive literature has been dedicated to this field (see, e. g., [1–27]). In this lecture, the global optimization problem of a multidimensional function satisfying the Lipschitz condition over a hyperinterval with an unknown Lipschitz constant is considered:

$$f^* = f(x^*) = \min_{x \in D} \; f(x), \tag{1}$$

$$|f(x') - f(x'')| \le L\|x' - x''\|, \quad x', x'' \in D, \tag{2}$$

where $L, \; 0 < L < \infty$, is called the Lipschitz constant,

$$D = [a, b] = \{x \in R^N : a(j) \le x(j) \le b(j)\}, \tag{3}$$

and $\| \cdot \|$ denotes, usually, the Euclidean norm (however, other norms can be also used). It is supposed that the objective function can be "black box", multiextremal, and non-differentiable. It is also assumed that evaluation of the objective function at a point is a time-consuming operation. Two statements are considered: the Lipschitz constant L is either known a priori or unknown (in this case it should be estimated).

A particular class of the Lipschitz global optimization problems is also discussed in this lecture, namely, the class of problems with differentiable objective functions having the Lipschitz gradient $f'(x)$, i.e.,

$$\|f'(x') - f'(x'')\| \le K\|x' - x''\|, \quad x', x'' \in D, \quad 0 < K < \infty. \tag{4}$$

Again, similarly to the situation regarding the constant L, the constant K can be either known a priori or unknown and, therefore, should be estimated in a way.

[*] This research was partially supported by the INdAM–GNCS 2014 Research Project of the Italian National Group for Scientific Computation of the National Institute for Advanced Mathematics "F. Severi".

2. One-dimensional techniques

Numerous algorithms for solving problems (1) – (4) have been discussed in the literature. They can be distinguished, for example, by the way of obtaining information about the Lipschitz constant and by the strategy of exploration of the search domain. Due to importance of the one-dimensional methods for constructing multi-dimensional generalizations we give an illustration how 12 one-dimensional methods work on the function no. 38 from the set of test functions from [15]. Fig. 1 shows the graph of the function and the points where the objective function has been evaluated by the 12 methods while minimizing this function, with accuracy $\varepsilon = 10^{-4}(b-a)$. The global minimum of the function, $f^* = 0$, is attained at the point $x^* = 3.3611804993$. The following methods (see [12] for their detailed description and an extensive testing) have been used in this examples:

- **PKC**: the basic method, called hereinafter GS, constructing piece-wise linear auxiliary functions, i.e., **P**iyavskii's method with the a priori **K**nown **C**onstant L;
- **GE**: GS using the **G**lobal **E**stimate of the Lipschitz constant L;
- **LT**: GS executing the **L**ocal **T**uning on the local Lipschitz constants;
- **PKC_LI**: GS with the a priori **K**nown **C**onstant L enriched by the **L**ocal **I**mprovement technique;
- **GE_LI**: GS using the **G**lobal **E**stimate of L enriched by the **L**ocal **I**mprovement technique;
- **LT_LI**: GS executing the **L**ocal **T**uning on the local Lipschitz constants enriched by the **L**ocal **I**mprovement technique;
- **DKC**: the basic method from [17], called hereinafter GS_D, constructing smooth piece-wise quadratic auxiliary functions using the first **D**erivatives and the a priori **K**nown Lipschitz **C**onstant K;
- **DGE**: GS_D using the first **D**erivatives and the **G**lobal **E**stimate of the constant K;
- **DLT**: GS_D using the first **D**erivatives and the **L**ocal **T**uning;
- **DKC_LI**: GS_D using the first **D**erivatives, the a priori **K**nown Lipschitz **C**onstant K, and the **L**ocal **I**mprovement technique;
- **DGE_LI**: GS_D using the first **D**erivatives, the **G**lobal **E**stimate of the Lipschitz constant K, and the **L**ocal **I**mprovement technique);
- **DLT_LI**: GS_D using the first **D**erivatives, the **L**ocal **T**uning, and the **L**ocal **I**mprovement.

3. Multi-dimensional generalizations

Different exploration techniques based on various adaptive partition strategies are analyzed. The main attention is dedicated to two types of algorithms. The first of them is based on using space-filling curves in global optimization. A family of derivative-free numerical algorithms applying space-filling curves to reduce the dimensionality of the global optimization problem is discussed. A number of unconventional ideas, such as adaptive strategies for estimating Lipschitz constant, balancing global and local information to accelerate the search, etc. are presented.

