# IIIrd Workshop on Global Optimization

Organized by the Austrian and by the Hungarian Operations Research Societies

Volume of extended abstracts


Preliminary version, last update: December 5, 1995

December 10-14, 1995
Szeged, Hungary

**Program Committee:**

Reiner Horst (Trier)
Panos M. Pardalos (Gainesville)

**Organization Committee:**

Immanuel Bomze (Vienna)
András Erik Csallner (Szeged)
Tibor Csendes (Szeged)
Gabriele Danninger (Vienna)
Zoltán Kovács (Szeged)

**Address of Organizing Committee**

Tibor Csendes
József Attila University, Institute of Informatics
H-6701 Szeged, P.O. Box 652, Hungary
Phone: +36 62 454 305, Fax: +36 62 312 292
E-mail: globopt@inf.u-szeged.hu
URL: http://www.inf.u-szeged.hu/~globopt/
Ftp: ftp.jate.u-szeged.hu, in the directory /pub/math/optimization/globopt

**Sponsors**

Aranyhal Restaurant
Hungarian Scientific Research Fund OTKA
József Attila University
MOL Oilindustrial Trust
Pick Szeged Rt.
Szeged City Mayor's Office
University of Veszprém

## Preface

Many technical, environmental and economic problems have challenging optimizational aspects which require reliable and efficient solution methods. A substantial part of such problems belongs to the class of nonlinear and nonconvex optimization problems where standard optimization methods fail since local optima different from the global ones (which we aim to find) exists (global optimization).

The workshop focuses on theoretical, modelling and algorithmic issues of global optimization problems with special emphasis to their real-life applications. The workshop aims to discuss and develop further most recent results in the wide range of the many diverse approaches to global optimization problems.

After the first (1985) and the second (1990) Workshops held in Sopron, Hungary, we are glad to announce the Third Workshop on Global Optimization. Since the earlier wokshops were fully supported by IIASA, and this time IIASA could not help, the main problem seemed to be how to help those who otherwise cannot afford to participate such a meeting. Thanks to our sponsors and to the organizing OR societies, we could find sources to support those who needed it most. This rises hope that future workshops can be organized in a similar basis.

One of the main problems was for many interested people the date of the workshop. The organizers wanted to follow the tradition of the earlier workhshops with this december date. We could also keep the five years period. All comments and suggestions are wellcome for the date, place and other organizational details of the next workshop.

Having so many interesting papers submitted, over ten editors of the JOGO participating, about 20 participants of earlier workshops and a number of young researchers, we look forward to a meeting which is very likely to match or even surpass the very successful two earlier meetings.

## Daily Schedule

### Saturday, December 9

12:00 – 20:00 Registration in the hall of the Forrás Hotel

Diner and get together party from 18:00

### Sunday, December 10

09:00 – 09:30 Opening of the IIIrd Workshop on Global Optimization
09:30 – 10:00 C.S. Adjiman, I.P. Androulakis, C.D. Maranas and *Christodoulos A. Floudas*: A Global Optimization Method, $\alpha$BB
10:00 – 10:30 *K.G. Ramakrishnan*, M.G.C. Resende and P.M. Pardalos: An LP-Based Branch and Bound Algorithm for the Quadratic Assignment Problem

Coffeebreak

11:00 – 11:30 *Immanuel M. Bomze*: Evolution Towards the Maximum Clique
11:30 – 12:00 *Victor Korotkich*: On a Mechanism of Natural Formation and its Use in Global Optimization

Lunchbreak

13:30 – 14:00 *Wilfried Bollweg*: Numerical Simulation of Crystal Structures by Simulated Annealing
14:00 – 14:30 *Inmaculada Garcia* and P.M. Ortigosa: A Parallel Implementation of the Control Random Search algorithm to optimize a reconstruction from projection problem
14:30 – 15:00 *Tale Geramitchioski* and Ilios Vilos: Optimisation of the Reducing Gear Box with Minimisation its Own Weight

Coffeebreak

15:30 – 16:00 *Eligius Hendrix*: Global Optimization and Decision Support
16:00 – 16:30 *Donald Jones*, William Baritompa and Yaroslav Sergeyev: The Pareto Approach to Balancing Local and Global Search

Diner from 18:00
Hungarian folk dance show from 20:00

**Monday, December 11**

09:00 – 09:30 Jorge Moré and *Zhijun Wu*: Smoothing Transform and Continuation for Global Optimization

09:30 – 10:00 *Panos M. Pardalos*: Continuous Approaches to Discrete Optimization Problems

10:00 – 10:30 *Emilio Carrizosa* and Frank Plastria: Locating an Undesirable Facility by Generalized Cutting Planes

Coffeebreak

11:00 – 11:30 *Hisham Al-Mharmah* and James Calvin: Average Performance of Composite and Non-composite Algorithms for Global Optimization of Stochastic Functions

11:30 – 12:00 *Marco Locatelli*: On Relaxing the Hypotheses for the Application of Multi Level Single Linkage

Lunchbreak

13:30 – 14:00 *Zoltán Kovacs*, F. Friedler and L.T. Fan: Algorithmic Generation of the Mathematical Programming Model for Process Network Synthesis

14:00 – 14:30 *András Pfening* and Miklós Telek: Optimal Rejuvenation Policy for Slowly Degrading Server Software

14:30 – 15:00 *Andrew T. Phillips*, J. Ben Rosen and Ken A. Dill: CGU: A Global Optimization Algorithm for Protein Structure Prediction

Coffeebreak

15:30 – 16:00 *Olga Yu. Polyakova*: Reducing the Problem of Organization Structure Adaptation to Optimization Problem in Boolean Space

16:00 – 16:30 *Jean-Francois Pusztaszeri*, Paul E. Rensing and Thomas M. Liebling: Tracking Elementary Particles near their Primary Vertex: A Combinatorial Approach

Reception given by the Rector of the József Attila University
(Central building of the university, Dugonics square 13, 2nd floor, Aula, from 19:00)

**Tuesday, December 12**

09:00 – 09:30 *Arnold Neumaier*: NOP - a Compact Input Format for Nonlinear Optimization Problems

09:30 – 10:00 *Stephan Dallwig*, Arnold Neumaier and Hermann Schichl: GLOPT - A Program for Constrained Global Optimization

10:00 – 10:30 Alexander S. Strekalovsky and *Igor L. Vasiliev*: On Global Search in Non-Convex Optimal Control Problem

Coffeebreak

11:00 – 11:30 *Chris J. Price*: A Multistart Linkage Algorithm Using First Derivatives

11:30 – 12:00 Marco Locatelli and *Fabio Schoen*: Analysis of Threshold Accepting Global Optimization Methods

Lunchbreak

13:30 – 14:00 *Kristina Holmqvist*, Athanasios Migdalas and Panos M. Pardalos: Greedy Randomized Adaptive Search for a Location Problem with Economy of Scale

14:00 – 14:30 Balázs Imreh, *F. Friedler* and L.T. Fan: Polynomial Algorithm for Improving the Bounding Procedure in Solving Process Network Synthesis by a Branch and Bound Method

14:30 – 15:00 *Roman G. Strongin*: Global Optimization (Systematic Approach Employing Peano Mappings)

Coffeebreak

15:30 – 16:00 *Yaroslav D. Sergeyev*: An Algorithm for Minimizing Functions with Lipschitzian Derivatives

16:00 – 16:30 *Jonas Mockus*, Audris Mockus and Linas Mockus: Bayesian Heuristic Approach to Discrete and Global Optimization

Reception given by the Mayor of Szeged
(Szeged City Mayor's Office, Széchenyi square 9, 1st floor, from 19:00)

**Wednesday, December 13**

09:00 – 09:30 *Alexander S. Strekalovsky* and Ider Tsevendorj: Reverse Convex Programming. Theory and Algorithms.

09:30 – 10:00 *Oleg V. Khamisov*: To the Global Minimization of Functions with Concave Minorant

10:00 – 10:30 *Reiner Horst*: Linearly Constrained Global Optimization of Functions with Concave Minorants

Coffeebreak

11:00 – 11:30 *Michael Nast*: Subdivision of Simplices Relative to a Cutting Plane with Applications in Concave Minimization and Volume Computation

11:30 – 12:00 E.S. Mistakidis and *Panagiotu D. Panagiotopoulos*: Hemivariational Inequailies and Global Optimization. Numerical Search for the Optima.

Lunchbreak

13:30 – 14:00 *Sonja Berner*: Parallel Methods for Verified Global Optimization — Practice and Theory

14:00 – 14:30 *András Erik Csallner* and Tibor Csendes: Convergence Speed of Interval Methods for Global Optimization and the Joint Effects of Algorithmic Modifications

14:30 – 15:00 *M.N. Vrahatis*, D.G. Sotiropoulos and E.C. Triantafyllou: Global Optimization for Imprecise Problems

Coffeebreak

15:30 – 16:00 *Victor P. Gergel*: Information Models and Methods to Support Global Optimization Procedures

16:00 – 16:30 *János D. Pintér*: LGO - An Implementation of a Lipschitzian Global Optimization Procedure

Diner from 18:00

**Thursday, December 14**

09:00 – 09:30 *Tamás Rapcsák*: An Unsolved Problem of Fenchel

09:30 – 10:00 *Nguyen Van Thoai*: A Method for Solving a Utility Function Program in Multiple Objective Nonlinear Optimization

10:00 – 10:30 *Gerardo Toraldo* and Panos M. Pardalos: Quadratic Programming with Box Constraints

Coffeebreak

11:00 – 11:30 *Jens Hichert*, Armin Hoffmann and H.X. Phu: The Computation of the Essential Supremum by using Integral Methods

11:30 – 12:00 *Zelda B. Zabinsky* and Birna P. Kristinsdottir: Complexity Analysis Integrating PAS, PRS and Simulated Annealing

Lunchbreak

13:30 – 14:00 *Chris Stephens* and William Baritompa: Global Optimization Requires Global Information

14:00 – 14:30 *Tibor Csendes*: Global Optimization Methods for Process Network Synthesis

14:30 – 15:00 Walter J.H. Stortelder and *János D. Pintér*: Numerical Approximation of Elliptic Fekete Point Sets: A Global Optimization Approach

Coffeebreak

15:30 – 16:00 *James M. Calvin*: Average Convergence Rate of a Class of Adaptive Optimization Algorithms for Brownian Motion

16:00 – 16:30 *Donald Jones*: DIRECT: a Global Optimization Algorithm for Computer-Aided Engineering

Diner from 18:00

**Friday, December 15**

Check-out at the hotel till 12:00

# Average Performance of Composite and Non-Composite Algorithms for Global Optimization of Stochastic Functions

Hisham Al-Mharmah and James Calvin

An algorithm is called composite if it maintains its features when going form $n$ to $(n+1)$ observations (see Zhigljavsky (1991)). In this paper we study the composite and the non-composite algorithms for finding the global maximum of a continuous function on the unit interval. Many algorithms have been proposed to optimize functions satisfying sufficient regularity conditions such as convexity and differentiability. On the contrary, investigation of the problem in the absence of these strong assumption is still relatively limited. This work compares the average performance of different algorithms under quite general assumptions. The Wiener measure on $C([0,1])$ will be taken as the probability distribution on $\mathcal{F}$; i.e., any $f \in \mathcal{F}$ is taken to be a sample path of a Brownian motion process, and the average convergence rate is the criterion which we use to characterize the average performance of each algorithm.

We concentrate on two classes of algorithms; the random non-adaptive class where the observation sites are generated according to a certain probability distribution, and the deterministic non-adaptive class where the function is observed at a sequence of fixed locations. Non-adaptive algorithms make no use of any prior information to choose the next observation site and the generated site sequence will not be changed by changing the optimized function. The random non-adaptive algorithms are all composite, while the deterministic non-adaptive algorithms is clearly not.

For the deterministic non-adaptive class we compare the average performance of three deterministic non-adaptive algorithms: the uniform grid algorithm, denoted by $\mathrm{D}^U$, that places observations at equally spaced locations, (i.e., if the number of observations $n$ is known in advance, then they are placed at fixed locations $1/n, 2/n, ..., 1$), the one-sided deterministic sequential algorithm, denoted by $\mathrm{D}^O$, that places the observations by always subdividing the largest interval closest to the zero end point, (i.e., for $n = 6$, the observations sites will be placed at 1, 1/2, 1/4, 3/4, 1/8, and 3/8 respectively), and the two-sided deterministic sequential algorithm, denoted by $\mathrm{D}^T$, that keeps subdividing the largest subinterval closest to one of the two end points and choosing the subinterval nearest to the zero endpoint in case of ties, (i.e., for $n = 6$, the observations sites will be placed at 1, 1/2, 1/4, 3/4, 1/8, and 7/8 respectively). We show that $\mathrm{D}^O!$ has a better convergence rate than the uniform grid algorithm and we show that $\mathrm{D}^T$ has the best performance.

For random non-adaptive algorithms we discuss the distribution characteristics which improve the average convergence rate and we compare the average performance of different random algorithms. Placing the observations independently according to a Beta(2/3,2/3) density function is shown to be the optimal random non-adaptive algorithm. This distribution gives a slightly better convergence rate than choosing the sites according to the distribution of the maximizer, which is the arcsine distribution. Denoting the errors after $n$ observations chosen according to the Beta(2/3,2/3), arcsine and uniform distributions by $\Delta_n^{beta}$, $\Delta_n^{arcsine}$, and $\Delta_n^{uniform}$ respectively, we show that

$$\begin{aligned}
\sqrt{n}E[\Delta_n^{beta}] &\rightarrow \frac{1}{\pi\sqrt{2}}\mathcal{B}(2/3,2/3)^{3/2} \approx 0.662281,\\
\sqrt{n}E[\Delta_n^{arcsine}] &\rightarrow \frac{1}{\sqrt{2\pi}}\mathcal{B}(3/4,3/4) \approx 0.675978,\\
\sqrt{n}E[\Delta_n^{uniform}] &\rightarrow \frac{1}{\sqrt{2}} \approx 0.707107.
\end{aligned}$$

Also, we compare the above results with the convergence rate for the uniform grid algorithm

that places $n$ points equally spaced. Calvin (1994) showed that for this algorithm,

$$\sqrt{n}E[\Delta_n] \to \frac{1+\mathcal{C}/2}{\sqrt{2\pi}} \approx 0.5826,$$

where

$$\mathcal{C} = \int_{t=1}^{\infty} \frac{t - \lfloor t \rfloor}{t^{3/2}} dt \approx 0.9207.$$

Thus the convergence is significantly faster with deterministic uniform grid, and thus with the one-sided and the two-sided sequential deterministic algorithms. However, the difference in convergence rate between the different random non-adaptive algorithms is small compared with the improvement gained by using a deterministic non-adaptive algorithm. An important advantage of the random algorithms is that they are composite, unlike a deterministic algorithm.

**References**


[1] Calvin, J. (1994), Average performance of passive algorithms for global optimization, (to appear in Journal of Mathematical Analysis and Applications).

[2] Denisov, I.V. (1984), A random walk and a Wiener process near a maximum, Theor. Prob. Appl. **28** 821 - 824.

[3] Imhof, J.-P. (1984), Density factorizations for Brownian motion, meander and the three dimensional Bessel process, and applications,

[4] Kushner, M.J. (1964), A new method of locating the maximum point of an arbitrary multi-peak curve in the presence of noise. Journal of Basic Engineering **86**, 97-106.

[5] Revuz, D. and M. Zor (1991), Continuous Martingales and Brownian Motion, Springer-Verlag, Berlin.

[6] Ritter, K. (1990), Approximation and optimization on the Wiener space, Journal of Complexity **6** 337-364.

[7] Törn, A. and A. Žilinskas (1989), Global Optimization, Springer-Verlag, Berlin.

[8] Zhigljavsky, A. (1991), Theory of Global Random Search, Kluwer, Dordrecht.

# Parallel Methods for Verified Global Optimization — Practice and Theory

Sonja Berner

The development of satisfying methods for *global* optimization is still a problem. Given a continuously differentiable function $f : D \to \mathbb{R}$, $D \subseteq \mathbb{R}^n$, and a set $X^0 \subseteq D$, one is searching for the global minimum $f^* = \min\{f(x) : x \in X^0\}$ and for the set of all global minimum points $X^* = \{x \in X^0 : f(x) = f^*\}$.

Interval methods are able to find reliably an enclosure of the global minimum and the global minimum points. Here $X^0$ is chosen as a box, i.e. an interval vector $X^0 = [a_1, b_1] \times \cdots \times [a_n, b_n]$. The solution is found by application of the *branch and bound* principle. The *bound*-step makes use of an *inclusion function* $F$, computed by means of interval arithmetic (cf. next section), that provides an inclusion $F(Y) \supseteq \{f(y) : y \in Y\}$ of the range of $f$ for each subbox $Y$ of $X^0$.

Problems of global optimization are usually hard to solve. Thus the development of parallel methods often is a must to make them solvable at all. Here a new parallel approach to parallelization of branch and bound methods for validated optimization is presented. The efficiency of this new method is shown by measurement on a CM5 parallel computer for a variety of test problems. It is compared with other existing parallel methods that are briefly described. Further some theoretical results for the parallel method are given.

### Interval arithmetic

For a compact real interval $A = [\underline{a}, \overline{a}]$ the bounds are denoted by $\inf(A) := \underline{a}$ and $\sup(A) := \overline{a}$, $w(A) := \overline{a} - \underline{a}$ is the diameter of $A$. The absolute value of an interval is defined by $|A| := \max\{|a| : a \in A\}$. Interval vectors are termed *boxes* here. The sum, product etc. of intervals can easily by obtained (cf. [1]).

### Serial method

The serial algorithm works in the following way: The starting box $X^0$ is successively subdivided. Subboxes $X$ which reliably do not contain global minimum points are deleted by use of the criterion $\inf(F(Y)) > \tilde{f}$ where $\tilde{f}$ is an upper bound for the global minimum. All other boxes $X$ are stored as pairs $(X, \inf(F(X)))$ either in a working list $L$ or in a list $\tilde{L}$ of possible solution boxes if $w(F(X)) \leq \varepsilon$ for a chosen $\varepsilon$.

**Choice of the next box** As long as $L$ is not empty, one of the boxes of $L$ is chosen for the next subdivision. Various choices are possible:

| | |
|---|---|
| *oldest-first* strategy: | the oldest pair of $L$ is chosen (queue); |
| *best-first* strategy: | takes the pair $(Y, y)$ with $y$ minimal (sorted list); |
| *depth-first* strategy: | chooses the pair last inserted into $L$ (stack). |

One can show [2] that the best-first strategy, which was used for our algorithm, is the most efficient, since only useful boxes, i.e. boxes with $\inf(F(X)) \leq f^* + \varepsilon$ are considered for further computations. Using the other two strategies, the number of investigated boxes depends highly on the actual value of $\tilde{f}$ and is normally higher.

Even if one restricts oneself to a division of boxes by several successive bisections there are still various options:

**Choice of the bisection direction** Different strategies on how to choose the direction to bisect a given box $X$ have been examined (compare [3]). A box is always bisected in direction $i \in \{1, \ldots, n\}$ where a merit function $D_i(X)$ is maximal. Different strategies arise for different choices of $D_i(X)$:

Strategy A: $D_i(X) := w(X_i)$

Strategy B: $D_i(X) := w(F_i'(X)) \cdot w(X_i)$
Strategy C: $D_i(X) := |F_i'(X)| \cdot w(X_i)$

Numerical experiments show that for some problems, especially for very large ones, a considerable amount of time (up to 92%) can be saved by using strategy C instead of A (similar for B), thus strategy C was chosen for our algorithm.

**Number of bisections** It was tried to subdivide a box $X$ by $l \geq 1$ bisections in one step. Experiments with $l = 1, 2, 3$ showed that $l = 2$ is a good choice. For all tested problems 20% less time was needed to get the solution, with $l = 3$ the time was sometimes higher than with $l = 1$.

Further, the monotonicity test, nonconvexity check and the interval Newton method (cf. [3, 9]) were used for acceleration.

**Parallelization**

The general idea of parallelization is that each processor applies the algorithm to a box assigned to it, independent of all others. There are three important goals:

1. Make all processors know a better upper bound $\tilde{f}$ found by one processor as fast as possible but with small amount of communication.

2. No processor should become idle prematurely, therefore dynamic load balancing is needed.

3. One should always try to work on the $p$ "best" boxes, where $p$ is the number of processors, to avoid to work on boxes that are not considered in the serial case.

**Existing parallelizations** Parallel methods can be found in [4, 6, 7, 8]. A master-slave model [6] does not seem to be very efficient, the master becomes a bottleneck. Better results are obtained with a processor farm. In [8] a rather high superlinear speedup of 170 on 32 processors was reached but it turns out that this was only possible due to an inefficient serial method using the oldest-first strategy.

**A new parallel approach** The combination of master-slave model and processor farm yields a new parallelization scheme. Each processor keeps its own sorted list, they work rather independent of each other. A better value for $\tilde{f}$ is distributed by an asynchronous broadcast. The best-first strategy is used on each processor. One processor is chosen as a *centralized mediator* (cf. [10]). It is responsible for the dynamic load balancing: It does not work on boxes. Instead it waits for requests of idle processors to send them new boxes. It also keeps a limit *max* that is changed dynamically. Processors with more than *max* boxes send some of these to the centralized mediator (Figure 1).

An advantage of this parallelization is that there is less work for the centralized mediator, thus it will not become a bottleneck as long as the number of processors is not too high. Furthermore, an idle processor knows whom to send a request to, it does not need to try to get boxes from several processors.

Our parallel algorithm [2] starts with an initial phase. The starting box is partitioned, each processor gets one subbox. A local optimization on each processor followed by a synchronous load balancing often provides a good upper bound $\tilde{f}$ and a good distribution of subboxes which is quite important for problems that do not parallelize very well but need a large amount of computing time. Further improvements of the algorithm are reached by some other modifications.

**Results** The new parallel algorithm [2] described above was implemented in Pascal-XSC on a Connection Machine CM5 with 32 nodes.

For large problems with high running time mostly slightly superlinear speedup is reached, for small problems the speedup decreases (Figure 2).

A good speedup is reached especially for least squares problems with several global minimum points or local minimum points with relatively small function value. The problems from geodesy

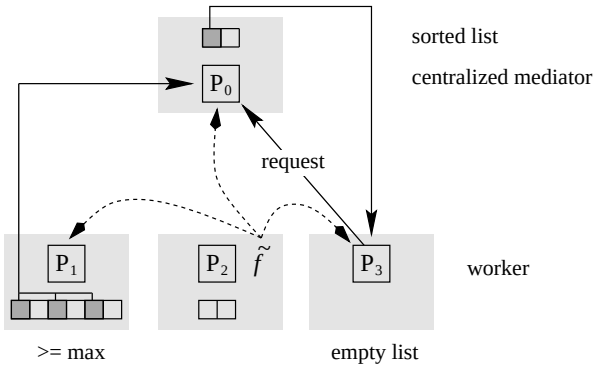



Figure 1: Used communication structure with centralized mediator.

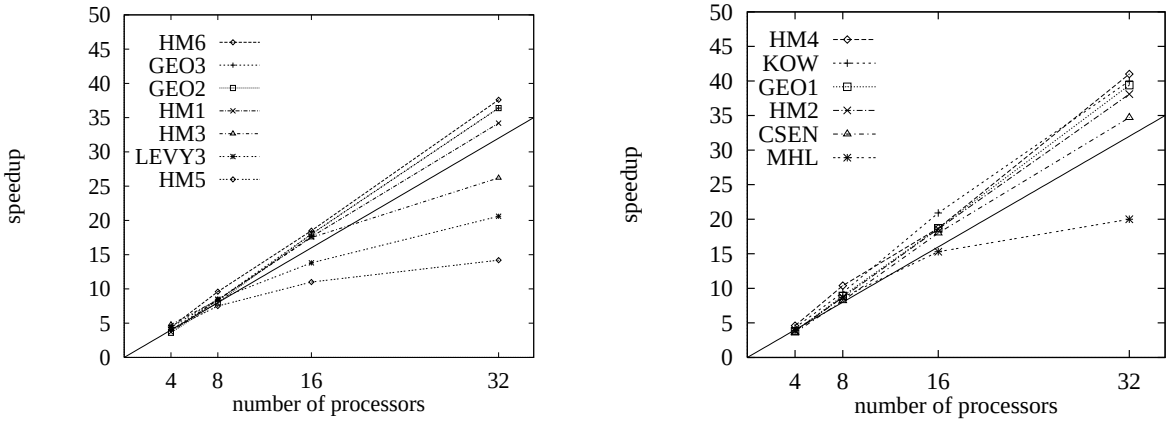



Figure 2: Sometimes no linear speedup is reached for smaller problems (left picture), but it is superlinear in most cases for larger problems (right picture).

GEO1, GEO2 and GEO3 considered here are least squares problems for example. The same is true for the Kowalik problem KOW and for the parameter estimation problems of Csendes (CSEN) and Moore, Hansen and Leclerc (MHL). For problem MHL less than linear speedup is reached, see the following for an explanation. HM1, HM2 etc. are test problems used in [6].

For some problems the improvement when using strategy C instead of A is higher in the serial than in the parallel case, thus the speedup decreases, although the parallel program is faster with strategy C. This is true e.g. for the problem MHL where the speedup decreases from 40.6 to 20.0 on 32 processors using strategy C instead of A. With this observation in mind a comparison by speedups with other parallelizations which all use strategy A for subdivision shows that the new parallelization is more efficient in almost all cases.

Some theoretical results have also been proven in [2]. One can show that the parallel method essentially provides the same enclosures for the global minimum and all global minimum points as the serial one. Further it turns out that applying the best first strategy true superlinear speedup cannot be expected.

**References**


[1] Alefeld, G., Herzberger, J., *Introduction to Interval Computations*, Academic Press, 1983.

[2] Berner, S., *Ein paralleles Verfahren zur verifizierten globalen Optimierung*, PhD thesis, submitted to Fachbereich Mathematik, Bergische Universität GH Wuppertal.

[3] Csendes, T., Ratz, D., *Subdivision direction selection in interval methods for global optimization*, to appear in SIAM Journal of Numerical Analysis.

[4] Eriksson, J., *Parallel global optimization using interval analysis*, Licentiate Thesis, University of Umeå (1991).

[5] Hansen, E., *Global Optimization Using Interval Analysis*, Marcel Dekker, Inc., 1992.

[6] Henriksen, T., Madsen, K., *Use of a depth-first strategy in parallel global optimization*, Report 92-10, Technical University of Denmark, Lyngby (1992)

[7] Leclerc, A., *Parallel interval global optimization and its implementation in C++*, Interval Computations **3**(1993), 148–163.

[8] Moore, R. E., Hansen, E., Leclerc, A., *Rigourous methods for global optimization*, in Floudas, C. A., Pardalos, P. M. (Hrsg.), Recent Advances in Global Optimization, Princeton University Press, 1992.

[9] Ratschek, H., Rokne, J., *New Computer Methods for Global Optimization*, Ellis Horwood Limited, 1988.

[10] Smith, S. L., Schnabel, R. B., *Dynamic scheduling strategies for an adaptive, asynchronous parallel global optimization algorithm*, Tech. Report CU-CS-625-92, University of Colorado (1992)

## Numerical Simulation of Crystal Structures by Simulated Annealing

Wilfried Bollweg

Three dimensional structures of crystals, proteins and nucleic acids can be characterized by energy potentials. From a numerical point of view appropriate structures appear when the observed chemical system reaches a state of minimal internal energy. Recently, theoretical simulation methods which try to find suitable structures using optimization methods have become more and more important [4, 5, 6, 10, 11]. Of particular interest is the topic of crystal structure simulation.

Crystals distinguish themselves as three dimensional constructs of their modules (groups of atoms, ions or molecules). Different modules form aequidistant rows or lattices of points, laying side by side at fixed distances in the so called "space lattice" built up by the energetic interactions of the lattice's modules. The even arrangement of the modules effects a very characteristic property of crystals, certain symmetries in their lattices. One of the most challenging questions concerning crystal structures is, whether it is possible to find these symmetries by minimizing an appropriate objective function using as little information as possible.

Most attempts at a mathematical generation of crystal structures presumed a knowledge of symmetry information on the structure. Our talk presents the results of ab initio calculations on different crystal structures based on a physical crystal model which combines short range interactions [7, 12] with long range Coulomb interactions. This method does not require any knowledge of internal symmetries.

The corresponding general nonconvex global optimization problem with an objective function $f : R^n \longrightarrow R$ can be written in the following form:

$$\begin{aligned} P'(x) = P(T(x)) \quad &: \quad \min f(T(x)) \text{ subject to} \\ &\quad a \leq x_i \leq b \text{ for } i \in \{1, \ldots, n\} \end{aligned}$$

where $T : R^n \longrightarrow R^n$ is a suitable linear transformation.

The first successful attempts to find chains and lattices of atoms by a minimization of this system were mostly based on deterministic optimization strategies [9]. The major disadvantage appeared in the fact, that starting with a fixed initial system state, always the next minimum in descend direction was found. A deterministic strategy usually is not able to leave this minimum again heading for a better one. The situation that an appropriate structure cannot be found with a deterministic method is commonly known for problems of crystal structure determination.

A better way to find suitable structures by potential energy minimization was achieved with a stochastic optimization strategy called "Simulated Annealing".

### The Optimization Strategy

Simulated Annealing is a general purpose optimization strategy proposed by Kirkpatrick et al. [8] for discrete optimization problems. The basic idea of Simulated Annealing is derived from an analogy to the annealing process of a molten solid. The method yields an efficient stochastic algorithm for determining local and global solutions of nonlinear optimization problems and extends the "Monte Carlo" method developed by Metropolis et al., to determine the equilibrium states of a set of interacting atoms at a given Temperature $T$.

In our presentation, we apply Simulated Annealing techniques to search for global and local minimum energy structures of potential energy functions associated with crystal structures. To improve the results, we introduce a variant of the algorithm which uses a modified state transition procedure derived from the physics crystal model.

**Potential Energy Functions**

The determination of an objective function for the crystal structure prediction is as yet a large unsolved problem but the literature contains several attempts for suitable expressions for special interactions [3]. One possible method is to find correct structures under the assumption that all crystallographic parameters of the energy terms and the symmetries of the structure are already known. In this field there exist some large program packages like `CFF91_CZEO` [2] which have been thoroughly tested.

However, we didn't try to follow this path. We simply tried to invest as little information as possible about the structure and didn't prescribe details on the internal structure like bond angles, torsion angles and structure symmetries. A first successful step in this direction was given by Pannetier, Bassas–Alsina and Rodruiges– Carjaval in 1990 [11]. With a relatively simple objective function using a Simulated Annealing technique they managed to find a couple of structures like $NaCl$ and $TiO_2$. Despite their success with ionic compounds, they found that their method fails when it is applied to materials like silica which contain high valence atoms like $Si$ that tend to form more directional bonds with oxygen. We recognized that it is possible to extend their method which is based on Pauling's valence [12] rule and an electrostatic potential derived from the electrostatic coulomb law calculated by a spherical cutoff [1]. In our talk we introduce the idea of Pannetier et al. [11] and present our extension of their method. With these ideas it is possible to predict a greater class of crystal structures containing also silica structures.

Though we could not find a potential energy function which incorporates all relevant bound forces and electrostatic potentials without detailed structure information, we propose a function which is suitable for determining the correct structure in a variety of ionic crystals only by using some empirical rules from electrostatics and crystal chemistry. By using a Simulated Annealing technique it is possible to generate symmetries in crystals without assuming a priori knowledge of symmetry elements.

**Examples**

Beside the found optimal structures local minima shall be mentioned as well. They are important in order to recognize energetic relations between different structures. These local minima often are comparable with natural phenomena. In reality during a too quickly performed annealing process a crystal gets stuck in such a "metastable phase" with a low but not optimal energy because the atoms of the crystal cannot arrange themselves properly. Some examples of predicted crystal structures like $CsCl$, $NaCl$, $TiO_2$, $SrTiO_3$ and $SiO_2$ will be used to complete our presentation.

**References**


[1] Allen, Tildesley – *Computer Simulation of Liquids*. Oxford Science Publications, 1987, 28 – 31.

[2] *CFF91_CZEO FORCE FIELD*. BIOSYM Technologies Inc., 9685 Scranton Road, San Diego, CA 92121–2777, USA.

[3] Burnham – *Mineral Structure Energetics and Modelling Using the Ionic Approach*. American Mineralogist, 347 – 388. Boulder, Colorado, USA, May 1993.

[4] Byrd, R. H., Derby, T., Eskow, E., Oldenkamp, K. B., Schnabel, R. – *A New Stochastic/Perturbation Method for Large-Scale Global Optimization and its Application to Water Cluster Problems*. University of Colorado at Boulder, Department of Computer Science, Boulder, Colorado, USA, May 1993.

[5] Floudas, C. A., Maranas C. D. – *Global Minimum Potential Energy Confirmations of Small Molecules.* Journal of Global Optimization, 4, 135 – 170, 1994.

[6] Journal of Global Optimization, Special Issue on Computational Chemistry, Vol. 4, 1994.

[7] Gilbert – *Soft-phere Modell for closed-shell atoms and ions.* J. Chem. Phys. 49, 2640 – 2642.

[8] Kirkpatrick, Gelatt Jr., Vecchi – *Optimization by Simulated Annealing Chains.* IBM Research Report RC 9355, 1982.

[9] Kroll, H., Maurer, H., Stöckelmann, D., Becker, W., Fulst, J., Krüsemann, R., Stutenbäumer, Th., and Zingel, A. – *Simulation of Crystal Structures by a Combined Distance Least Squares and Valence Rule Method.* Zeitschrift für Kristallographie **199**, 49-66 (1992).

[10] Mumenthaler, C., Braun, W. – *Folding of Globular Proteins by Energy Minimization and Monte Carlo Simulations with Hydrophic Surface Area Potentials.* Journal of Molecular Modeling **1**, 1995.

[11] Pannetier, Bassas-Alsina, Rodriguez-Carjaval, Calgnert – *Prediction of Crystal Structures from Crystal Chemistry Rules by Simulated Annealing.* Nature Vol. 346, July $26^{th}$ 1990.

[12] Pauling. *The Principles Determining the Structure of Complex Ionic Crystals.* Journal of the American Chemistry Society, **51**, 1926, 1010 – 1026.

## Evolution towards the maximum clique

Immanuel M. Bomze

As is well known, the problem of finding a maximum clique in a graph is NP-hard. Nevertheless, NP-hard problems may have easy instances. This paper proposes a new, global optimization algorithm which tries to exploit favourable data constellations, focussing on the continuous problem formulation: maximize a quadratic form over the standard simplex. Some general connections of the latter problem with dynamic principles of evolutionary game theory are established. As an immediate consequence, one obtains procedure which consists (a) of an iterative part similar to interior-path methods based on the so-called replicator dynamics; and (b) a routine to escape from inefficient, locally optimal solutions. For the special case of escaping from maximal cliques not of maximal size, part (b) uses efficient block pivoting techniques.

Consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $n$ vertices. A *clique* $\sigma$ is a subset of the vertex set $\mathcal{V}$ which corresponds to a complete subgraph of $\mathcal{G}$ (i.e., any pair of vertices in $\sigma$ is an edge in $\mathcal{E}$, the edge set). A clique $\sigma$ is said to be *maximal* if there is no larger clique containing $\sigma$. A (maximal) clique is said to be a *maximum* clique if it contains the most elements of all cliques. The search for such a maximum clique is an NP-hard problem, and can be formulated as a special quadratic optimization problem: let $A$

$$a_{ij} = \begin{cases} \frac{1}{2} & \text{if } \; i = j \, ; \\ 1 & \text{if } \; (i,j) \in \mathcal{E} \, , \\ 0 & \text{else.} \end{cases} \tag{1}$$

Then it turns out that the maximal (maximum) cliques correspond to the local (global) maximizers of the problem

$$x' A x \to \max ! \quad \text{subject to } \; x \in S^n \, , \tag{2}$$

where $S^n = \{x \in \mathbb{R}^n : x_i \geq 0 \;\; \text{for all } \; i \in \mathcal{V}, \; \sum_i x_i = 1\}$. The procedure proposed in this paper consists of two parts. At first, a local solution of (2) will be generated very quickly; in the second step we escape from an inefficient local maximizer in a way such that improvement in the objective is guaranteed. Both parts work also for the general problem (2) where $A$ is an arbitrary (positive) symmetric $n \times n$ matrix, not necessarily of regularized adjacency form (1).

Empirical evidence suggests that the resulting procedure indeed has some merits.

# Average Convergence Rate of a Class of Adaptive Optimization Algorithms for Brownian Motion

James M. Calvin

The purpose of this paper is to characterize the average performance of a class of adaptive global minimization algorithms under the Brownian motion model for the objective function. The object of a global minimization method is to approximate the global minimum $f^*$ of a function $f$, and sometimes also a location $t^*$ where the minimum is attained. We take $f$ to be a continuous function defined on the unit interval, and adopt the framework that the approximation is based on observation of the function value at sequentially selected points in the unit interval. That is, the searcher chooses points $t_1, t_2, \ldots \in [0,1]$ and forms an approximation $(f_n^*, t_n^*)$ to $(f^*, t^*)$ based on $\{t_i, f(t_i) : i = 1, 2, \ldots, n\}$. An adaptive algorithm chooses each new point $t_{n+1}$ as a function of $\{t_i, f(t_i) : i = 1, 2, \ldots, n\}$, while a non-adaptive algorithm chooses each point independently of the function values. We allow the possibility that the algorithm uses auxiliary randomness in the choice of observation sites.

We consider a class of adaptive algorithms that use only limited past information. For any $\delta > 0$, we construct an algorithm for which the error converges to 0 at rate $n^{-(1-\delta)}$, in contrast to the $n^{-1/2}$ rate characteristic of non-adaptive algorithms. We also identify the limiting distribution of the normalized error. The improved efficiency relative to non-adaptive algorithms comes from using information from past observations to concentrate the search in decreasing sub-regions of the minimizer.