Diagonal global optimization algorithms is the second type of methods under consideration. They have a number of attractive theoretical properties and have proved to be efficient

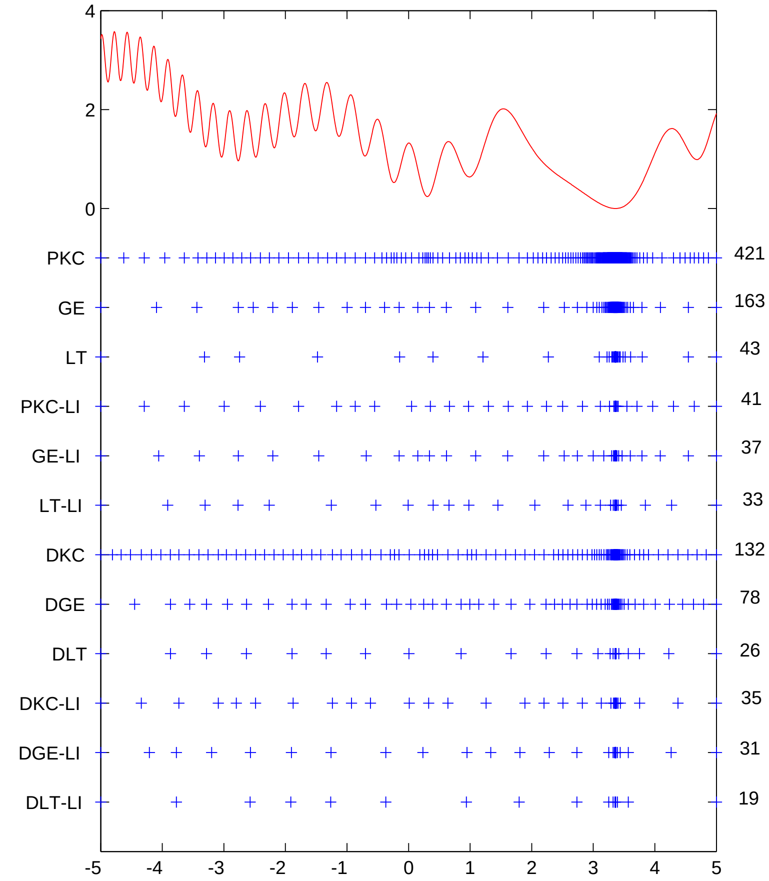


Figure 1: *Graph of the function number 38 from [15] and trial points generated by the 12 methods tested.*

in solving applied problems. In these algorithms, the search hyperinterval is adaptively partitioned into smaller hyperintervals and the objective function is evaluated only at two vertices corresponding to the main diagonal of the generated hyperintervals. It is demonstrated that the traditional diagonal partition strategies do not fulfil the requirements of computational efficiency because of executing many redundant evaluations of the objective function.

A new adaptive diagonal partition strategy that allows one to avoid such computational redundancy is described. Some powerful multidimensional global optimization algorithms based on the new strategy are introduced. Results of extensive numerical experiments performed on the GKLS-generator (see [2]) to test the proposed methods demonstrate their advantages with respect to traditional diagonal algorithms in terms of both number of trials of the objective function and qualitative analysis of the search domain, which is characterized by the number of generated hyperintervals. A number of directions of possible developments is discussed briefly. Among them we can mention problems with multiextremal partially generated constraints, the usage of parallel non-redundant computations, and theoretical results on the possible speed-up.

References

[1] C. A. Floudas and P. M. Pardalos, *State of the Art in Global Optimization*. Kluwer, Dordrecht, 1996.

[2] M. Gaviano, D. E. Kvasov, D. Lera, and Ya. D. Sergeyev, "Algorithm 829: Software for generation of classes of test functions with known local and global minima for global optimization", *ACM Transactions on Mathematical Software*, **29**(4):469–480, 2003.

[3] S. Yu. Gorodetsky, "Paraboloid triangulation methods in solving multiextremal optimization problems with constraints for a class of functions with Lipschitz directional derivatives", *Vestnik of Lobachevsky State University of Nizhni Novgorod*, **1**(1):144–155, 2012. In Russian.

[4] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, "Lipschitzian optimization without the Lipschitz constant", *Journal of Optimization Theory and Applications*, **79**: 157–181, 1993.

[5] R. Horst and P. M. Pardalos, *Handbook of Global Optimization*, Kluwer, Dordrecht, 1995.