Several methods have been used to compare the performance of different global optimization algorithms. In this paper we will be concerned with the average performance criterion. The idea is to regard $f$ as the sample path of a stochastic process and then classify algorithms based on the average error in their approximations. This method has been used to study the average performance of non-adaptive algorithms in the case where $f$ is taken to be a sample path of a Brownian motion process. Ritter (1990) showed that for any non-adaptive method, the average error decreases at rate $n^{-1/2}$ in the number of observations $n$. Calvin (1994) compared the average error for deterministic uniformly spaced observations with the expected error with random uniform sampling. Al-Mharmah and Calvin (1994) show that the optimal non-adaptive sampling density for approximating the error for Brownian motion is a Beta distribution. Calvin and Glynn (1994) extend many of these results to a more general class of diffusions.

Let $(B(t) : 0 \le t \le 1)$ be a standard Brownian motion defined on a probability space $(\Omega_1, \mathcal{F}_1, P_1)$, and let $\{U_1, U_2, \ldots\}$ be a sequence of independent, uniform $(0,1)$ random variables defined on a probability space $(\Omega_2, \mathcal{F}_2, P_2)$. Set $(\Omega, \mathcal{F}, P) = (\Omega_1 \times \Omega_2, \mathcal{F}_1 \times \mathcal{F}_2, P_1 \times P_2)$. Let $B^*$ denote the global minimum of the Brownian motion, and $t^*$ the (first) location where $B^*$ is attained. Denote by $u_n^*$ the value $U_i$ $(1 \le i \le n)$ such that $B(U_i) \le B(U_j), 1 \le j \le n$, and $U_i$ is the smallest value with this property. Let $\Delta_n$ denote the difference between the smallest value seen in the first $n$ observations and the global minimum, and let $\Gamma_n$ denote the difference between the global minimizer and the minimizer of the first $n$ observations.

We are concerned with algorithms constructed according to the following general framework. On the $(n+1)$st iteration, with probability $1/2$ we choose the observation site uniformly over the entire unit interval, and with probability $1/2$ we choose the site uniformly over a small subinterval centered at $\hat{t}_n^*$, where $\hat{t}_n^*$ is the location of the smallest observed value over those points chosen uniformly over the entire interval. The width of the interval of the local search decreases over time, so that the local searches become more concentrated as the search progresses.

Formally, let $\{\beta_i : i \ge 1\}$ be a sequence of independent Bernoulli$(1/2)$ random variables defined on $(\Omega_2, \mathcal{F}_2, P_2)$, independent of the $\{U_i\}$. Let $\{a_n\}$ be a decreasing (deterministic) sequence of positive numbers, with $a_n \downarrow 0$. The algorithms have the following form:

Set $t_1 = \hat{t}_1^* = t_1^* \leftarrow U_1$, $\hat{B}_1^* = B_1^* \leftarrow B(t_1)$;
For $k = 1, 2, \ldots, n-1$,
    If $\beta_{k+1} = 0$,
        Set $t_{k+1} \leftarrow U_{k+1}$;
        If $B(t_{k+1}) < B_k^*$, then set $t_{k+1}^* \leftarrow t_{k+1}, B_{k+1}^* \leftarrow B(t_{k+1})$;
        If $B(t_{k+1}) < \hat{B}_k^*$, then set $\hat{t}_{k+1}^* \leftarrow t_{k+1}, \hat{B}_{k+1}^* \leftarrow B(t_{k+1})$;
    Else if $\beta_{k+1} = 1$,
        Set $t_{k+1} \leftarrow t_k + a_{k+1}(U_{k+1} - \frac{1}{2})$;
        If $B(t_{k+1}) < B_k^*$, then set $t_{k+1}^* \leftarrow t_{k+1}, B_{k+1}^* \leftarrow B(t_{k+1})$.

After the last step, $B_n^*$ is our approximation to $B^*$ and $t_n^*$ is our approximation to $t^*$. We are interested in the quality of the approximations produced by the algorithm as the number of steps $n \to \infty$. Is is easy to see that this algorithm is consistent (for any choice of sequence $\{a_n\}$) in the sense that the error converges to zero $P_2-$a.s. for any Brownian path. The only information from the past maintained by the algorithm consists of $\hat{t}_n^*$, $\hat{B}_n^*$, $t_n^*$, and $B_n^*$.

To complete the description of the algorithm, it remains to determine a choice of the sequence $\{a_n\}$. If $a_n$ goes to 0 too fast relative to the speed at which $\hat{t}_n^* - t^*$ goes to 0, then the local search will tend to concentrate in subregions away from $t^*$. On the other hand, if $a_n$ goes to 0 too slowly, then the performance gain relative to uniform sampling will be small. To determine an appropriate rate, it is necessary to know the rate at which $\hat{t}_n^* - t^*$ converges to 0. We will show that $n(u_n^* - t^*)$ converges in distribution, thus giving the convergence rate needed to determine the $\{a_n\}$ sequence for our adaptive algorithm. In fact, we will derive the limiting joint distribution of $(\sqrt{n}(B_n^* - B^*), n(u_n^* - t^*))$ as $n \to \infty$.

Let $R_1$ and $R_2$ be two independent 3-dimensional Bessel processes, and define a "two-sided Bessel process" $R$ by

$$R(t) = \begin{cases} R_1(t), & t \geq 0, \\ R_2(-t), & t \leq 0. \end{cases}$$

Let $\{\tau_i\}$ be an enumeration of the points of a Poisson point process on the line with unit intensity, independent of $R$, and set $\Delta = \inf_i R(\tau_i)$, $\Gamma = \inf\{|\tau_i| : R(\tau_i) = \Delta\}$. We will establish that the joint Laplace transform of $(\Delta, \Gamma)$ is given by

$$\int_{t=0}^{\infty} \int_{y=0}^{\infty} e^{-\alpha t - \beta y} P(\Gamma \in dt, \Delta \in dy)$$
$$= 2 \int_{x=0}^{\infty} \frac{\sqrt{1+\alpha}\, e^{-\beta x/\sqrt{2}}}{\sqrt{1+\alpha}\cosh\left(x\sqrt{1+\alpha}\right) + \sqrt{\alpha}\sinh\left(x\sqrt{1+\alpha}\right)} \frac{\sinh(x)}{\cosh^2(x)} dx.$$

It follows that $n(\hat{t}_n^* - t^*)$ converges in distribution as $n \to \infty$.

We now turn our attention to the class of adaptive algorithms we set out to construct. Let $\delta$ will denote a fixed (small) positive number. We will use the sequence $a_n = [2(2-\delta)]^{-1} n^{-(1-\delta)}$ in the definition of the algorithm. Since $n(\hat{t}_n^* - t^*)$ converges in distribution, this choice of $a_n$ ensures that the distance between $\hat{t}_n^*$ (the center of the local search) and $t^*$ will be asymptotically negligible compared to the scope of the local search.

Our main result is that under the adaptive algorithm for any $0 < \delta < 1$,

$$\left(n^{1-\delta/2}\Delta_n, n^{2-\delta}\Gamma_n\right) \xrightarrow{d} (\Delta, \Gamma)$$

as $n \to \infty$. In particular, the marginal limiting distribution of the function approximation error is given by

$$P(n^{1-\delta/2}\Delta_n \leq y) \to \tanh^2\left(y\sqrt{2}\right),$$

and the corresponding normalized mean is

$$n^{1-\delta/2}E(\Delta_n) \to \frac{1}{\sqrt{2}}.$$

**References**


[1] Al-Mharmah, H. and Calvin, J. (1994). Optimal random non-adaptive algorithm for optimization of Brownian motion. To appear in *Journal of Global Optimization*.

[2] Calvin, J. (1993). Average performance of non-adaptive algorithms for global optimization of Brownian motion. *Journal of Mathematical Analysis and Applications* **191** 608-617.

[3] Calvin, J. and Glynn, P. (1994). Complexity of non-adaptive optimization algorithms for a class of diffusions. Submitted for publication.

[4] Ritter, K. (1990). Approximation and optimization on the Wiener space. *J. Complexity* **6** 337-364.

## Locating an Undesirable Facility by Generalized Cutting Planes

Emilio Carrizosa and Frank Plastria

In this paper we address the problem of locating an undesirable facility within a compact set $S$ by minimizing a strictly decreasing boundedly lower subdifferentiable function of the squared Euclidean distances to a set of fixed points.

Particular instances are the following optimization problems:

- $$\min_{x \in S} \sum_{a \in A} f_a(\|x-a\|^2),$$

  where each $f_a$ is strictly decreasing and convex.

- $$\min_{x \in S \setminus A} \sum_{a \in A} f_a(\|x-a\|),$$

  where each $f_a$ is strictly decreasing and convex, with $\lim_{t \downarrow 0} f_a(t) = 0$ for each $a \in A$.

- $$\min_{x \in S} \max_{a \in A} f_a(\|x-a\|^2),$$

  where each $f_a$ is strictly decreasing and Lipschitz.

Using (generalized) cutting planes, the resolution of this problem is reduced to solving a sequence of maxmin problems. These maxmin problems have a clear geometrical interpretation as generalized *power diagrams*, which enables to solve them sequentially by means of an on-line enumeration of the vertices of polyhedra in higher dimensions.

## Convergence Speed of Interval Methods for Global Optimization and the Joint Effects of Algorithmic Modifications[1]

András Erik Csallner and Tibor Csendes

Interval subdivision methods involve robust and reliable algorithms for global optimization. Their usefulness hinges mainly on their convergence speed.

A substantial amount of effort (e.g. [1 – 4]) has been invested to improve the efficiency of these methods. The most of the algorithms tested can be reduced to a single model algorithm. In this algorithm the place and the way of the modifications are easy to describe. The most of the studies investigate the properties of the model algorithm when using different interval selection and subdivision direction selection rules.

This presentation deals mainly with two things. The first part gives a survey on the theoretical results and shows some numerical tests to support these results.

The second part of the talk studies the modifications themselves. Some possible versions for the interval selection rule and the subdivision direction selection rule are listed. Subsequently the cross-effects of these modifications and the cut-off test are investigated by numerical testing.

The proper combinations of different modifications can influence the behaviour of the subdivision methods significantly, and thus they can show new ways how to increase the efficiency of these algorithms.

### References

[1] Hansen, E. (1992), Global Optimization Using Interval Analysis, Marcel Dekker, New York.

[2] Kearfott, R.B. and Novoa, M. (1990), INTBIS, a Portable Interval Newton/Bisection Package, ACM T. on Mathematical Software, 16, 152–157.

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

[4] Ratz, D. (1992), Automatische Ergebnisverifikation bei globalen Optimierungsproblemen, Dissertation, Universität Karlsruhe.

[1] The work was supported in part by the Grants OTKA-T016413 and OTKA-T017241

## Global Optimization Methods for Process Network Synthesis[1]

Tibor Csendes

Process network design problems (e.g. [4, 6]) form an important application field for nonlinear optimization algorithms. The problems range from basically bound constrained ones to heavily constrained ones, where the set of feasible points is of much lower dimension than the search space. The reliability and the sharpness of the results play an indispensable role. We discuss three classes of algorithms: interval techniques [3, 5], convex underestimator methods [1, 6] and traditional penalty function approaches (in a clustering framework) [2].

A substantial part of the talk is devoted to the advantages and drawbacks of interval methods. Much efficiency improvement reserves are still in the proper tuning of the interval global optimization algorithms. We discuss also the problem how to find suboptimal intervals in the search domain containing exclusively feasible points for constrained global optimization problems [5]. The suboptimal solutions with preset tolerances can be very useful in production level applications.

The numerical experiences with the three approaches will be discussed on the basis of several chemical network design problems. According to the early results, no single algorithm can be suggested for the whole class — each method has problems for which it is optimal. Better a sensible combination of the studied algorithms can serve as an efficient procedure that provides meaningful results. The talk will review these real life applications together with theoretical background and numerical efficiency figures.

### References

[1] Androulakis, I.P., C.D. Maranas and C.A. Floudas: $\alpha$BB: a global optimization method for general constrained nonconvex problems, manuscript, 1995.

[2] Csendes, T.: Nonlinear parameter estimation by global optimization — efficiency and reliability, Acta Cybernetica, 8 (1988), pp. 361–370.

[3] Csendes, T. and D. Ratz: Subdivision direction selection in interval methods for global optimization, to appear in SIAM Journal of Numerical Analysis.

[4] Kovács, Z., F. Friedler, L.T. Fan: Recycling in a separation process structure. AIChE J., 39(1993) 1087–1089.

[5] Kristinsdottir, B.P., Z.B. Zabinsky, T. Csendes, M.E. Tuttle: Methodologies for tolerance intervals. Interval Computations, 3(1993) No. 3, 133–147.

[6] Quesada, I. and I.E. Grossmann: Global optimization algorithm for heat exchanger networks, Industrial and Engineering Chemistry Research, 32 (1993), pp. 487–499.

[1] The work was supported in part by the Grants OTKA-T016413 and OTKA-T017241

## GLOPT – A Program for Constrained Global Optimization

Stefan Dallwig, Arnold Neumaier and Hermann Schichl

GLOPT finds the global minimizer of a block-separable objective function subject to bound constraints and block-separable constraints of the form

$$\sum_k f_k(x_{J_k}) \in [q], \quad \text{or}$$
$$\sum_k f_k(x_{J_k}) + b = x_j.$$

where $x_{J_k}$ is a subvector indexed by a one- or two-dimensional index list $J_k$, and $[q]$ is a possibly unbounded interval.

GLOPT is written in Fortran 77; since it does not use directed rounding, its reliability is that expected of other numerically stable floating point calculation; i.e., because of rounding errors, we find a nearly globally optimal point that is near a true local minimizer. Unless there are several nearly global local minimizers, we thus find a good approximation to the global minimizer.

GLOPT uses a branch and bound technique to split to problem recursively into subproblems which are either eliminated or reduced in their size. This is done by an extensive use of the block separable structure of the optimization problem.

For processing with GLOPT, constrained optimization problems are coded in the input format NOP [4] that explicitly displays the internal structure of the problem with very little overhead.

As in a method by Jansson & Knüppel [2] for bound constrained global optimization, no derivative information is used in GLOPT for problems where the feasible domain has a nonempty (and not too tiny) interior. For problems where the feasible domain has empty interior, local optimization techniques are used to locate feasible points.

At the time of writing (Sept. 15th), the algorithm solves successfully most problems from Chapter 2 of the problem collection by Floudas & Pardalos [1], and all problems considered by Jansson & Knüppel [2].

Our present implementation does not yet implement some features supported by the NOP format (integer or threshold constraints, multiobjective optimization), but this might change soon.

### References

[1] C. A. Floudas and P.M. Pardalos, A collection of test problems for constrained global optimization algorithms, Lecture Notes Comp. Sci. 455, Springer, Berlin 1990.

[2] C. Jansson and O. Knüppel, A global minimization method: The multi-dimensional case, Preprint, 1992.

[3] A. Neumaier, NOP – a compact input format for nonlinear optimization problems, talk at this conference.

## A Global Optimization Method, $\alpha$BB

C.S. Adjiman, I.P. Androulakis C.D. Maranas and C.A. Floudas

Optimization problems abound in diverse areas of chemical engineering such as process design and control, operations planing, product design. Except for a few special instances (i.e., convex problems), optimization problems are characterized by the presence of multiple local minima whose number increases rapidly with the size of the problem. Locating the global minimum point despite the presence off a plethora of local minima is of particular importance. This is because the global minimum point typically describes a unique state of the system being optimized.

In this paper, the deterministic global optimization algorithm, $\alpha BB$, ($\alpha$-based **B**ranch and **B**ound) is presented. This algorithm offers mathematical guarantees for convergence to a point arbitrarily close to the global minimum. The key idea is the construction of a converging sequence of upper and lower bounds on the global minimum through the convex relaxation of the original problem. A convex relaxation of the original nonconvex problem is obtained by (i) replacing all nonconvex term of special structure i.e. bilinear, univariate concave) with customized tight convex lower bounding functions and (ii) by utilizing the $\alpha$ parameter as defined by Maranas and Floudas (1994), to underestimate nonconvex terms of generic structure. In most cases, the calculation of the exact value of the parameter $\alpha$ in order to construct valid convex underestimating is a challenging task. A novel approach which generates a valid bound on $\alpha$ is proposed. It has the advantage being computationally tractable while preserving the global optimality guarantees of the algorithm. This method relies on the generation of the interval Hessian matrix of the function being investigated, or an enclosure of that matrix. The extremal eigenvalues of an appropriate subset of its vertex matrices are then calculated, based on a Kharitonov-like theorem and thus yielding a guaranteed bound on the value of $\alpha$.

The implementation of the $\alpha$BB algorithm includes a user-friendly parser, which facilitates problem input and provides flexibility in he selection of a suitable underestimating strategy. In addition, the package features both automatic differentiation and interval arithmetic capabilities.

The proposed approach is illustrated with a large number of process systems examples involving design problems of various sizes, distillation sequencing and reactor design.

## Polynomial Algorithm for Improving the Bounding Procedure in Solving Process Network Synthesis by a Branch and Bound Method

B. Imreh, F. Friedler and L.T. Fan

The MIP model of process network synthesis (PNS) contains a large number of binary variables associated with operating units. This renders the model difficult to solve by any available method without exploiting the specific features of process structures. The branch and bound method has various advantages in solving MIP problems over other methods. Nevertheless, the general branch and bound method is far from efficient in solving the MIP model of PNS because it tends to give rise to a large number of partial problems, each of which contains unnecessarily large number of variables. Combinatorial analysis of the MIP models of PNS and that of feasible process structures have yielded mathematical tools for exploiting the unique characteristics of PNS (Friedler et al., 1995). These tools accelerate the branch and bound search for the optimal solution by minimizing the number of partial problems to be solved and by reducing the size of an individual partial problem.

The present work is concerned with the bounding procedure for further acceleration of the branch and bound search under the assumption that the cost function of an operating unit includes a positive fixed charge in addition to a linear or nonlinear variable charge cost. The conventional bounding procedures, e.g., LP relaxation of a MILP problem, do not consider the fixed charges of the operating units not included in a partial problem; however, if the partial problem is not a solution of the PNS problem, it must be extended with some operating units. Thus, the conventionally generated lower bound can be increased with the sum of the fixed charges of these operating units. This sum should be the minimum among the sums of the fixed charges of such sets of operating units that can extend the partially defined structure to a feasible structure of the PNS problem. This minimal sum can only be generated by available algorithms in exponential time. The present work introduces a combinatorial algorithm that gives a sharp lower estimation for this minimal sum in polynomial time.

### Reference

1. Friedler, F., J. B. Varga, E. Fehér, and L. T. Fan, Combinatorially Accelerated Branch-and-Bound Method for Solving the MIP Model of Process Network Synthesis, presented at the International Conference on State of the Art in Global Optimization: Computational Methods and Applications, Princeton University, Princeton, NJ, U.S.A., April 28-30, 1995; also to be published in Nonconvex Optimization and its Applications, Kluwer Academic Publishers, Norwell, MA, U.S.A. (in press).

# A Parallel Implementation of the Controlled Random Search Algorithm to Optimize an Algorithm for Reconstruction from Projections[1]

Inmaculada Garcia, P.M. Ortigosa, L.G. Casado, G.T. Herman and S. Matej

A parallel implementation of a global optimization algorithm is described in this paper. The algorithm is based on a probabilistic random search method. Computational results are illustrated through application of the algorithm to a time consuming problem which arises from the field of image reconstruction from projections.

In areas such as tomography and electron microscopy the reconstruction from projections problem is solved by several kinds of methods. One of the problems in these fields is to determine which method is the best approach. Also the question for a particular method is: what are the values for the free parameters that optimize the quality of the reconstructed images?. An answer can be obtained by the application of a global optimization algorithm on a function (Figure Of Merit) that estimates the quality of the reconstructed image compared to the original image. The reconstruction method used in this work is a particular implementation of the well known ART (Algebraic Reconstruction Techniques) algorithm [1] and the objective function to be minimized is the distance between the original and the reconstructed images. Briefly the problem can be posed as: Given a set of projection data $(g_1, \ldots, g_m)$, which have been obtained from an image $(F_1, \ldots, F_p)$, and the reconstructed image $(X_0^r, \ldots, X_p^r)$, obtained by the application of ART, find the minimum value of the function:

$$\Phi(\lambda_1, \ldots, \lambda_r) = \sqrt{\frac{\sum_{i=1}^{p}(F_i - X_i^r(\lambda_1, \ldots, \lambda_r))^2}{n}}, \tag{1}$$

where vector $\lambda = \lambda_1, \ldots, \lambda_r$ is a set of free parameters used in the reconstruction algorithm. In our application we have chosen a fixed relaxation parameter of the ART method as the variable for the optimization problem. ART is an iterative algorithm that at the $j$th iteration updates the vector $X^j$ by applying $m$ times the following equation:

$$X^{j,k+1} = X^{j,k} + \lambda^j \frac{[g_{k+1} - \langle h_{k+1}, X^{j,k}\rangle]}{\|h_{k+1}\|^2} h_{k+1}; \quad k = 0, 1, \ldots, m-1 \tag{2}$$

The convergence and performance of this general algorithm have been studied by Herman et al. (see for example [2, 3] and their references). They point out that ART using blobs may be the most efficacious algorithm for reconstruction from projection and also that the value of the relaxation parameters is likely to have a large influence on the quality of the reconstructions.

In the selection of a global optimization method for our particular application we have taken into account the fact that the computational cost of the function to be evaluated is enormous (15-20 minutes on a Sparc 10, 50Mhz). As a consequence, a parallel algorithm for global optimization seems to be the most appropriate. In this work we propose a parallel algorithm which is based on the Controlled Random Search (CRS) algorithm of Price [4, 5]. Some parallel approaches have been proposed by McKeown [6], Sutti [7], Ducksbury [8], Price [9] and Woodhams and Price [10] using various kind of parallel computers and strategies. Our proposal, described in the next section, makes little modifications to the original sequential version of CRS. These modifications are aimed at estimating the objective function on several processors simultaneously. Nevertheless, the general strategy used in CRS remains in our parallel version (PCRS).

[1]This work was supported by the Ministry of Education of Spain (DGICYT PR94-357), by the Consejeria de Educación de la Junta de Andalucia (07/FSC/MDM), and by the National Institutes of Health (HL28438 and CA54356).

In Section 3 we show some results obtained from the application of PCRS to a classical set of test functions [12]. Results of the PCRS algorithm for image reconstruction from projections are shown in Section 4.

**The PCRS algorithm and its parallel implementation**

The Parallel Controlled Random Search (PCRS) algorithm is based on a master-worker communication model. In this strategy the master processor executes the PCRS algorithm and a worker processor only evaluates the objective function at the trial points supplied by the master processor. After every evaluation it sends the result back to the master. PCRS starts with the evaluation at $N$ trial points chosen at random from the search domain $V$ over which the objective function $\Phi : \mathbf{R}^n \rightarrow \mathbf{R}$ is to be optimized. In our description the coordinates of a trial point $j$ are stored in a vector $A_1^j, \ldots, A_n^j$ and $A_0^j = \Phi(A_1^j, \ldots, A_n^j)$. The objective function at the $N$ trial points is computed in parallel by the worker processors. Two procedures, called SEND and RECEIVE, are used by the master and worker processors to exchange a real vector $A_0, \ldots, A_n$ (see the algorithmic description of PCRS in the Appendix at the end of the paper). The master processor chooses randomly $n+1$ points $(R^0, \ldots, R^n)$ from the set $A^0, \ldots, A^{N-1}$ and determines their centroid $\overline{G}$ and a trial point $\overline{P}$. If $\overline{P}$ is in the domain $V$, then $\overline{P}$ is sent to one of the idle worker processors, otherwise a new random choice of $(R^0, \ldots, R^n)$ is made. In order to get the best efficiency of the parallel implementation, this procedure is repeated $NP$ times ($NP$ is the number of worker processors). As a consequence every processor in the parallel system is doing useful work and the workload of the parallel system is balanced. At this moment, a procedure is executed by the master processor iteratively until a stopping criterion is satisfied. During an iterative step the greatest value $A_0^m$ in the set $A_0^0, \ldots, A_0^{N-1}$ is determined. If a value $B_0$ smaller than $A_0^m$ is received from a worker processor, then $A^m$ is replaced by this new trial point $B(B_0, \ldots, B_n)$. The stopping criterion is based on the maximum distance between any two points in the set $A^0, \ldots, A^{N-1}$ and on the maximum difference of the objective function in the set $A_0^0, \ldots, A_0^{N-1}$.

**Results on a set of test functions**

Some results are given in Tables 1 and 2 for the problems of Goldstein/Price, Hartman and Shekel. For each problem the same series of five random sequences were used. Data in Table 1 are the maximum number of evaluations over the series (the sum totals for all $NP$ worker processors). The index of success for finding the global minimum was 100% for every test functions. The percentage of increasing (or decreasing) in the number of function evaluations using $NP$ processors, relative to the sequential case, is represented in parentheses. The results suggest that the number of function evaluations does not increase with the number of worker processors; it seems to depend on the specific function within a range of $\pm 20\%$ as compared to the sequential version.

Table 1: Maximum number of function evaluations for a set of test functions versus the number of worker processors ($NP$)

| NP | Gold/Price | | Hartman-6 | | Hartman-3 | | Shekel-5 | | Shekel-7 | | Shekel-10 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 338 | | 2904 | | 852 | | 1419 | | 1270 | | 1190 | |
| 2 | 384 | (+13.6) | 2845 | (-2.0) | 853 | (+0.1) | 1215 | (-14.4) | 1242 | (-2.2) | 1258 | (+5.7) |
| 4 | 388 | (+14.8) | 2781 | (-4.2) | 865 | (+1.5) | 1218 | (-12.8) | 1240 | (-2.4) | 1250 | (+5.0) |
| 8 | 395 | (+16.9) | 2869 | (-1.2) | 943 | (+10.7) | 1254 | (-11.6) | 1236 | (-2.7) | 1245 | (+4.6) |
| 16 | 376 | (+11.2) | 2784 | (-4.1) | 983 | (+15.3) | 1157 | (-18.5) | 1155 | (-9.1) | 1254 | (+5.4) |

The performance of a parallel algorithm is usually measured by the speed-up. Speed-up is

defined as the ratio $\frac{t_1}{t_{NP}}$, where $t_1$ and $t_{NP}$ are the times spent by the algorithm using one and $NP$ processors, respectively. It is clear that $t_1$ and $t_{NP}$ depend on the number of evaluations in a particular execution of the algorithm and $t_{NP}$ is also a function of the delay introduced in the parallel system because of the interprocessor communications. Let $t_f$ and $t_c$ be the CPU times for evaluating once the objective function and for the interprocessor communication delay, respectively. Let $n_1$ and $n_{NP}$ be the number of evaluations for a uniprocessor system and for a multiprocessor system with $NP$ worker processors. Then $t_1 = n_1 \times t_f$, $t_{NP} = n_{NP} \times (t_f + t_c)$, and

$$speed-up = \frac{t_1}{t_{NP}} = \frac{n_1}{n_{NP}} \times \frac{1}{1 + \frac{t_c}{t_f}} \tag{3}$$

There are two terms in the speed-up equation; the ratio $\frac{n_1}{n_{NP}}$ and that due to the delay for communicating data in the parallel system. Table 2 provides the ratio $NP \times \frac{n_1}{n_{NP}}$ for the set of test functions of Table 1. From Table 2 it can be concluded that, when $\frac{t_c}{t_f} < 0.1$, almost a linear speed-up and sometimes a super speed-up can be achieved.

Table 2: Values for speed-up considering that $\frac{t_c}{t_f} << 1$

| $NP$ | Goldstein/Price | Hartman-6 | Hartman-3 | Shekel-5 | Shekel-7 | Shekel-10 |
|---|---|---|---|---|---|---|
| 2 | 1.8 | 2.0 | 2.0 | 3.4 | 2.1 | 1.9 |
| 4 | 3.5 | 4.2 | 3.9 | 4.6 | 4.1 | 3.8 |
| 8 | 6.9 | 8.1 | 7.2 | 9.1 | 8.2 | 7.6 |
| 16 | 14.4 | 16.7 | 13.9 | 19.7 | 17.6 | 15.2 |

**PCRS for reconstruction from projections**

In image reconstruction from projections the input data are noisy. Consequently, the result of optimization based on only one data set may not be reliable. For this reason we have used four data sets. The PCRS optimization algorithm has been applied to the image reconstruction problem for the unidimensional case when the relaxation parameter is the free variable and the objective function is the average root mean squared error over the reconstructed images. Figure 1 presents the values of the objective function for the four sets of projection data and for their average. Clearly, the noise introduced in the projections data does not greatly modify the solution (the maximum difference of the objective function is $5 \times 10^{-5}$ which is 0.1% of the average value.

This work, presenting results for unidimensional optimization, can be considered to be our first step towards multidimensional optimization for image reconstruction from projections. We are know testing PCRS and other optimization algorithms in multidimensional spaces ($n = 4, n = 7$) using several reconstruction algorithms.

**References**


[1] G.T. Herman. *Image Reconstruction from Projections: The fundamentals of computerized Tomography.* Academic Press, New York, 1980.

[2] H.T. Herman and L.B. Meyer. Algebraic reconstruction techniques can be made computationally efficient. *IEEE trans. Med. Imag.*, MI-12(3):600–609, 1993.

[3] S. Matej, G.T. Herman, T.K. Narayan, S.S. Furuie, R.M. Lewit and P.E. Kinahan. Evaluation of task-oriented performance of several fully 3D PET reconstruction algorithms. *Phys. Med. Biol.*, 39:355–367, 1994.

[4] W.L. Price. A controlled random search procedure for global optimization. In L.C.W Dixon and G.P. Szegö, editor, *Towards Global Optimization 2*, pages 71–84. North Holland, Amsterdam, 1978.

[5] W.L. Price. Global optimization algorithms by controlled random search. *Journal of Optimisation Theory and Applications*, (40):333–348, 1983.

[6] J.J. McKeown. Aspect of parallel computations in numerical optimization. In F. Archetti and M. Cugiani, editor, *Numerical techniques for stochastic systems*, pages 297–327. 1980.

[7] C. Sutti. Local and global optimization by parallel algorithms for MIMD systems. *Annals of Operating Research*, (1):151–164, 1984.

[8] P.G. Duckbury. *Parallel array processing.* Ellis Horward, Chichester, 1986.

[9] W.L. Price. Global optimization algorithms for a CAD workstation. *Journal of Optimisation Theory and Applications*, (55):133–146, 1987.

[10] F.W.D. Woodhams, W.L. Price. Optimizing accelerator for CAD workstation. *IEE Proceedings Part E*, 135(4):214–221, 1988.

[11] I. García, G.T. Herman. Global optimization by parallel constrained biased random search. In C.A. Floudas and P.M. Pardalos, editor, *State of art in global optimization: Computational Methods and Applications.* In press.

[12] A. Torn and A. Zilinskas. *Global Optimization. Lecture Notes in Computer Science 350.* Springer-Verlag, Berlin, 1989.

## Appendix: PCRS algorithm

Begin **PCRS**$(N, n, V, NP)$
  Choose $N$ points at random over $V \rightarrow A^0, \ldots A^{N-1}$.
  do $j = 0 : min(N-1, NP-1)$
    SEND $A^j$ to $PE_j$ ( $PE_j$ compute $A_0 = \Phi(A_1, \ldots, A_n)$)
  $k = 0$
  if $N > NP$
    do $j = NP-1 : N-1$
      RECEIVE $(A, IDP)$
      $A \rightarrow A^k$
      SEND $A^j$ to $PE_{IDP}$
      $k = k+1$
  do $j = 0 : min(N-1, NP-1)$
    RECEIVE $(A, IDP)$
    $A \rightarrow A^k$
    $k = k+1$
  do $j = 0 : NP-1$
    Choose randomly $n+1$ points $R^0, \ldots, R^n$ from the set $A^0, \ldots, A^{N-1}$.
    Determine the centroid $\overline{G}$ for $R^1, \ldots, R^n$
    $\overline{P} = 2 \times \overline{G} - \overline{R}_0$
    if $\overline{P} \in V$ SEND $\overline{P}$ to $PE_j$
    else $j = j-1$
  $flag = 0$
  while until convergency
    Determine the stored point $m$ which has the greatest function value $A_0^m$
    Choose randomly $n+1$ points $R^0, \ldots, R^n$ from the set $A^0, \ldots, A^{N-1}$.
    if $flag = 0$
      Determine the centroid $\overline{G}$ for $R^1, \ldots, R^n$
      $\overline{P} = 2 \times \overline{G} - \overline{R}^0$

else $\overline{P} = \frac{\overline{G}+\overline{R}^0}{2}$
$flag = 0$
if $\overline{P} \in V$
RECEIVE $(B, IDP)$
SEND $\overline{P}$ to $PE_{IDP}$
if $B_0 < A_0^m$ then $B \rightarrow A^m$ and compute success rate $(succ)$
else if $(succ < 50\%)$ then $flag = 1$
End while
End **PCRS**

# Optimization of the Reducing Gear Box with Minimization its Own Weight

Tale Geramitchioski and Ilios Vilos

One access for thick sheet metal optimisation of the reducing gear box construction with the principal of gear box mass minimization is given.

The reducing gear box has a multiplex significance in reduce gears functions. All loads concentrated in bearings are accepting and transiting to the platform by gear box, it ensure respective mutually position of the reduce gear parts, it ensures physical separating of the atmosphere space and the inner gear box space where exists the driving gears, shafts, bearings where the inner space is using as a lubricant container too.

The gearing box construction has two parts - upper and lower. The gear box must have appropriate ultimate strength and stiffness, but also the simplicity and easy for construction and production because of the cost price reducing

Reducing gear box are producing most often by cast iron or by welding sheets metals. The first proceeding is priority for the complex geometrical constructions, and the other is most acceptable for the cases when the gear box weight minimization is the priority condition.

## The Function of Minimization

The first step to solving the problems is to compose the function of all influence construction weight parameters. As a variables in this problem we are using the gear box fundament thick, the thick of the length and transversal walls where the resultant bearing loads are accepts, and finally, the strengthening ribs thick using under the bearings, on the gear box lower part.

In finite shape, the function of the purpose which valye minimizing is given with this formulation.

$$\begin{aligned} FC &= (L*H1*(B-B_v)+2*B_v*H)+(B_v*(L-2))*\delta_1+(2*L*H2-\Pi/2*(D1^2+ \\ &\quad D2^2)-2*B_v)*\delta_2-(2*L*H1+\Pi/2*D1^2+D2^2)-4*H1+D2+D1)*\delta_3+ \\ &\quad (4*L-6)*\delta_1*\delta_3 \end{aligned}$$

The parameters using in upper formulation are specified at the end of the paper, and on fig 2. we can see the graphical presentation of the gear box cross section with using variables.

## The Function of Limitations

The first criterion - necessary warping of transversal and length walls

$$\frac{H1-D1/2-2*\delta_1}{i} \leq 25 \tag{1}$$

- cross section radius of inertia with necessary satisfied the buckling strength

$$\frac{R1}{25*\delta_1^2+\delta_2*\delta_3} \leq \sigma_D \tag{2}$$

for the wall under the biggest bearing hole with diameter and load .

$$\frac{H1-D2/2-2*\delta_1}{i} \leq 25 \tag{3}$$

with satisfied the buckling strength

$$\frac{R2}{25*\delta_1^2+\delta_2*\delta_3} \leq \sigma_D \tag{4}$$

Second criterion - limitation of surface pressure on the bearing holes

$$\frac{R1}{\Pi * D1 * (\delta_2 + \delta 3)} \leq p_d \tag{5}$$

$$\frac{R2}{\Pi * D1 * (\delta_2 + \delta 3)} \leq p_d \tag{6}$$

The third criterion - limitation of walls strength

$$\frac{MC}{I_{x-x_c}} * y_{max-c} \leq \sigma_{cd} \tag{7}$$

$$\frac{M2}{I_{x-x_2}} * y_{max-2} \leq \sigma_{cd} \tag{8}$$

The fourth criterion - constructive limitations

$$2 * \delta_2 + 2 * \delta 3 + B_v \leq B5 < \delta_1 > 50; 5 < \delta_1 > 50; 5 < \delta_1 > 50 \tag{9}$$

**Mathematical Solving the Problem with One Example**

Complex method by M.J.Box [3] was used for solving the existing nonlinear problem. That method is a kind of modification of the simplex Nelder-Mead method [4]. One example of the gear box optimization using the method of minimization it's own weight, and input parameters of the solving problem is given.