[6] D. E. Kvasov, D. Menniti, A. Pinnarelli, Ya. D. Sergeyev, and N. Sorrentino, "Tuning fuzzy power-system stabilizers in multi-machine systems by global optimization algorithms based on efficient domain partitions", *Electric Power Systems Research*, **78**(7):1217–1229, 2008.

[7] D. E. Kvasov and Ya. D. Sergeyev, "A univariate global search working with a set of Lipschitz constants for the first derivative", *Optimization Letters*, **3**(2):303–318, 2009.

[8] D. E. Kvasov and Ya. D. Sergeyev, "Lipschitz gradients for global optimization in a one-point-based partitioning scheme", *Journal of Computational and Applied Mathematics*, **236**(16):4042–4054, 2012.

[9] D. E. Kvasov and Ya. D. Sergeyev, "Univariate geometric Lipschitz global optimization algorithms", *Numerical Algebra, Control and Optimization*, **2**(1):69–90, 2012.

[10] D. Lera and Ya. D. Sergeyev, "Lipschitz and Holder global optimization using space-filling curves", *Applied Numerical Mathematics*, **60**(1-2):115–129, 2010.

[11] D. Lera and Ya. D. Sergeyev, "An information global minimization algorithm using the local improvement technique", *Journal of Global Optimization*, **48**(1):99–112, 2010.

[12] D. Lera and Ya. D. Sergeyev, "Acceleration of univariate global optimization algorithms working with Lipschitz functions and Lipschitz first derivatives", *SIAM Journal on Optimization*, **23**(1):508–529, 2013.

[13] R. Paulavičius and J. Žilinskas, *Simplicial Global Optimization*, Springer, New York, 2014.

[14] J. D. Pintér, *Global Optimization in Action (Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications*, Kluwer, Dordrecht, 1996.

[15] J. D. Pintér, Global optimization: software, test problems, and applications, In P. M. Pardalos and H. E. Romeijn, editors, *Handbook of Global Optimization*, volume 2, pages 515–569. Kluwer Academic Publishers, Dordrecht, 2002.

[16] Ya. D. Sergeyev, "An information global optimization algorithm with local tuning", *SIAM Journal on Optimization*, **5**(4):858–870, 1995.

[17] Ya. D. Sergeyev, "Global one-dimensional optimization using smooth auxiliary functions", *Mathematical Programming*, **81**(1):127–146, 1998.

[18] Ya. D. Sergeyev, P. Pugliese, D. Famularo, "Index information algorithm with local tuning for solving multidimensional global optimization problems with multiextremal constraints", *Mathematical Programming*, **96**(3):489–512, 2003.

[19] Ya. D. Sergeyev and D. E. Kvasov, "Global search based on efficient diagonal partitions and a set of Lipschitz constants", *SIAM Journal on Optimization*, **16**(3):910–937, 2006.

[20] Ya. D. Sergeyev and D. E. Kvasov, *Diagonal Global Optimization Methods*, FizMatLit, Moscow, 2008. In Russian.

[21] Ya. D. Sergeyev and D. E. Kvasov, "Lipschitz global optimization", in J. J. Cochran et al., (Eds.), *Wiley Encyclopedia of Operations Research and Management Science (in 8 volumes)*, John Wiley & Sons, New York, **4**:2812–2828, 2011.

[22] Ya. D. Sergeyev, R. G. Strongin, and D. Lera, *Introduction to Global Optimization Exploiting Space-Filling Curves*, Springer, New York, 2013.

[23] R.G. Strongin, *Numerical Methods in Multi-Extremal Problems (Information-Statistical Algorithms)*, Nauka, Moscow, 1978. In Russian.

[24] R. G. Strongin and Ya. D. Sergeyev, *Global Optimization with Non-Convex Constraints: Sequential and Parallel Algorithms*, Kluwer, Dordrecht, 2000.

[25] A. A. Zhigljavsky, *Theory of Global Random Search*, Kluwer, Dordrecht, 1991.

[26] A. A. Zhigljavsky and A. Žilinskas, *Stochastic Global Optimization*, Springer, New York, 2008.