OPTIMIZATION OF THE REDUCING GEAR BOX CONSTRUCTION INPUT PARAMETERS FOR SOLVING

The total high of the gearing box H(mm)=212. The total length of the gearing box L(mm)=315. Axle base A(mm)=100. High of the gearing box lower part H1(mm)=112. High of the gearing box upper part H2(mm)=100. The first bearing hole diameter D1(mm)=32. The second bearing hole diameter D2(mm)=40. The first bearing reaction R1(N)=1500. The second bearing reaction R2(N)=2000. Diameter of the smaller gear Dz1(mm)=60. Diameter of the bigger gear Dz2(mm)=110. The critical normal strength of the material SIGD($N/mm^2$)=60. Number of the variable's in function N=3. Number of the limiting function's M=9. Constructive limit's number : 3.

$$5 \leq X(1) \leq 50, 5 \leq X(2) \leq 50, 5 \leq X(3) \leq 50$$

THE MINIMIZATION FUNCTION:

$$FC = K1 + K2 * X(1) + K3 * X(2) + K4 * X(3) + K5 * X(1) * X(3)$$

With the constants:

$$K1 = L * H1 * (B - B_v) + 2 * B_v * H$$

$$K2 = (B_v * (L - 2))$$

$$K3 = 2 * L * H2 - \Pi/2 * (D1^2 + D2^2) - 2 * B_v$$

$$K4 = 2 * L * H1 + \Pi/2 * D1^2 + D2^2) - 4 * H1 + D2 + D1$$

$$K5 = 4 * L - 6$$

THE LIMITS The first initial solutions: X(1)=20 X(2)=20 X(3)=20

$$(H1 - D1/2 - 2 * X(1))/i \leq 25$$

$$R1/(25 * X(1)^2 + X(2) * X(3)) \leq \sigma_D$$

$$(H1 - D2/2 - 2 * X(1))/i \leq 25$$

$$R2/(25 * X(1)^2 + X(2) * X(3) \leq \sigma_D$$

$$R1/(\Pi * D1 * (X(2) + X(3))) \leq p_d$$

$$R2/(\Pi * D1 * (X(2) + X(3))) \leq p_d$$

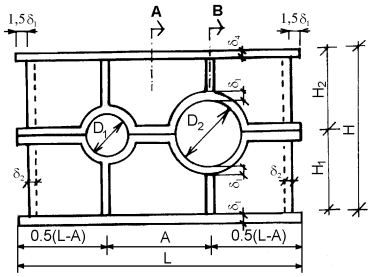
$$MC/(I_{x-x_c}) * y_{max-c} \leq \sigma_{cd}$$

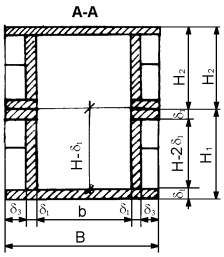
$$M2/(I_{x-x_2}) * y_{max-2} \leq \sigma_{cd}$$

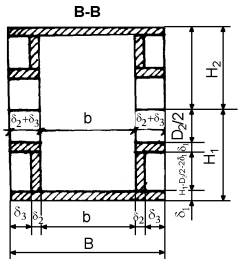
RESULTS

THE OPTIMAL VALUES OF THE VARIABLES:

X1(mm)=29.24120 X2(mm)=24.09092 X3(mm)=23.10294 THE MINIMUM OF THE FUNCTION (MINIMUM VOLUMEN mm ) IS: 3503045 THE MINIMUM WEIGHT OF THE GEARING BOX SHEET METAL T(kg) IS: 269.7643 NUMBER OF THE ITERATIONS TO THE FINITE SOLUTION ARE: 7




## References


[1] Anfimov,M.I.: Reduktory,konstrukcii I rascjot, Masinostroenije, Moskva, 1965

[2] Acerkan,N.S.: Detali masin, spravocnik, Sudostroenije, Leningrad, 1970

[3] Box,M.,J.: A new method of constrained optimisation and a comparison with other method's, The Compt. Journal 8, 45-52, 1965

[4] Nelder, J., Mead, R.: A Simplex Method For Function Minimization, The Comput. Journal 7, 308-313, 1965

[5] Bunday, B.D.: Basic Optimisation Methods, School of Mathematical Sciences, University of Bradford, UK, 1988

# Information Models and Methods to Support Global Optimization Procedures

Victor P. Gergel

In our lecture we intend to discuss some problems of collecting, storing and processing search information obtained in the course of global optimization. Within the framework of our discussion we shall propose a number of new models for the presentation of this information. These models make it possible to apply effective methods to process search data. As we expect, such results can be used for obtaining numerical solutions to the problems of global optimization, pattern recognition and etc.

1. Consider the N-dimensional problem

$$\min f(y), y \in D(1) \tag{1}$$

where the search domain

$$D = y \in R^n : a_i \leq y_i \leq b_i, 1 \leq i \leq N,$$

$\mathbb{R}^n$ is the N-dimensional Euclidean space and the objective function f(y) to be minimized may be multiextremal. We suppose also that evaluating values of $f(y)$ at any point $y \in D$ may require extensive computing efforts.

Let us make two assumptions about the nature of how a global optimization method selects iteration points to solve the problem (1).

As we suppose, for many well-known global search techniques a procedure for selecting a new iteration point $y^{k+1}$ after making $k$, $k > 1$, search iterations at the points $y^1, y^2, \ldots, y^k$ can be described as the mapping (the decisive rule)

$$y^{k+1} = G_k(y^1, y^2, \ldots, y^k; z^1, z^2, \ldots, z^k), \tag{2}$$

where $z^i, 1 \leq i \leq k$, are the values of the function $f(y)$ at the points $y^i$, $1 \leq i \leq k$. That is, when a new iteration point is selected the method uses function values calculated in previous iterations. It is important to note that reducing the number of iteration points taken into account in (2) may deteriorate the convergence properties of the method.

We suppose also, that when the method selects $y^{k+1}$ from (2) it estimates possible function values at points from $D$. To do that, for instance, at some point $y \in D$ the method takes into account mainly function values $z^i$, $1 \leq i \leq k$, calculated at the nearest points, i. e. at the points

$$y^i : 1 \leq i \leq k, \rho(y^i, y) \leq \delta,$$

where $\rho$ is the metric in the $N$-dimensional Euclidean space. As a rule, the value of $\delta$ is unknown which requires to sort points $y^i, 1 \leq i \leq k$ in accordance with the distance to the point $y$. The complexity of such operation can be evaluated as $O(Nk \log k)$. Let us suppose that the number of neighbourhoods estimated in the course of selecting a new iteration point is directly proportional to the number of previous iterations and the number of iteration points are inversely proportional to a required accuracy $\epsilon$ of the global minimum estimate, viz. we suppose that $k = (1/\epsilon)^N$. In this case the total complexity of making optimization iteration can be evaluated as

$$T_1 = O(Nk^2 \log k) = O(N^2 (1/\epsilon)^{2N} log(1/\epsilon)). \tag{3}$$

This dependence demonstrates that even for the small dimensions (e.g. $N = 4$) and for the rough accuracy (e.g. $\epsilon = 0.1$), executing the optimization iterations requires hard computations.

As a result we can conclude that the problem of search information processing is one of the main problems of global search implementation.

2. Search information can be very useful also for solving optimization problems which can be transformed in the course of global search. For instance, solving a multicriteria problem can be performed as solving a sequence of scalar multiextremal problems [1]

$$\min F(y), F(y) = \max_{1 \le j \le s} \lambda_j f_j(y), y \in D, \tag{4}$$

with various values of the weight coefficients $\lambda_j, 1 \le j \le s$, of the partial criteria $f_j, 1 \le j \le s$. If in this case we store values $z_j^i, 1 \le j \le s, 1 \le i \le k$, of the partial criteria $f_j, 1 \le j \le s$, calculated at previous iteration points $y_i, 1 \le i \le k$, then for any $\lambda$ we can calculate values $Z_i, 1 \le i \le k$, of the aggregated criterion $F(y)$ at the same points $y_i, 1 \le i \le k$, by the expression

$$Z^i = \max_{1 \le j \le s} \lambda_j z_j^i, 1 \le i \le k$$

without repeating time-consuming computations of the values $f_j(y^i)$. As a result the method can start to solve the problem (4) with a new value $\lambda$ having known values of $F(y)$ at previous iteration points. This, undoubtedly, will speed up the problem solution.

3. The search information obtained in the course of optimization can be presented as the set

$$\Omega_k = (y^i, z^i) : 1 \le i \le k, \tag{5}$$

where $y^i, 1 \le i \le k$, are points of previous iterations, $z^i, 1 \le i \le k$, are function values calculated at these points (in general $z^i, 1 \le i \le k$, may be vector values - see Section 2).

A possible way to decrease the complexity of search information operations is based on employing the set of joint space-filling curves [2-4] for reducing multidimensional data.

Consider the plural mapping

$$Y(x) = y^{[x]}(x - [x]), x \in (0, L + 1) \tag{6}$$

where $y^l(x), 0 \le l \le L$, is the partial mapping that maps the segment $[l, l+1]$ of real axes $x$ onto the $N$-dimensional search domain $D$ from (1), $[x]$ is the integer part of $x$. This mapping $p(x)$ can be formed in such a way that for any two close points $y'$,$y$" from $D$ will exist a partial mapping $y^l(x), 0 \le l \le L$, which produces close preimages $x'$,$x$" for the points $y'$,$y$" [2-4]. Therefore the procedure of searching "nearest neighbours" among multidimensional iteration points can be replaced by a search among scalar preimages.

Using the mapping $Y(x)$ from (6) we can transform the search information set $\Omega_k$ from (5) to the form

$$\omega_k = (x_i, z_i) : 1 \le i \le K = (L + 1)k, x_1 < x_2 < ... < x_k, \tag{7}$$

which contains all the preimages of previous iteration points $y^i, 1 \le i \le k$, in accordance with the mapping $Y(x)$. As it can be noted preimages in $\omega_k$ are placed in the ascending order.

Search information presented in the form (7) can be supplied with effective procedures for searching "nearest neighbours" for any point $y \in D$:

1. indicate the value $\eta, 1 \le \eta \le L$, that determines the neighbourhood of $y$

   $$D(y, \eta) = y' \in D : |y_j - y'_j| < (b_j - a_j)/2^\eta, y_i = y'_i, 1 \le i \le N, i \ne j$$

   where the vectors $a = (a_1, ..., a_N)$, $b = (b_1, ..., b_N)$ from (1);

2. calculate the preimages $x_j, 0 \le j \le L$, of the point $y$ in accordance with $Y(x)$ from (6), i.e. $y = Y(x_j), 0 \le j \le L$

3. form the set

$$I = i, 1 \le i \le K : \exists j, 0 \le j \le L, |x_i - x'_j| \le 2^{-\eta N},$$

which keeps the number of iteration point preimages located near with some preimages of $y$;

4. select the numbers

$$J = j, 1 \le j \le k : \exists i \in I, \exists l, 0 \le l \le L, x_l^j = x_i$$

which indicate the points $y^i, 1 \le i \le k$, having at least one preimage whose number belongs to the set $I$.

The complexity of the presented procedure can be evaluated as

$$T_2 = O(2N(\log_2(1/\epsilon))^2).$$

This dependence demonstrates that the complexity depends linearly on the dimension of the problem and as a binary logarithm on the number of previous iterations. As a result, the computational efforts needed for processing the search information decrease substantially (see (3) for comparison).

In particular for global optimization it can be added that we can construct multiextremal methods which don't require "nearest neighbours" search procedures at all. Instead of it these methods analyze one-dimensional intervals $(x_{i-1}, x_i), 1 \le i \le K$, from (7) directly (see, for instance, [4]).

To illustrate proposed approach we intend to present suitable optimization software.

**References**


[1] Gergel V.P. (1993), A Software System for Multiextremal Optimization, European Journal of Operation Research 65, N 3, 305-313.

[2] Gergel, V.P., Strongin, L.G., Strongin, R.G. (1987), The Vicinity Method in Pattern Recognition, Engineering Cybernetics (Transl. from Izv. Acad. Nauk USSR, Techn. Kibernetika 4, 14-22).

[3] Gergel, V.P., Strongin, R.G. (1992), Multiple Peano Curves in Recognition Problems, Pattern Recognition and Image Analysis 2, N 2, 161-164.

[4] Strongin, R.G. (1992), Algorithms for Multi-Extremal Mathematical Programming Problems Employing the Set of Joint Space-Filling Curves, Journal of Global Optimization 2, 357-378.

## Global Optimization and Decision Support

Eligius Hendrix

Global optimization methods is the name for a group of algorithms with the purpose to find the global optimum of a real valued continuous function over a feasible set, defined by a set of inequalities. Until recently most literature on global optimization focused on theoretical properties of the methods. At the moment we are working on a monograph based on the experience on finding answers for Global Optimization questions in the past five years. The central theme of this work is, what global optimization has to offer to a group of potential users. At one side there exists literature on global optimization, which mainly focuses on theoretical achievements of the methods, see Handbook on Global Optimization (Horst and Pardalos, 1995). At the other side there are potential users. The target group of our study uses mathematical modelling for research, though it does not exist of experts in optimization. In the work, stimulated by experience at the Agricultural University, the following categories of modellers and potential users of global optimization methods are distinguished:

- Researchers in agricultural and environmental studies
- designers
- OR decision scientists of environmental and agricultural planning problems.

In all the categories, mathematical modelling and optimization are used to get a better understanding of a practical problem, an object system. When the answers of the optimization are satisfactory for practical planning problems, the optimization routines can be build into Decision Support Systems to generate suggestions for the decisions to be made. In contrast to methods from linear programming and combinatorial optimization, the GLOP methods have hardly reached this level. This inability motivates our study.

An important question is, how the modeller can use his knowledge to select global optimization methods and to apply the knowledge further interactively during the solution process. Reversely, which useful information do the algorithms generate, which may help the modeller to get a deeper understanding of the practical problem which has been modelled and to speed up the solution process. An important notion is the division of GLOP methods in two groups. On one side there are deterministic methods such as Branch-and-Bound, which require special structure or at least analytical expressions (interval methods) of the model feeding the optimization problem. At the other side there are methods based on random search and/or local search which do not require any special structure. The choice of using a particular method depends on the information in the head of the modeller of one of our target groups who poses the question. This can be structure information e.g. bilinearity, or value information such as amount of optima, promising regions and bounds on first or second derivatives. The Oracle structure where a criterion value is provided by a (sub)program e.g. performing numerical integration, occurs very frequently. Bounds on the parameter values may be available, but we cannot make use of any information on the criterion function to be optimized. We will use the opportunity of the presentation at the workshop to enumerate globally the global optimization related questions we have worked on in the past few years since the last workshop and to report on the results. At the rest of the workshop there is the possibility to discuss the central theme and to discuss individual topics more thoroughly. A list of the reports which appeared on paper is added. Some topics follow here.

We start where we left five years ago. At the 1990 workshop we presented an algorithm for a design question which mathematically translated to "Find a feasible point of a set defined by quadratic inequalities" (Hendrix and Pinter, JOGO)

- What happened with the practical implementation? Confronted with interior point solutions, the designers where very pleased as they could allow mistakes in the production process without the product being out of its specifications. They called this a robust design.

- How can we optimise the robustness of the design? (Hendrix, Mecking and Hendriks, EJOR)

  Biologists applying population dynamics models derived criteria describing the development of a population depending on the ability for animals to travel from one living place (patch) to another.

- Given a budget for infrastructural improvements for a certain species, which improvements should be carried out?

  Technologists applying systems control came with the following mathematical question.

- Given a set of points in Rn, find a hyperrectangle (axes to be chosen freely) with minimum volume containing the set of points.

  Researchers applying farm management models came across some nasty nonlinear environmental restrictions.

- How to cope with the specific bilinear restrictions in a further linear model? (Bloemhof and Hendrix, EJOR)

  In 1993 we invested in the implementation of a derivative free local optimization routine and a graphical user interface for experimental purposes on questions with a typical Oracle structure.

- What does it look like?

  A firm producing metal filters used in sugar refiners offered a nice opportunity to use the optimizer for Oracle structured design problems.

- Create designs for filters which are strong and have a high throughput.

  For an ecological model builder fitting some 8 parameters to a 'sophisticated' model a local optimizer generated infinitely many 'optima', which after all appeared to be caused by numerical reasons, which we called ill conditioning.

- Develop heuristical methods which deliver the real optimum. (Hendrix, Mous, Roosma and Scholten, technical note)

  Ecological model builders are sometimes interested in the specific shape of a level set of a (oracle) goodness of fit function.

- Develop a method which generates a sample of a uniform distribution over a level set. (Klepper and Hendrix, Ecological modelling, Environmental Toxicology and Chemistry)

  Many random based global optimization methods guarantee to reach the optimum in limit. A typical question however is: Give me the best result you can obtain before tomorrow 9.00 a.m.

- What are good strategies allocating the budget of calculation time to local search and to global (random) search?

- When is it useful to do a global search anyway? (and other questions in Hendrix and Roosma, technical note)

  When science is rather a matter of questions than of answers, we are really proceeding. The topics mentioned here leave many questions open for research and discussion.

**References**


[1] E.M.T.Hendrix, J.Pinter, An Application of Lipschitzian Global Optimization to Product Design, Journal of Global Optimization 1, 389-401, 1991

[2] E.M.T.Hendrix, C.J.Mecking, Th.H.B.Hendriks, A Mathematical Formulation of Finding Robust Solutions for a Product Design Problem, Technical Note 9302, Department of Mathematics Wageningen, 1993 (accepted by EJOR)

[3] J.M.Bloemhof-Ruwaard, E.M.T.Hendrix, Generalized Bilinear Programming: an application in farm management, Technical Note 9305, Centre for Environmental Studies Wageningen, 1993 (accepted EJOR)

[4] O.Klepper, E.M.T.Hendrix, A Method for Robust Calibration of Ecological Models under Different Types of Uncertainty, Ecological Modelling 74, 161- 182, 1994

[5] E.M.T.Hendrix and J.Roosma, Global Optimization with a Limited Solution Time, Technical Note 94-06, July 1994 (submitted to JOGO)

[6] E.M.T.Hendrix, S.L.J.Mous, J.Roosma and H.Scholten, Optimality and search strategies for ill conditioned parameter estimation, Technical Note 94-14, Department of Mathematics, Wageningen 1994

[7] O.Klepper and E.M.T. Hendrix, A comparison of algorithms for global characterization of confidence regions for nonlinear models, Environmental Toxicology and Chemistry, 13, 1887-1899, 1994

# The Computation of the Essential Supremum by using Integral Methods

Jens Hichert[1], Armin Hoffmann and H.X. Phú

We give an equivalence between the tasks of computing the essential supremum of a summable function and of finding a special zero of a one-dimensional convex function. Interpreting the integral method as NEWTON-type method we show that the algorithm can work very slowly. For this reason we propose an idea of a faster version of the algorithm which is in some respect similar to the method of AITKEN/STEFFENSEN.

## Introduction

In many practical problems a global optimum of a real-valued objective function is needed. Therefore, global optimization became an important field of research in optimization. On the other hand, with some practical problems the global optimum can be an unsuitable solution, namely in such cases in which the global optimizer is an unstable state in the system investigated. This can happen if, for instance, the objective function has a jump on the location of the global optimum, or if there is a very narrow peak in a neighbourhood of this location, and so, from a numerical point of view, any small perturbation of the state parameters of the system will imply a significant worsening of the objective function. Therefore, it is sometimes more convenient to compute the essential supremum (or infimum) instead of the global one. Some global optimization methods work under certain conditions guaranteeing the equality of the essential supremum and the global supremum, such as continuity, robustness ([2], [3]), density([1]). Other methods try to find the global supremum even if it is much larger than the essential one. The first aim of our paper is to present an algorithm that really computes the essential supremum whenever the objective function is an $L_\infty$ function. This will be done in the next section. Secondly, in Section 3 we investigate the convergence speed of the algorithm depending on different types of the objective. In Section 4 we present a faster version of this algorithm in order to decrease the amount of computing time using more information about the function.