Solving a Huff-like Stackelberg problem on networks[*]

Kristóf Kovács and Boglárka G.-Tóth

Budapest University of Technology and Economics, Hungary, kkovacs@math.bme.hu, bog@math.bme.hu

Abstract This work deals with a Huff-like Stackelberg problem, where the leader facility wants to decide its location so that its profit is maximal after the competitor (the follower) also built its facility. It is assumed that the follower makes a rational decision, maximizing their profit. The inelastic demand is aggregated into the vertices of a graph, and facilities can be located along the edges.

This Stackelberg model is a bi-level problem that makes global solvability extremely hard. Even though the problem is tackled by a Branch and Bound method, so that global optimality is provided.

Keywords: Branch&Bound, Stackelberg problem, facility location, Interval Analysis, DC decomposition, global optimization, bi-level problem

1. Introduction

In competitive facility location the general aim is to locate one or more new facilities for an existing or a newcomer chain maximizing its market share or profit. When competitors are likely to react with their own expansion, the owner has to take that into account. This leads to a bi-level optimization problem, where the optimal location of the first player, the *leader*, has to be determined depending on the location of the second player, the *follower*, who decides its location with the knowledge of the location of the leader. This problem is called the Stackelberg problem, or the (r, p)-centroid problem when r leader and p follower facilities are located.

The underlying location problem depends on many factors starting from the decision space, through properties of the demand till costumer's choices. In this work static competition with inelastic demand is considered. Demand is concentrated in a discrete set of points, called demand points. Costumers are assumed to follow the probabilistic choice for the facilities, i.e. they split their demand proportionally to the attraction they feel to the facilities. Attraction of a facility determined by its quality and the distances to it, through a gravitational or logit type model. The objective function to be maximized is the profit obtained by the chain, to be understood as the income due to the market share captured by the chain minus its operational costs. The location space in our model is a network, with the vertices being demand points and the facilities located on its edges.

Many papers dealing with Stackelberg problems assume binary costumer choice, that allows to narrow the solution candidates to a discrete set of points, transforming it a combinatorial optimization problem [5], or already offering only a discrete set for the locations [2]. Continous problems on the plane with analogous objectives has been addressed by [1, 3] offering heuristic methods. In [4] a similar problem was proposed and solved reliably, although on a planar space for the maximization of the market share.

[*] This research has been supported by the Ministry of Economy and Competitiveness of Spain under the research project ECO2011-24927, in part financed by the European Regional Development Fund (ERDF), and the Fundación Séneca (The Agency of Science and Technology of the Region of Murcia) under the research project 15254/PI/10 and by Junta de Andalucía (P11-TIC7176.)

2. Problem Formulation

Let us now introduce formally the problem under consideration. Let us given a network $N = (V, E)$, where each $e_{ij} \in E$ refers to the edge with end points a_i and $a_j \in V$ denoting its length by $l_{e_{ij}}$. It allows us to talk about points in an edge: edge e_{ij} is identified with the interval $[0, l_{e_{ij}}]$, and we thus denote any $x \in [0, l_{e_{ij}}]$ by the point in the edge e_{ij} at distance x of a_i and distance $l_{e_{ij}} - x$ of a_j.

The demand is concentrated at the vertices of N, where each $a \in V$ has associated its buying power ω_a. The function $d_a(x)$ gives the distance between demand point a and facility x. Assuming that x is located on edge e_{ij}, it is calculated as follows

$$d_a(x) = \min\{x + d(a_i, a), l_{e_{ij}} - x + d(a_j, a)\}$$

where $d(a_i, a)$ is the length of the shortest path from demand point a_i to a.

In a competitive environment it is usual to assume that both firms have preexisting facilities. Considering m existing facilities, we refer to the leader's facilities as x_i $i = 1 \ldots k$ and to the follower owed ones as x_j $j = k + 1 \ldots m$. Every facility is given its fixed quality q_i for $i = 1, \ldots m$. The new facility of the leader and the follower is denoted by the index l and f, respectively, thus the location of the leader's new facility is denoted by x_l its quality by q_l, similarly we have x_f and q_f for the follower.