## An integral method

Consider a measurable function $f : D \to \mathbb{R}$ defined on a measurable set $D$ of $\mathbb{R}^n$ with respect to the Lebesgue measure $\mu$, where $0 < \mu(D) < +\infty$. Throughout this paper we assume $f \in L_\infty(D)$. We use the abbreviation

$$[f \geq \alpha] := \{x \in D : f(x) \geq \alpha\}.$$

In order to develop an algorithm for computing $\operatorname{ess\,sup}\{f(x) : x \in D\}$ we investigate the volume function $F : \mathbb{R} \to \mathbb{R}$ defined by

$$F(\alpha) = \int\limits_{[f \geq \alpha]} [f(x) - \alpha]\, d\mu,$$

which is motivated by the following properties (see [1]):

**Theorem 1** *$F$ is Lipschitzian, non-negative, non-increasing and convex for any $f \in L_\infty(D)$. Furthermore, $F$ has almost everywhere the derivative $F'(\alpha) = -\mu[f \geq \alpha]$.*

[1]This work was partially supported by the Deutsche Forschungsgemeinschaft within the Graduiertenkolleg "Automatisierung des Entwurfs analoger und gemischt analog-digitaler Strukturen"

**Theorem 2**

$$\alpha^* := \sup\{\alpha \in I\!R : F(\alpha) > 0\} = \operatorname{ess\,sup} f$$

*and*

$$\mu[f \geq \alpha] > 0 \qquad \forall \alpha > \alpha^*.$$

Based on these two theorems we state a NEWTON algorithm for finding the smallest zero $\alpha^*$ of the function $F$ which equals the essential supremum of $f$. We assume that one initial value $\alpha_0$ with $\alpha_0 < \alpha^*$ was found. Then, the algorithm generates an increasing level sequence $(\alpha_k)$ by the rule

$$\alpha_{k+1} \quad = \quad \alpha_k + \frac{F(\alpha_k)}{\mu\,[f \geq \alpha_k]} \qquad k = 0, 1, \ldots \qquad . \tag{1}$$

**Proposition 1 ([1], [2])** *The sequence $(\alpha_k)$ generated by (1) converges monotonously to $\alpha^*$.*

Due to the special structure of $F$, for every number $\alpha \in I\!R$ with $F(\alpha) = 0$ there is not any information about the distance $|\alpha - \alpha^*|$. However, Proposition 2.1 states that this situation does not occur within the NEWTON algorithm, apart from the situation $\alpha_k = \alpha^*$ for any $k$, which means that the algorithm has been terminated after a finite number of steps.

**Proposition 2 ([2])** *The set sequence $([f \geq \alpha_k])$ generated by (1) converges monotonously to $[f = \alpha^*]$, the set of the locations of the essential supremum of $f$.*

The iteration rule (1) is equivalent to the mean value level set method in [2]. But by considering the convex function $F$ we can easily come to other stopping conditions and further interesting aspects concerning the convergence speed and its acceleration. In connection with convergence investigations, some smoothness properties of $F$ are of interest.

**Proposition 3** *The function $m : I\!R \to I\!R$ with $m(\alpha) := \mu[f \geq \alpha]$ is left-hand continuous.*

**Proposition 4** *$F$ is continuously differentiable in an open set $U \subset I\!R$ iff $\mu[f = \alpha] = 0$ for all $\alpha \in U$.*

A connection between the smoothness of the objective function $f$ and the volume function $F$ provides the

**Proposition 5** *Let $D \subset I\!R^n$ an open set and $f \in C^1(D)$.*
*If $\nabla f(x) \neq 0$ a.e. on $D$, then $\mu[f = \alpha] = 0 \quad \forall \alpha \in I\!R$.*

We do not know whether the statement of Proposition 5 is reversible.

**Convergence speed of the level sequence**

The practicability and effectiveness of the integral method stated above mainly depends on the way of evaluating the functions $F(\alpha)$ and $m(\alpha)$. Until now, there has been just one appropriate method based on Monte Carlo models, computing both $F(\alpha)$ and $m(\alpha)$ in one procedure. Nevertheless, in every practical realization of the theoretical NEWTON algorithm one function evaluation will be very expensive. Therefore both the convergence order and speed of the level sequence $(\alpha_k)$ are of interest. In order to investigate this convergence order we are going to distinguish between two "types" of objective functions $f$. We want to consider $f$ to be of the type A if $\mu[f \geq \alpha^*] > 0$ holds. Otherwise, we consider $f$ to be of type B.
If $f$ is a function of type A, the well-known theory of the NEWTON algorithm is available and provides satisfactory statements about the speed of the convergence of the algorithm:

**Theorem 3** *Let $f$ be a function of type A and there exists an $\alpha \in \mathbb{R}$ with $F \in C^1(\alpha, \alpha^*)$. Then, either the algorithm stops after finite steps or the convergence of the sequence $(\alpha_k)$ generated by (1) is Q-superlinear. If $F'$ satisfies a Lipschitzian condition in a neighbourhood of $\alpha^*$ the convergence is Q-quadratic.*

However, a function of type A is not so typical for problems arising in practice. A more usual situation is given if the essential supremum of a function $f$ is only reached on a set of measure zero. In this case, due to Theorem 1 and Proposition 3, $F$ is differentiable in $\alpha^*$ and $F'(\alpha^*) = 0$. Therefore, the zero $\alpha^*$ of the function $F$ has a higher order. This means that the NEWTON algorithm for computing the essential supremum of a type B function converges in general only Q-linearly. The order of the zero increases with the dimension n of the problem. This means that an increasing dimension implies a decreasing convergence speed (see also Proposition 6). To avoid this drawback we propose an appropriate and faster algorithm useful for type B functions $f$ using more information about the order of the zero $\alpha^*$ of $F$.

**Speeding up the level sequence**

Let $\varphi : \mathbb{R} \to \mathbb{R}$ and $\alpha^*$ a zero of $\varphi$ of the order $\beta$ (this means that $\varphi \in C^{\beta+1}$ and $\varphi(\alpha^*) = \varphi'(\alpha^*) = \ldots = \varphi^{(\beta-1)}(\alpha^*) = 0$, $\varphi^{(\beta)}(\alpha^*) \neq 0$). Under this condition it is well-known fact that the modified NEWTON algorithm

$$\alpha_{k+1} = \alpha_k - \beta \frac{\varphi(\alpha_k)}{\varphi'(\alpha_k)} \qquad k = 0, 1, \ldots$$

converges locally Q-quadratically to $\alpha^*$. In order to apply this improvement in NEWTON's algorithm to our situation of type B functions, we consider at first a "prototype" of the function $f$.

**Proposition 6** *Let $f(x) = -\sum_{i=1}^{n} a_i |x_i - x_i^*|^p + y^*$, $a_i > 0 \quad \forall i = 1, \ldots n$ with some $p \geq 1$. Then, for each $\alpha_0 < \alpha^*$ the iteration*

$$\alpha_{k+1} = \alpha_k + \left(1 + \frac{n}{p}\right) \frac{F(\alpha_k)}{m(\alpha)} \qquad k = 0, 1, \ldots$$

*leads to $\alpha^*$ in one iteration step, that is, $\alpha_1 = \alpha^* = y^*$.*

Note that in Proposition 6 no information is needed about $x^*$ and $y^*$. We now assume that functions

$$\begin{aligned} g_{lb}(x) &= -\sum_{i=1}^{n} a_i |x_i - x_i^*|^p + y^* \\ g^{ub}(x) &= -\sum_{i=1}^{n} b_i |x_i - x_i^*|^p + y^* \qquad a_i,\, b_i > 0 \qquad \forall\, i \end{aligned}$$

exist being a lower and upper bound of $f$ respectively, that is,

$$g_{lb}(x) \leq f(x) \leq g^{ub}(x) \qquad \text{a.e. on } D. \tag{2}$$

Then with $q := \prod_{i=1}^{n} \left(\frac{b_i}{a_i}\right)^{1/p} < 1$ we can choose a stepsize

$$\beta := \left(1 + \frac{n}{p}\right) q \tag{3}$$

(again independent of $x^*$ and $y^*$) having the following properties:

**Proposition 7** *1. Using formula (3)*

$$\bar{\alpha} := \alpha + \beta \frac{F(\alpha)}{m(\alpha)} \leq \alpha^* \qquad \forall \alpha < \alpha^*.$$

*2. Furthermore,*

$$\frac{|\bar{\alpha}|}{|\alpha|} \leq 1 - q^2 \qquad \forall \alpha < \alpha^*.$$

In other words, if the inclusion (2) of $f$ is sufficiently sharp, the (theoretical) stepsize $\beta$ guarantees an increasing sequence $(\alpha_k)$ converging monotonously to $\alpha^*$, but faster than the sequence $(\alpha_k)$ generated by (1). The values $q$ and $p$ are, of course, not known. But under certain conditions we might assume that $q$ tends to 1, and an ideal stepsize $\beta = 1 + \frac{n}{p}$ with some $p \geq 1$ will be reached. In our algorithm we update the stepsize by assuming $f$ to be a prototype function as in Proposition 6:

$$\beta_k = \left(1 + \frac{n}{p_k}\right) \sqrt{\frac{s_k}{s_{k-1}} \frac{n + p_k}{n}} \tag{4}$$

where $s_k := \frac{F(\alpha_k)}{m(\alpha_k)}$. One way to update $p_k$ can be chosen by

$$p_k = n \left(\frac{s_k}{s_{k-1}} - 1\right). \tag{5}$$

Our algorithm contains additional considerations in order to ensure a monotonously increasing sequence $(\alpha_k)$ by

$$\alpha_{k+1} = \alpha_k + \beta_k s_k \qquad k = 0, 1, \ldots$$

with updates (4) and (5). Nevertheless, if such an iteration step works without any problems, (4) and (5) lead to

$$\alpha_{k+1} = \alpha_k + \frac{s_k^2}{s_k - s_{k-1}}$$

which is similar to one step of the method of AITKEN/STEFFENSEN (see [4]) to speed up the convergence of an iteration method.

## References


[1] PHÚ, H.X., HOFFMANN, A.: Essential Supremum and Supremum of Summable Functions. Submitted in 1995.

[2] CHEW SOO HONG, ZHENG QUAN: Integral Global Optimization. Springer, 1988.

[3] KOSTREVA, M.M., ZHENG QUAN: Integral Global Optimization Method for Solution of Nonlinear Complementarity Problems. Journal of Global Optimization, 5 (1994), pp. 181-193.

[4] STOER, J.: Numerische Mathematik 1. Springer 1994.

# Greedy Randomized Adaptive Search for a Location Problem with Economies of Scale

Kristina Holmqvist[1], Athanasios Migdalas and Panos M. Pardalos

One of the most central problems in continuous location theory requires the generation of optimal sites for $m$ new facilities to serve the demands from a set of $n$ customers. In distribution systems, the facilities typically represent warehouses or depots. The optimal locations of the new facilities are those which result in a minimum of total transportation and warehousing costs. In planning of real distribution systems, the number of new facilities to add is a key decision variable. The optimal value of $m$ is determined by considering the trade-off between warehousing costs, which include investment and operating costs, and transportation costs. Clearly, as warehousing costs are increased by adding new facilities to the system, the transportation costs will decrease since the average travel distance between warehouses and customers will be decreasing.

Obviously, this is a problem with great importance in planning shipment of goods. The concave costs implies that the more we ship through facility $i$, the cheaper it is per unit.

Problem Definition

There are $n$ customers each with demand $b_j$ which should be fulfilled by facilities out of a given set of $m$ candidate locations. At facility $i$ the total amount of goods that is shipped away from the facility is defined as the throughput level $y_i$. The cost for shipping one unit from facility $i$ to customer $j$ is denoted by $c_{ij}$ and the amount of goods that is transported the same way is denoted by $x_{ij}$. Thus, we can describe the location problem as:

$$\min \sum_{i=1}^{m} g_i(y_i) + \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

subject to

$$\begin{aligned}
\sum_{j=1}^{n} x_{ij} &= y_i, \quad i = 1,\ldots,m \\
\sum_{i=1}^{m} x_{ij} &= b_j, \quad j = 1,\ldots,n \\
y_i &\geq 0, \quad i = 1,\ldots,m \\
x_{ij} &\geq 0, \quad i = 1,\ldots,m, \;\; j = 1,\ldots,n
\end{aligned}$$

In this paper $g$ is a concave function and we chose to consider the following form of warehousing costs

$$g_i(y_i) = \begin{cases} 0 & \text{if } y_i = 0 \\ a_{1i} + a_{2i}y_i + a_{3i}\sqrt{y_i} & \text{if } y_i > 0 \end{cases}$$

where $a_{1i}$, $a_{2i}$ and $a_{3i}$ are given non-negative parameters. The constant term $a_{1i}$ represents a fixed investment cost, while the remaining terms provide a variable operating cost that depends on the throughput of the facility. The square root results in concavity of the $g$ function and accommodates economies of scale in the operation of the facility. Economy of scale is a normal phenomenon, since larger facilities can operate more efficiently and can utilize automated technologies in cost effective manner. Whenever the facilities are homogeneous, the cost parameters do not vary with $i$.

The problem is thus to determine the throughput level $y_i$ for each facility and the amount $x_{ij}$ of goods to be transported from facility $i$ to customer $j$, so as to satisfy all demands with minimum total warehousing-transportation cost.

[1]Research partially supported by CENIIT (Center for Industrial Information Technology).

| Problem | Size |
|---|---|
| cap71,72,73,74 | $16 \times 50$ |
| cap101,102,103,104 | $25 \times 50$ |
| cap131,132,133,134 | $50 \times 50$ |
| capa,b,c | $100 \times 1000$ |

Table 1: Problem sizes

Greedy Randomized Adaptive Search Procedure

The Greedy Randomized Adaptive Search Procedure (GRASP) [1] is an iterative process. This randomized sampling technique provides a feasible solution within every iteration. The final result is simply the best solution found over all iterations. Each iteration consists of two phases, a construction phase and a local search procedure. In the construction phase a randomized greedy function is used to build up an initial solution. This solution is then exposed for improvement attempts in the local search phase.

When implementing a GRASP for a particular problem the procedure for constructing the initial (feasible) solution must be decided. Briefly, the construction phase can be described as iteratively adding one element to the incumbent (incomplete) solution. The strategy for choosing the next element is based on randomly choosing the element from a list which is built up with regard to a greedy function. The heuristic is adaptive in the sense that the effect of already chosen elements are regarded.

Also, the neighbourhood function used in the local search phase must be defined. Of course, different problems require different construction and local search strategies but the advantage of GRASP before other heuristics is that when these strategies are defined, there are only a couple of parameters to tune (the size of the candidate list and the number of GRASP iterations.)

**Test Problems**

We have used test problems from the OR-library, available on the world wide web at `http://mscmga.ms.ic.ac.uk/`. However, these problems are not including economies of scale and therefore we have to add some values for the variables in the concave cost function $g$. The problems have similar structures but different number of facilities and customers. There is also a distance matrix for the distances between each facility and customer. This distance matrix is used to evaluate the objective function value, since we use the distance as transportation cost. In Table 1 the problem sizes (number of facilities $\times$ number of customers) are given. The values used for the variables in the concave function are the same for all locations in all problem instances. We have chosen to set the parameter values of $a_{2i} = 20$ and $a_{3i} = 100$. The fix cost for allocating facility $i$ gives us the $a_{1i}$ values. This fixed cost is given in the test problems. The values of $a_{2i}$ and $a_{3i}$ should ensure that the transportation cost will not be dominating.

Numerical Results

The measurements presented in this section were done on an unloaded Sun SPARCstation 20/50. We compiled the C program with the SunSoft C compiler version 3.0.1 using the option `-xO2` and the Fortran 90 program with the Cray Fortran 90 compiler version 1.0.3 using the option `-O2`. Timing was done with the routines `clock_gettime(3R)` and `TIMEF(3F)` respectively.

**Conclusions**

The GRASP we present is capable of solving large problems in acceptable time.

**References**

[1] Thomas A. Feo and Mauricio G. C. Resende (1995) Greedy Randomized Adaptive Search Procedures, JOGO 6, 109-133

# Linearly Constrained Global Optimization of Functions with Concave Minorants

Reiner Horst
(based on different joint work with M. Nast and N.V. Thoai)

## 1. Functions with concave minorants

A common property of Lipschitz functions, d.c. functions and some other function classes of interest in global optimization is that, at every point of the domain, one can construct a concave function which coincides with the given function at this point, and underestimates the function on the whole domain (concave minorant, cf. Khamisov (1995)). We present a new branch and bound algorithm for minimizing such a function over a polytope which, when specialized to Lipschitz or d.c. functions, yields improved lower bounds as compared to the bounds in previous branch and bound methods. Moreover, the linear constraints will be incorporated in a straightforward way so that "deletion–by–infeasibility" rules can be avoided. Finally, we show that these bounds can be improved further then the algorithm is applied to solve systems of inequalities.

**Definition 1.1.** (Khamisov (1995)) *A function $f : S \to \mathbb{R}$, defined on a nonempty convex set $S \subseteq \mathbb{R}^n$ is said to have a concave minorant on $S$ if, for every $y \in S$, there exists a function $F_y : S \to \mathbb{R}$ satisfying*

*(i)* $F_y(x)$ *is concave on* $S$,

*(ii)* $f(x) \geq F_y(x) \; \forall \; x \in S$,

*(iii)* $f(y) = F_y(y)$.

*The functions $F_y(x)$ are called concave minorants of $f(x)$ (at $y \in S$) and the class of functions having a concave minorant on $S$ will be denoted by CM(S).*

**Example 1.2.** Examples of functions on CM(S) include Lipschitz continuous functions (and more general certain Hoelder continuous functions), functions representable as differences of convex functions (d.c. functions), weakly convex functions on $\mathbb{R}^n$.

**Lemma 1.3.** *Let $\{x_k\}$ and $\{y_k\}$ be sequences in $S$ such that $\lim_{k\to\infty} x_k = \lim_{k\to\infty} y_k = s \in S$. Then, for each of the concave minorants given in Example 1.2, we have*

$$\lim_{k\to\infty} F_{y_k}(x_k) = f(s).$$

## 2. Branch and Bound Approach

We consider the problem

$$\begin{array}{l} \text{minimize } (x), \\ x \in D \end{array} \tag{1}$$

where $D$ is a polytope in $\mathbb{R}^n$ with nonempty interior, and $f \in CM(S)$ for some $n$–simplex $S \supseteq D$.

A lower bound for $f$ over the intersection of an $n$–simplex $S$ with the feasible set is obtained by the following result:

**Proposition 1.4.** *Let $S = [v_0, \ldots, v_n]$ be an $n$–simplex with vertices $v_0, \ldots, v_n$, $D$ be a polytope in $\mathbb{R}^n$, $T$ be a nonempty finite set of points in $S$, and $f \in CM(S)$ with concave minorants $F_y$. For each $y \in T$, let $\varphi_y$ denote that affine function which is uniquely defined by the system of linear equations*

$$\varphi_y(v_i) = F_y(v_i), \quad i = 0, \ldots, n.$$

*Then, the optimal value $\beta(S \cap D)$ of the linear program*

$$\begin{array}{ll} \textit{minimize} & t \\ \textit{s.t.} & \varphi_y(x) \leq t,\ y \in T,\ x \in S \cap D \end{array}$$

*is a lower bound for* $\min\{f(x) : x \in S \cap D\}$.

Notice that, while solving the LP for $\beta(S \cap D)$ a finite set $Q(S)$ of feasible points is detected.

**Algorithm**

Initialization:

Determine an initial $n$–simplex $S \supseteq D$, the lower bound $\beta(S \cap D)$, and the set $Q(S)$. Set $\beta(S) = \beta(S \cap D)$, $Q = Q(S)$, $\alpha = \min\{f(x) : x \in Q\}$ and choose $z \in Q$ satisfying $f(z) = \alpha$. Define $M = \{S\}$, set $\beta = \beta(S)$, $k = 1$.