The market share captured by the leader (with new facility at x_l) after the follower locates at x_f is

$$M_l(x_l, x_f) = \sum_{a \in V} \omega_a \frac{q_l/\varphi_a(d_a(x_l)) + \sum_{j=1}^{k} q_j/\varphi_a(d_a(x_j))}{q_l/\varphi_a(d_a(x_l)) + q_f/\varphi_a(d_a(x_f)) + \sum_{j=1}^{m} q_j/\varphi_a(d_a(x_j))},$$

while the market share captured by the follower is

$$M_f(x_l, x_f) = \sum_{a \in V} \omega_a \frac{q_f/\varphi_a(d_a(x_f)) + \sum_{j=k+1}^{m} q_j/\varphi_a(d_a(x_j))}{q_l/\varphi_a(d_a(x_l)) + q_f/\varphi_a(d_a(x_f)) + \sum_{j=1}^{m} q_j/\varphi_a(d_a(x_j))}.$$

The function φ is a positive nondecreasing function on non negative values. The usual choice is $\varphi(t) = t^\lambda$, where $\lambda = 2$ gives the so called gravitational model. Both firms are assumed to have renting and/or operational costs depending on the facility's proximity to demand points. Locations near highly populated areas likely to be more expensive, therefore

$$G_l(x_l) = \sum_{a \in V} \omega_a \frac{q_l}{\psi_a(d_a(x_l))}, \quad G_f(x_f) = \sum_{a \in V} \omega_a \frac{q_f}{\psi_a(d_a(x_f))}$$

are considered, where ψ is a similar function to φ, though the two should not be the same. Thus the profit of the two firms are

$$F_l(x_l, x_f) = M_l(x_l, x_f) - G_l(x_l), \quad F_f(x_l, x_f) = M_f(x_l, x_f) - G_f(x_l).$$

Using the previous functions we can formulate the objective of the leader problem as

$$\begin{aligned} \max_{x_l \in E} \quad & F_l(x_l, x_f^*) \\ \text{s.t.} \quad & x_f^* = \operatorname*{argmax}_{x_f \in E} F_f(x_l, x_f) \end{aligned}$$

Naturally the objective of the follower problem for a given x_l is

$$\max_{x_f \in E} F_f(x_l, x_f).$$

3. A Branch and Bound procedure

The defined model gives a very difficult bi-level optimization problem, where the second level itself, the location of the follower is an NP-hard problem. Indeed, we aim to solve this problem with a reliable method, the well-known Branch and Bound procedure using Interval Analysis and DC decomposition for the bound calculations.

When solving the leader's problem using a Branch and Bound method we need a reliable way to estimate lower and upper bounds for both the leader's and the follower's profit functions. These bounds for the follower's profit can be easily calculated by interval arithmetic or using DC decomposition when the leader is already located at a point, but we also need bounds when the leader can be anywhere within an interval. In the latter case we can only compute bounds by interval analysis, more specifically, by natural inclusion.

The DC bound is computed by the DC decomposition of F_f, for which let us first transform the market share of a given demand point, $m_a(x_f)$, to the following form

$$\begin{aligned} m_a(x_f) =& \frac{q_f/d_a^\lambda(x_f) + \sum_{j=k+1}^m q_j/d_a^\lambda(x_j)}{q_l/d_a^\lambda(x_l) + q_f/d_a^\lambda(x_f) + \sum_{j=1}^m q_j/d_a^\lambda(x_j)} = \\ =& \alpha_a + (1-\alpha_a)\frac{1}{1+\gamma_a d_a^\lambda(x_f)} \end{aligned}$$

where α_a and γ_a are constants depending on $a \in V$ and x_l which can be considered a constant since it is given at a certain point. A DC decomposition of this S-shaped function if $\alpha_a < 1$ is

$$m_a(x_f) = m_a^+(x_f) - m_a^-(x_f)$$

$$m_a^+(x_f) = \begin{cases} \alpha_a + (1-\alpha_a)\left[g_a(c_a) + g_a'(c_a)(d_a^\lambda(x_f) - c_a)\right] & \text{if } d_a^\lambda(x_f) \le c_a \\ \alpha_a + (1-\alpha_a) g_a(d_a^\lambda(x_f)) & \text{if } d_a^\lambda(x_f) > c_a \end{cases}$$

$$m_a^-(x_f) = \begin{cases} \alpha_a + (1-\alpha_a)\left[g_a(c_a) + g_a'(c_a)(d_a^\lambda(x_f) - c_a) - g_a(d_a^\lambda(x_f))\right] & \text{if } d_a^\lambda(x_f) \le c_a \\ 0 & \text{if } d_a^\lambda(x_f) > c_a \end{cases}$$

where

$$c_a = \left(\frac{\lambda - 1}{(1+\lambda)\gamma_a}\right)^{\frac{1}{\lambda}}, \quad g_a(t) = \frac{1}{1+\gamma_a t^\lambda}.$$

A similar decomposition can be formulated for the core of the build cost function, let us denote that by $g_a(x_f) = g_a^+(x_f) - g_a^-(x_f)$. Using these functions we have the DC decomposition of the profit function

$$F(x_f) = F^+(x_f) - F^-(x_f)$$

$$F^+(x_f) = \sum_{a\in V} \omega_a \left(m_a^+(x_f) + g_a^-(x_f)\right), \quad F^-(x_f) = \sum_{a \in V} \omega_a \left(m_a^-(x_f) + g_a^+(x_f)\right)$$

which we can use to estimate a lower bound for the follower's profit when the leader's position is given.