Iteration k:

If $\alpha = \beta$, then stop; $z$ is an optimal solution, and $\alpha$ is the optimal objective function value of Problem (1).
Otherwise, choose

$$S \in M \quad \text{satisfying} \quad \beta(S) = \beta.$$

Bisect $S$ into the simplices $S_1$ and $S_2$.
Compute $\beta(S_i \cap D)$, $i = 1, 2$; and

$$\beta(S_i) = \max\{\beta(S);\ \beta(S_i \cap D)\} \quad (i = 1, 2).$$

Set $Q = Q \cup \{Q(S_1), Q(S_2)\}$, update $\alpha = \min\{f(x) : x \in Q\}$, and choose $z \in Q$ satisfying $f(z) = \alpha$.
Set

$$\begin{aligned} \overline{M} &= (M \setminus \{S\}) \cup \{S_1, S_2\}, \\ M &= \overline{M} \setminus \{S : \beta(S_1) \geq \alpha\}, \\ \mu &= \begin{cases} \min\{\beta(S) : S \in M\} & \text{if } M \neq \emptyset \\ \alpha & \text{if } M = \emptyset \end{cases} \end{aligned}$$

and go to iteration $k + 1$.

**Proposition 1.5.** *In Problem (1), let $f \in CM(S)$ be continuous on the initial simplex $S$. Moreover, for each pair of sequences $\{x_k\}$, $\{y_k\} \subset S$ such that $\lim_{k\to\infty} x_k = \lim_{k\to\infty} y_k = s$ assume that $\lim_{k\to\infty} F_{y_k}(x_k) = f(s)$. Then, if the algorithm does not terminate after a finite number of iterations, we have*

$$\lim_{k\to\infty} \beta_k = \lim_{k\to\infty} f(z_k) = \lim_{k\to\infty} \alpha_k,$$

*and every accumulation point $z*$ of the sequence $\{z_k\}$ is an optimal solution of Problem (1).*

### 3. Systems of Inequalities and Numerical Results

Probably the most interesting application of the approach is in solving systems of inequalities where all of the functions involved are in CM(S). Here we obtain a drastical further improvement of earlier bounds. Details on this and on numerical experiments and comparisons are reported in the talk.

**References** (selected)


[1] Horst, R., Nast, M. and Thoai, N.V. (1995), New LP–Bound in Multivariate Lipschitz–Optimization: Theory and Applications. *Journal of Optimization Theory and Applications*, Vol. 86, No. 2, 369–388.

[2] Horst, R. and Nast, M. (1996), Linearly Constrained Global Minimization of Functions with Concave Minorants; accepted for publication in *Journal of Optimization Theory and Applications*.

[3] Horst, R. and Thoai, N.V. (1988), Branch and Bound Methods for Solving Systems of Lipschitzian Equations and Inequalities. *Journal of Optimization Theory and Applications*, Vol. 58, 189–146.

[4] Khamisov, O. (1995), Functions with Concave Minorants; accepted for publication in Floudas, C. and Pardalos, P.M. (eds.), *State of the Art in Global Optimization*, Kluwer, Dordrecht–Boston–London.

## The Pareto Approach to Balancing Local and Global Search

Donald R. Jones, William Baritompa and Yaroslav D. Sergeyev

For an algorithm to be truly global, some effort must be allocated to what may be called global search—search done primarily to ensure that potentially good parts of the space are not overlooked. On the other hand, to achieve efficiency, some effort must also be placed on local search—search done in the area of the current best solution(s). In fact, one can argue that the essence of a global search procedure lies in how it balances these competing objectives.

In this paper, we show that Lipschitzian optimization, interval analysis, and Bayesian optimization—however different they may be conceptually—all balance global and local search in essentially the same way. We then reduce this common technique for global/local balance to its bare essentials, and use this simplified technique as the basis of a new heuristic called DIRECT. This heuristic turns out to be truly remarkable, and with small modifications can be applied problems that are continuous, discrete, smooth, nonsmooth, constrained, or unconstrained.

The origin of DIRECT goes back four years, when the first author was simultaneously working in the areas of Lipschitzian optimization, Bayesian optimization, and interval analysis. These methods have many similarities. For one thing, all of them are space-partitioning algorithms. That is, they all work with a partition of the search space into regions and, in each iteration, they select a region, subdivide it, and sample new points within the resulting subregions. But there is a deeper similarity: in all three methods, the criterion for selecting regions can be divided into two terms. The first term depends only on the goodness of the *sampled points* in a region. By itself, this term would cause us to do local search. The second term, on the other hand, depends only on the *size* of the region. By itself, this term would lead us to do global search. The three methods combine the two terms in different ways, but every combination has the following key property: the attractiveness of a region increases as the sampled function values get better or as the size gets bigger.

Once the above similarities were apparent, all the heavy theoretical baggage of stochastic processes, Lipschitz constants, and so on, began to seem like nothing more than very elaborate ways to justify particular *formulas* for balancing global and local search. But if our goal is merely to select those regions which do well on sampled function values and size, why not just select those regions that are Pareto optimal with respect to these two criteria? That is, why not select all those regions that are *not dominated* by another region on the two criteria of sampled function values and size. Now a region is said to "dominate" another on two criteria if it is strictly better on both criteria, or the same on one criterion and strictly better on the other. Thus a nondominated region has the property that all regions of the same size or larger have *worse* sampled function values. This Pareto idea was the key insight that led to DIRECT and all its variants.

The Pareto selection criteria is extremely clean. One of its nicest properties is the lack of any parameters (Lipschitz constants, convergence tolerances, etc.) that determine the balance between local and global search. Instead of using such parameters to *fix* the global/local balance and then select a *single* region, the Pareto method identifies *several* regions in each iteration (the Pareto optimal set). Some regions in the Pareto set are good for local search and some are good for global search. Another attractive feature of Pareto selection is that it is nonmetric. Whether or not one region dominates another depends only on *relative* function values. As a result, we get exactly the same sequence of iterates whether we are minimizing $f$, $\log(f)$, or $\exp(f)$.

The DIRECT algorithm is nothing but a space-partitioning heuristic that uses an easy-to-manage partitioning strategy together with this Pareto selection rule or intuitive variations of the rule. On the traditional Dixon-Stiglitz test suite, DIRECT works about the same as, and often much better than, the best existing black-box heuristics. In our opinion, it is simply

amazing that a deterministic algorithm with no tuning parameters, one that relies entirely on the Pareto criterion, even works, let alone works so well.

The present paper makes significant extensions to the original version of DIRECT that was published in 1993. One of the most interesting extensions is to discrete problems such as 0-1 programming and permutation problems (scheduling, etc.). We also show how the method can be extended to incorporate gradient information (if it is available) and to handle general nonlinear inequality constraints. The details of these extensions are given in the paper. Here we will here limit ourselves to a few comments.

With respect to discrete problems, notice that when we discussed the idea of dividing the "search space" into "regions," nothing we said implied that the search space was Euclidean or that the regions were rectangles (as in the original DIRECT algorithm). This is why DIRECT applies to discrete problems. To apply DIRECT, one only need be able to devise a way of partitioning the discrete space into subsets, measuring the size of these subsets, and sampling points within subsets.

With respect to using gradient information, the reader may wonder why we even bother. After all, using gradient information seems to be against the spirit of general-purpose black-box heuristics. The reason we have considered using gradient information is that using gradients tends to focus the search, allowing the optimum to be found in fewer iterations. In low dimensions, the savings in iterations may be swamped by the high cost of performing numerical derivatives, and so the value of gradients is doubtful. But in higher dimensions, the use of gradients can allow the optimum to be found before the search tree explodes exponentially and the search becomes hopelessly bogged down. So the true value of using derivatives lies in the way it helps DIRECT stretch into higher dimensions.

Extending DIRECT to handle constraints was the most difficult part of this research. In fact, we have only succeeded so far in handling *in*equality constraints. The technique we use for handling these constraints is fairly novel, as it does not involve penalty or Lagrangian functions. At an intuitive level, the technique amounts to developing a criterion function that is related to the likelihood that further search in a region will reveal a feasible point that beats our current best solution by $\epsilon$. We then select all those rectangles that have the property that, for some particular $\epsilon > 0$, they have the best value of this criterion. This constrained version is less simple than the unconstrained version, but it is the natural generalization of it.

In its constrained version, DIRECT has been used at General Motors for solving small to medium (5 to 20 variable) problems in which the goal is to optimize the parameters of a mechanical design. These mechanical design problems are hard enough to confuse local optimizers, but they are far from pathological, and DIRECT has proven to be quite effective. So far the applications include optimizing the design of piston shapes, crankshaft counterweights, heat exchangers, blanking dies, and advanced shock absorbers.

## To the global minimization of functions with concave minorant[1]

Oleg V. Khamisov

Let a set $R \subset E^n$ and a real valued function $f(x), \quad f : R \to E^1$ be given.

**Definition 1** *A function f(x) is said to have a concave minorant or concave support function on R if there exists function $\varphi(x,y), \quad \varphi : E^n \times R \to E^1$, continuous in x for any fixed y, such that*

1. *$\varphi(x,y)$ is concave in $x$;*
2. $f(x) \geq \varphi(x,y) \quad \forall (x,y) \in R \times R;$ (1)
3. $f(y) = \varphi(y,y) \quad \forall y \in R.$ (2)

The function $\varphi(x,y)$ is called a concave minorant or concave support function of $f(x)$, constructed at the point $y \in R$. A set of all functions $f(x), \ f : R \to E^1$ which has a concave minorant on $R$ is denoted by $CM(R)$ and each function $f \in CM(R)$ is called a c.m. function on $R$ ("c.m." is an abbreviation of "function with a concave minorant"). The function $f \in CM(E^n)$ is called a c.m. function.

Below we assume that $R \in E^n$ is a compact set. The functional class $CM(R)$ is quite large, since it is not difficult to see that any Lipschitzian function $f(x)$ is also a c.m. function with

$$\varphi(x,y) = f(y) - L\|x - y\|,$$

where L is a Lipschitz constant. Main properties of c.m. functions and comparison with other classes of functions are given in [1]. Let us shortly describe some properties of c.m. functions.

**Proposition 1** *Each function $f \in CM(R)$ is a l.s.c. function on R.*

**Definition 2** *Let $D \subset E^n$ be a convex set. A function $f : D \to E^1$ is called d.c. on D if there are two convex functions $p : D \to E^1, q : D \to E^1$ such that*

$$f(x) = p(x) - q(x), \ \forall x \in D.$$

*A function that is d.c. on $E^n$ will be called a d.c. function.*

**Proposition 2** *A function $f : E^n \to E^1$ is a d.c. function if and only if*

$$f(x) = \sup_{y \in Y} \psi(x,y),$$

*where*

$$\psi(x,y) = c(y)^T(x - y) + r(y) - q(x),$$

*$c(y) \in E^n, r(y) \in E^1, \ y \in Y$, Y is some nonempty set and $q(x)$ is a continuous convex function.*

The next proposition follows directly from Definition 1.

**Proposition 3** *Let c.m. functions $f(x), f_i(x), i = 1, \ldots, m$ be given. Then the following statements are true.*

(*i*) *Any nonnegative combination of functions $f_i(x)$ is a c.m. function;*
(*ii*) $\max_{1 \leq i \leq m} f_i(x)$ *and* $\min_{1 \leq i \leq m} f_i(x)$ *are c.m. functions;*
(*iii*) $f^+(x) = \max\{0, f(x)\}$ , $f^-(x) = \min\{0, f(x)\}$ *are c.m. functions.*

[1]Supported partialy by the Russian Fund of Fundamental Investigations

Proposition 1 describes quite general properties of c.m. functions. If we have some (nonconvex) function how to recognize whether this function is a c.m. function and, moreover, if yes, then how to construct its concave minorant ? We give a description of rather a wide subclass of the class of c.m. functions and give rules for the constructing the concave minorant for a function from this subclass.

**Definition 3** *If function* $f : R \to E^1,\ R \subset E^n$ *satisfies*

$$f \in CM(R), -f \in CM(R), \tag{3}$$

*then* $f$ *is called a c.m. symmetric function on* $R$.

A set of all c.m. symmetric functions on $R$ is denoted by $CMS(R)$. If $f \in CMS(E^n)$, then $f$ is called a c.m. symmetric function. It follows from (3) that for every $f \in CMS(R)$ there exists the function $\varphi^-(x,y), \varphi : E^n \times R \to E^1$ that is continuous and concave in $x$ and the function $\varphi^+(x,y), \varphi : E^n \times R \to E^1$ that is continuous and convex in $x$, such that

$$\varphi^-(x,y) \le f(x) \le \varphi^+(x,y),$$

$$\varphi^-(y,y) = f(y) = \varphi^+(y,y), y \in R.$$

Function $\varphi^+(x,y)$ is called a convex majorant and $\varphi^-(x,y)$ is called a concave minorant of the function $f(x)$. By virtue of Proposition 1.1 a c.m. symmetrical function is continuous.

**Proposition 4** *Let* $f \in CMS(R)$ *and* $f_i \in CMS(R)$, $i = 1, \dots, m$, $m > 1, R$ *be a compact set. Then*

$(i)$ $\sum_{i=1}^m \lambda_i f_i \in CMS(R), \lambda_i \in E^1$;

$(ii)$ $f^2 \in CMS(R)$;

$(iii)$ $f_1 * f_2 \in CMS(R)$;

$(vi)$ *if* $f(x) > 0, \forall x \in R$, *then* $\frac{1}{f(x)} \in CMS(R)$;

Consider the following mathematical programming problem

$$\min f(x),$$

$$x \in R,$$

where $R \subset E^n$ is a compact convex set., $f \in CM(R)$. We call this problem the c.m. programming problem.

Let $x^1, x^2, \dots, x^k$ be some points in $R$. Then

$$f(x) \ge \max_{1 \le j \le k} \varphi(x, x^j) = f_k(x), \forall x \in R$$

where $\varphi(x,y)$ is a concave minorant of $f(x)$. We call the problem

$$\min f_k(x), \tag{4}$$

$$x \in R \tag{5}$$

an approximating c.m. problem since

$$f^* = \min_{x \in R} f(x) \ge \min_{x \in R} f_k(x) = f_k^*.$$

Problem (4)-(5) is again a c.m. programming problem and, therefore, multiextremal, but here we have the advantage in the special form of the objective function. More exactly, it was shown in [2] that $f_k(x)$ is a d.c. function since

$$f_k(x) = f_k^+(x) - f_k^-(x),$$

where

$$f_k^+(x) = -\min_{1\le i\le k}\{\sum_{j=1,j\neq i}^{k}\varphi(x,x^k)\},$$

$$f_k^-(x) = -\sum_{j=1}^{k}\varphi(x,x^k)$$

and both $f_k^+(x)$ and $f_k^-(x)$ are convex. Introducing now an additional variable $x_{n+1}$ we can reduce the approximation c.m. problem to the following one

$$\min\{x_{n+1} - f_k^-(x)\}, \tag{6}$$

$$f_k^+(x) \le x_{n+1}, \tag{7}$$

$$x \in R. \tag{8}$$

The feasible domain in problem (6)-(8) is convex and the objective function is concave. Hence, (6)-(8) is a concave programming problem. Thus, the problem of global minimization of a c.m. function over a convex set can be approximated by the sequence of concave programming problems. This fact seems to be the natural generalization of the approximation of a convex programming problem by a sequence of linear programs. Therefore, we can say that in general almost each mathematical programming problem can be approximated by a sequence of auxiliary problems which are not more complicated than the concave programming problem.

We used term "approximation" to emphasize that in this way we obtain only bounds of the global minimum since, strictly speaking, we have to prove some convergence conditions.

Essential part of the contribution is devoted to the numerical testing of the described approach in global optimization and discrete and stochastic programming.

**References**


[1] O.Khamisov, *Functions with concave minorant.A general view* Manuscript of Institute of Operations Research, University of Zurich, 1994.

[2] H.Tuy, *Convex programming with an additional reverse convex constraint*, J.O.T.A. 52, 463-485, 1987.

# On a Mechanism of Natural Formation and its Use in Global Optimization

Victor Korotkich

Molecular conformation and protein folding have been subjects of considerable recent interest in global optimization [1], [2]. The development of efficient algorithms calls for further investigation on the laws governing these processes of formation in nature which still have many questions unanswered. It is becoming clear that what happens in biochemical systems is not a thermodynamical but dynamical problem which is not explainable by extremum principles but by the laws of spacetime dynamics [3]. In this sense simulated annealing which is based on an extremum principle has limits to simulate formation in nature.

In this paper we propose and address a model of formation in nature in an attempt to determine a mechanism which can be used for the algorithms development, i.e. to make the artificial dynamics of algorithms to reproduce formation in nature. A new approach to provide a way to deal with the problem is put forward. The main concepts of the approach are structural complexity and symmetry. There are many concepts of complexity which appear as different manifestations of intuitive notions of what the word ought to mean [4]-[9]. A concept of structural complexity is introduced in an attempt to provide a measure of complexity for global description of a particular sequence [10]. The concept is expressed in terms of relationships between integers and can be viewed as a measure of relationships between sequence components. A concept of structural symmetry is defined to formally capture a symmetry which underlies the relationships.

In section 1 a short account of the foundations upon which our study rests is provided, namely: the concepts of structural complexity and symmetry. Of necessity, we concentrate on background material giving first a brief description of the concepts and afterwards sketch how they come together. As well as summarizing the concepts we take the opportunity to introduce some notation and terminology.

At the heart of the approach is the idea that structural complexity and symmetry can be used to study formation in nature. In particular, structure-forming with the maximum in structural symmetry is suggested as a decisive criterion for its modeling. This statement becomes a formal definition after we specify the concepts involved in the formulation.

To provide a context in which to explore the idea a model of structures formation is introduced in section 2. It is employed in an indirect effort to get insight into formation in nature. The model ignores many complex effects and is a simple tool that gives a way to describe structures formation in terms of the spacetime symmetry only, i.e. structural symmetry. Symmetry is one of the most fundamental concepts in describing nature the emphasis on which has led to many impressive successes in a variety of fields and it arises naturally to use structural symmetry as a guiding principle in describing formation in nature.

Specifically, in the model a formation of a structure is represented by a certain spacetime pattern emerging from interactions with the environment. Structural symmetry is brought into focus when the performance of a formation is characterized by the structural symmetry of the pattern. To assess the total performance all patterns produced by a formation in terms of structural symmetry are considered. A formation is called optimal if it has the highest total performance with respect to all formations. In such a manner a concept of optimality is introduced without appeal to any explicit criterion of goodness and becomes rooted only in the spacetime symmetry, i.e. structural symmetry.

The problem of fundamental importance is to propose a model of structures formation which is relevant to formation in nature. We attempt to approach this problem by the proposal that the optimal formation of the model imitates formation in nature. In a search for the optimal formation we exploit a connection between structural complexity and symmetry [11], [12] and use it to represent the optimal formation in terms of structural complexity.

It turns out that the model under the proposed definition of optimality exhibits many of new phenomena encountered in biochemical systems. In particular, the optimal formation operates at "the edge of chaos" and is in agreement with the recent experimental finding of long-range correlations in the human noncoding sequences, which are very likely to be general characteristic feature of nucleotide organization in DNA [13] - [17]. More importantly, the fact that a natural phenomena formation appears to be optimal under the model definition seems to indicate that the model is indeed related to formation in nature.