The computation of the leader's bounds can be done similarly, but the bounds will be valid only if the follower position is the optimal one corresponding to its profit function, or the interval of the follower contains its global optimizer. We considered the same methods we used for the follower but also running a few Branch and Bound steps for the follower problem and then using the reduced intervals for estimating the leader's bounds.

The main Branch and Bound method operates on a set of *partial solutions*. A partial solution constains a segment for leader placements, and a set of segments for the corresponding follower location which could not be eliminated so far for non-optimality. The set of partial solutions is initialized by taking every edge of the network as a leader segment and for every specific leader segment each edge as follower segments. Thus the whole location space is taken into account.

The partial solutions are stored in a binary search tree with the order defined by the upper bounds on the leader's profit function, such that the selection rule selects the element with the highest upper bound. The branching rule used for the main method splits the leader segment of the partial solution along its midpoint and leaves the follower segments unchanged. A global lower bound is maintained for the leader, this is used in the elimination step to eliminate non-optimal leader segments. A leader segment is non-optimal, i.e. cannot contain the global optimum, if its upper bound is smaller than the global lower bound.

The follower segments must be refined throughout the algorithm as well, for this we considered two methods. The first and simpler method refines the follower segments to have equal or smaller diameters than their leader segments. This way the number of follower segments might get out of hand, but by using an elimination rule their amount can be kept controlled. The elimination rule discards those segments whose upper bounds on the follower's profit function are smaller than the lower bound of the follower maximal profit associated with the given partial solution.

The second method for the refinement of the follower's segments uses a Branch and Bound method on the follower problem. The partial solution gives one segment for the leader and the list of segments for the follower. In this method the data structure, elimination, branching and selection rules are the same as in the main Branch and Bound method, although maximizing the follower's profit function. This will not converge to a specific follower position unless the leader's segment is small enough, thus we only run it for a given number of iterations. The difficulty in using this method is to find the optimal number of iterations needed to refine the follower segments enough so that the leader problem converges and yet not refining the follower segments unnecessarily.

4. Summary

We have addressed a Huff-like Stackelberg problem on a network maximizing the profit of both the leader's and the follower's chain. A Branch and Bound procedure is designed to solve the problem where bounds are computed by Interval Analysis and using DC decomposition. Our preliminary results are very promising even for medium sized networks, thus we aim to accelerate the method such that the problem can be solved for large networks as well.

References

[1] T. Drezner and Z. Drezner. Facility location in anticipation of future competition. *Location Science*, 6(1–4):155–173, 1998.

[2] H. Kucukaydin, N. Aras, and I. K. Altinel. A leader–follower game in competitive facility location. *Computers & Operations Research*, 39(2):437–448, 2012.

[3] J.L. Redondo, J. Fernandez, I. Garcia, and P.M. Ortigosa. Heuristics for the facility location and design (1|1)-centroid problem on the plane. *Computational Optimization and Applications*, 45(1):111–141, 2010.

[4] M.E. Sáiz, E.M.T. Hendrix, J. Fernández, and B. Pelegrín. On a branch-and-bound approach for a Huff-like Stackelberg location problem. *OR Spectrum*, 31:679–705, 2009.

[5] D.R. Santos-Peñate, R. Suarez-Vega, and P. Dorta-Gonzalez. The leader-follower location model. *Networks and Spatial Economics*, 7(1):45–61, 2007.

Topic Index

Author Index

ISBN: 978-84-16027-57-6
DEPOSITO. LEGAL: AL 695-2014

IN LUMINE SAPIENTIA · UNIVERSITAS ALMERIENSIS

S-AL
Supercomputación - Algoritmos

ESCUELA TÉCNICA
SUPERIOR DE
INGENIERÍA
INDUSTRIAL

DAC MÁLAGA UNIVERSIDAD

UNIVERSITAS MALACITANA