Having established in section 2 the concept of optimality, we consider in section 3 a mechanism, which constructs rules of the optimal formation. It is important that the mechanism admits two different descriptions.

The first description, called structural complexity, is expressed in terms of the structural complexity machinery and allows to discover explicitly principles that are at work in the optimal formation. The description sheds light on the optimal formation rules construction, in particular it shows how they change as a result of interacting with the environment.

The second description, called natural phenomena, is based on using causal powers of natural phenomena processes. This description arises from the properties of the first one and turns out to be connected with self-organizing processes of formation in nature. Figuratively, it equates the computational powers of computing devices with the causal powers of natural phenomena processes and offers a means to propose a "super-Turing" model in computer science, whose computational power can surpass that of the Turing model(for more details see [18], [19]).

This mechanism description gives a way to use these processes for the development of global optimization algorithms which can reproduce formation in nature. In particular, at the end of section 3 the natural phenomena description of the mechanism is extended to be used in global optimization and a global optimization algorithm represented in its terms is proposed.

**References**


[1] P.M. Pardalos, D. Shalloway, and G. Xue, *Optimization Methods for Computing Global Minima of Nonconvex Potential Energy Functions*, Journal of Global Optimization, Vol. 4, (1994), pp. 117-133.

[2] D.M. Deaven and K.M. Ho, *Molecular Geometry Optimization with a Genetic Algorithm*, Physical Review Letters, Vol. 75, (1995), pp. 288-291.

[3] I. Prigogine, *New Perspectives on Complexity*, in The Science and Praxis of Complexity, The United Nations University, Tokyo, (1985), pp. 107-118.

[4] G.J. Chaitin, *Information, Randomness, and Incompleteness*, World Scientific, (1987).

[5] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman and Company, San Francisco, (1987).

[6] M. Gell-Mann, *The Quark and the Jaguar. Adventures in the Simple and the Complex*, Freeman and Company, New York, (1994).

[7] G. Nicolis and I. Prigogine, *Exploring Complexity*, Freeman and Company, New York, (1989).

[8] A.N. Kolmogorov, Probl. Inf. Transm., Vol. 1 (1965).

[9] R.J. Solomonoff, Inf. and Control., Vol. 7 (1964).

[10] V. Korotkich, *Integer Code Series with Some Applications in Dynamical Systems and Complexity*, Communications on Applied Mathematics, Russian Academy of Sciences, The Computing Center, Moscow, (1993).

[11] V. Korotkich, *Multicriteria Analysis in Problem Solving and Structural Complexity*, Advances in Multicriteria Analysis, P.M. Pardalos, Y. Siskos and C. Zopounidis (Editors), Kluwer Academic Publishers B.V., (1995), (to appear).

[12] V. Korotkich, *Symmetry in Structural Complexity*, Physical Review Letters, LG5781, 10 July 1995 (submitted).

[13] A. Arneodo et. al., *Characterizing Long-Range Correlations in DNA Sequences from Wavelet Analysis*, Physical Review Letters, Vol. 74, (1995), pp. 3293-3296.

[14] M. Ya. Azbel, *Universality in a DNA Statistical Structure*, Physical Review Letters, Vol. 75, (1995), pp. 168-171.

[15] W. Ebeling and T. Poshel, Europhys. Lett. 26, 241 (1994), and references therein.

[16] I. Kanter and D.A, Kessler, *Markov Processes: Linguistics and Zipf's Law*, Physical Review Letters, Vol. 74, (1995), pp. 4559-4562.

[17] C.K. Peng et. al., Nature, London, Vol. 356, 168, (1992).

[18] H.T. Siegelmann, *Computation Beyond the Turing Limit*, Science, Vol. 268, 28 April 1995, and references therein.

[19] J. Glanz, *A Quantum Leap for Computers*, Science, Vol. 269, 7 July 1995.

# Algorithmic Generation of the Mathematical Programming Model for Process Network Synthesis

Z. Kovacs, F. Friedler and L. T. Fan

Process network synthesis (PNS) is involved in the design of any process system for producing desired products from available materials. A process system is a network of operating units where a specified number of input materials with known quality are transformed in an operating unit into a specified number of output materials by altering their physical, chemical or biological properties. The importance of PNS arises from the fact that essentially every product of the chemical and allied industries, including the petrochemical industry, is manufactured by such a network; moreover, the profitability of the same product from different networks may vary widely. Therefore, the generation of the globally optimal solution of PNS is indeed essential.

The capability of PNS methods has expanded due to the recent development of high performance computers and effective mathematical programming methods. Thus, optimal or near optimal solutions can often be generated for mathematical programming models of PNS problems. Nevertheless, little fundamental information is available on the relation between the optimal solutions of a PNS problem and its model. New questions, therefore, have arisen: (i) what form should the mathematical model take to have a consistently valid optimal solution of the PNS problem modeled? (ii) how can the model be generated algorithmically? Unexpected answers to these questions for simple classes of PNS problems have indicated that it is indeed not trivial to resolve these questions.

Traditionally, PNS has been viewed as involving three major tasks: representation, evaluation, and search. This division has been useful in establishing the theory of PNS. To facilitate the resolution of the new questions, it has been divided into two major steps: (i) the generation of a mathematical programming model of a given class of PNS problems, and (ii) the solution of the resultant model. While the first step is the true "synthesis" part, the second, in reality, is the "analysis" part of PNS. The majority of the available methods, in fact, deals only with the second step, i.e., analysis. So far, the first step is essentially performed manually in an ad hoc manner.

The present work focuses on the model generation step, i.e., the first step, of PNS from the mathematical point of view. It proposes the basic definitions and introduces fundamental theorems for solving this step of PNS. The result is illustrated with the exact solution of a simple class of PNS problems.

## On relaxing the hypotheses for the application of multi level single linkage

Marco Locatelli

We consider the global optimization problem:

$$\min_{x \in X} f(x), \;\; X \subseteq \Re^d.$$

Multi Level Single Linkage (MLSL) is one of the most well known stochastic global optimization methods. The main idea is based on the will of doing clever local searches, improving upon methods like Multistart, where a local search is started from every sampled point. In order to avoid useless and expensive local searches, i.e. local searches which take to already detected minima, MLSL analyzes chains of descending and "close" sampled points and starts a local search from any point such that it has no "close" points with function value not greater than its function value. The meaning of "close" is related to a parameter $\alpha_k$ which is decreased, slowly enough, at any step. The algorithm can be described as follows:

1. let $N > 0$ and $\gamma \in (0, 1]$ be fixed and $\alpha_k := \pi^{-1/2}(\Gamma(1 + d/2)\sigma \frac{\log kN}{kN})^{1/d}$;
2. at step $k$ generate a uniform random sample of size $N$ in $X$;
3. sort the whole sample of $kN$ points in order of increasing function value, and select the $\gamma kN$ best of them; the resulting sample is called reduced sample;
4. apply a local search to the points of the reduced sample which satisfy the following conditions:
   - have not any point with lower function value at a distance smaller or equal than $\alpha_k$;
   - are not within a distance $d_1 > 0$ from an already detected minimum;
   - are not within distance $d_2 > 0$ from the border;
5. check some stopping rule and if it says of going on repeat the whole procedure.

A part from its good practical performance, it has also nice theoretical properties:

- almost sure convergence of the record (the best point observed till the current step) to the global optimum;
- finite expected number of local searches for $\sigma > 4$;
- every local minimum is detected in finite time with probability 1.

The preceding results are obtained under the following assumptions:

1. $f \in C^2$;
2. $X$ compact, convex and with non empty interior;
3. finite number of stationary points;
4. stationary points in the interior of $X$.

In this paper we relax all these hypotheses except the second one. The reason is that a lot of functions which can occur in global optimization do not enjoy the properties given above. For instance, the fourth hypothesis excludes all concave functions whose minima belong to the border of $X$. A simple function as:

$$f(x,y) = \begin{cases} x^2 & -1 \leq x \leq 1 \quad 0 \leq y \leq 1 \\ x^2 + (y-1)^2 & -1 \leq x \leq 1 \quad 1 \leq y \leq 2 \end{cases}$$

violates both the first and the third hypothesis. We now introduce some definitions, which are generalizations of the classical definitions of maximum and minimum points:

**Definition 1** *Let $A \subseteq X$ be a set of points such that:*

1. *$A$ is connected and maximal (with respect to the inclusion);*
2. *$\forall\ x \in A,\ f(x) = f(A) = const.$*

*Then we define $A$ as a:*

- *minimum set if $\exists\ \epsilon > 0 :\ \forall\ y \in X \setminus A, d(y,A) \leq \epsilon,\ f(y) > f(A)$;*
- *saddle set if it contains more than one element and $\forall \epsilon > 0\ \exists\ y_1, y_2 \in X \setminus A,\ d(y_i, A) \leq \epsilon,\ i = 1,2 : f(y_1) < f(A) < f(y_2)$;*
- *maximum set if $\exists\ \epsilon > 0 :\ \forall\ y \in X \setminus A, d(y,A) \leq \epsilon,\ f(y) < f(A)$*

*($d(x,A)$ denotes the distance of the point $x$ from the set $A$). A set which is a minimum, saddle or maximum set is called a stationary set.*

Now we are ready to substitute the original hypotheses with some weaker ones:

1. $f$ lipschitzian with a constant $L$ that we can assume, without loss of generality, greater than 1;
2. $X$ compact, convex and with non empty interior;
3. there exist a finite number of stationary sets.

We can finally introduce the following algorithm, which is inspired, but not equal to MLSL:

- at step $k$ generate a random point $X_k$ from the uniform distribution on $X$;
- a chain is defined as a sequence of points $x_i = X_{v_j},\ v_j < k$ starting from $x_0 = X_k$, with the following characteristics:

$$\begin{aligned} &\|x_i - x_{i-1}\| \leq \alpha_k,\ i \geq 1 \\ &f(x_i) \leq \min_{j<i} f(x_j) + \beta_k \end{aligned} \tag{1}$$

  where $\alpha_k$ is a parameter corresponding to the one present in MLSL and $\beta_k$ is a further parameter;

- we stop the sequence in the point $x_{final}$ if at least one of the following conditions is satisfied:
  1. another point satisfying (1) can not be found;
  2. from $x_{final}$ a local search has already been started;
  3. $x_{final}$ is at a distance not greater than $\alpha_k$ from an already detected minimum (in the case of minimum sets with more than one element by this we mean an already detected point which represents the whole set);

- in the first case we start a local search from $x_{final}$;
- we check a stopping rule and if it says of continuing we go to step $k+1$.

We outline here the differences with classical MLSL.

1. the "batchsize" $N$ in MLSL is set to 1;
2. we do not reconsider every point at any step to decide from which points starting a new local search (actually at any step we can not start more than one local search);
3. the value $\gamma$ of MLSL is set to 1;
4. the rule to decide whether to start or not a local search is different: we can possibly consider even not "too worse points" compared with the current one.

While the first three differences do not seem to be crucial, and we are confident that the same development which will follow can be applied also to algorithms which are more similar to classical MLSL, the fourth difference is very important. The main difference with MLSL is the fact that also some non monotonic chains of sampled points are considered when it has to be decided whether to start or not a local search, while MLSL considers only monotonic chains. This modification is inspired to non monotonic methods for local searches.

## Bayesian Heuristic Approach to Discrete and Global Optimization

Jonas Mockus, Audris Mockus and Linas Mockus

### Different Approaches to Numerical Techniques, Different Ways to Regard Heuristics: Possibilities and Limitations

General ideas

We call "heuristics" a formally expressed subjective expert knowledge about a problem of discrete and global optimization. The main goal is to describe different ways to apply heuristics. The different ways represent different degrees of formalization. We start from the traditional Bayesian Approach (BA), where heuristics are included by a choice of an a priori distribution, see [5] We extend the formal BA to semi-formal Bayesian Heuristic Approach (BHA), see [4], where heuristics might be included more flexibly. We finish by the description of informal Dynamic Visualization Approach (DVA), see [3]. Using DVA we may include heuristics directly by-passing formal mathematical framework.

All the theoretical results are applied to some real-life or test problems. All application examples illustrates some theoretical results. One of the real life application examples- the scheduling of batch operations- is described in details, see [7], because this single example illustrates most of the theoretical results.

Different Formalization Degrees

The maximal deviation is traditionally used while developing various numerical techniques. We call that a Minimax Approach (MMA) or the worst case analysis.

The advantage of MMA is the complete formalization. Everything is well defined, and we get the results with guarantee. The disadvantage is the high price we have to pay for the guarantee. Often the price is an algorithm of exponential complexity.

In this paper we consider an average deviation as a criterion when designing numerical optimization techniques and algorithms. We call that a Bayesian Approach (BA).

The disadvantage of BA is a degree of uncertainty while defining an a priori distribution on a set of problems to be optimized. Thus we get no guarantee. The advantage is the possibility to include some elements of the expert knowledge while defining an a priori distribution. By involving the expert knowledge we may tailor the algorithm to the specific problems. This way we may increase algorithms efficiency.

We may involve the expert knowledge by defining the a priori distribution on a set of randomized heuristic decision rules, too. We call this extension of traditional BA a Bayesian Heuristic Approach (BHA). Thus we get less formalization but more flexibility.

Both BA and BHA help the algorithm developers to include the expert knowledge by a regular mathematical framework.

The dynamic visualization helps the decision makers to include the expert knowledge directly. We denote that as a Dynamic Visualization Approach (DVA)

The objective of this paper is to discuss the possibilities and limitations of different approaches, different techniques of optimization, using heuristics. From a strictly formal BA, to a semi-formal BHA, to an informal DVA. Therefore, in addition to BA and BHA algorithms we discuss various visualization techniques considering the case studies of optimal decision making.

We start the description from the application of the BA to the continuous global optimization. Then we show how to extend the results to the BHA, namely, to the optimization of parameters of randomized heuristic techniques of global continuous and discrete optimization.

A new theoretical idea of this research is to define an a priori distribution on a set of randomized heuristic. The usual way is to define it on a set of functions to be minimized. The

definition of the a priori distribution on the set of heuristic decision rules helps to include the expert knowledge more flexibly and to speed up the search.

We show the advantages and disadvantages of BA and BHA applying those approaches in different problems of global and discrete optimization.

**Application List**

We apply BA to the following problems, see [5, 6], of global continuous optimization:

- modeling and yield maximization of electrical circuits
- optimization of the sock-absorber
- estimation of parameters of an immunological model
- estimation of parameters of bilinear time series
- estimation of parameters of fractionally integrated time series describing the exchange rates
- search for the equilibrium in a competitive economic model
- optimization of composite laminates
- minimization of molecule potential energy
- optimization of thermostable polymeric composition

We apply BHA to these discrete optimization problems, see [6]:

- knapsack
- flow-shop
- travelling salesman
- parameter grouping
- scheduling of batch operations.

We describe the dynamic visualization techniques which could be useful while solving the ill-defined optimization problems. We call an optimization problem "ill-defined" if we update the objective and the model during the optimization process. It means that we have to define the objective and the model interactively. Considering various real life optimization problems we see that many of them are "ill-defined".

We illustrate the dynamic visualization techniques by the following examples:

- smooth dynamic representation of data collected at fixed locations, see [1] (we want to minimize the deviations from a constant temperature over space and time)
- dynamic representation of observations in the form of averages over regions in space and time, exemplified by epidemiological data, see [2] (we are looking for spatial-temporal patterns that can suggest the most efficient ways of prevention and control)
- visual indexing, a dynamic index to a collection of 30,000 images, see [2] (we search for the "most interesting" subsets of images by visual inspection of the index)

We describe most of the examples as some illustrations that show how to apply the techniques developed in this research. We consider in detail one example. That is the scheduling of batch processes. A reason is that the batch scheduling can be considered either as a continuous or as a discrete optimization problem. Besides, the batch scheduling is an important and well known engineering problem so we may conveniently compare BHA with the results of other approaches.

We describe the software for UNIX and DOS platforms, see [6].

**References**


[1] W.F. Eddy and A. Mockus. An example of non-interactive dynamic graphics for manufacturing process data. *International Statistical Review*, 61:81–95, 1993.

[2] W.F. Eddy and A. Mockus. An example of the estimation and display of a smoothly varying function of time and space - the incidence of the disease mumps. *Journal of the American Society for Information Science*, 45(9):686–693, 1994.

[3] William Eddy and Audris Mockus. Dynamic visualization in modeling and optimization of ill defined problems, case studies and generalizations. *Journal of Global Optimization.* in print.

[4] A. Mockus, J. Mockus, and L. Mockus. Bayesian approach adapting stochastic and heuristic methods of global and discrete optimization. *INFORMATICA*, 5:167–122, 1994.

[5] J. Mockus. *Bayesian approach to global optimization.* Kluwer Academic Publishers, Dordrecht-London-Boston, 1989.

[6] Jonas Mockus. Application of Bayesian approach to numerical methods of global and stochastic optimization. *Journal of Global Optimization*, 4(4):347–366, June 1994.

[7] L. Mockus and G.V. Reklaitis. A new global optimization algorithm for batch process scheduling. *Journal of Global Optimization.* in print.

# Subdivision of Simplices Relative to a Cutting Plane with Applications in Concave Minimization and Volume Computation

Michael Nast

## Subdivision of simplices relative to a cutting plane

Let $S = [v^0, \ldots, v^n]$ be an $n$–simplex with vertex set $V(S) = \{v^0, \ldots, v^n\}$, $v^i \in \mathbb{R}^n$, $0 \leq i \leq n$, and for $a \in \mathbb{R}^n, b \in \mathbb{R}$ let $H = \{x \in \mathbb{R}^n : ax - b = 0\}$ be a hyperplane generating the half–spaces

$$H^{\leq} = \{x \in \mathbb{R}^n : ax - b \leq 0\}, \; H^{\geq} = \{x \in \mathbb{R}^n : ax - b \geq 0\}.$$

Define the corresponding open half–spaces $H^- := H^{\leq} \setminus H, \; H^+ := H^{\geq} \setminus H$,the vertex sets

$$V^-(S) := V(S) \cap H^-, V^+(S) := V(S) \cap H^+, V^=(S) := V(S) \cap H$$

and their cardinalities $n^-(S) := |V^-(S)|, \; n^+(S) := |V^+(S)|, \; n^=(S) := |V^=(S)|$.

**Definition 1** *The hyperplane $H$ is called irredundant for $S$, iff*

$$S \cap H^{\leq} \neq S \neq S \cap H^{\geq}.$$

**Lemma 1** *$H$ is irredundant for $S \Longleftrightarrow \min\{n^+(S), n^-(S)\} \geq 1$.*

Now let $S$ be an $n$–simplex, $H$ a hyperplane irredundant for $S$. Then $S$ can be written as $S = \operatorname{conv}(V^-(S) \cup V^+(S) \cup V^=(S))$ where $V^-(S) \neq \emptyset \neq V^+(S)$. For any pair of vertices $u \in V^-(S), \; v \in V^+(S)$ there exists a unique intersection point $h = e \cap H$ of the edge $e = [u, v]$ with the hyperplane $H$, which is given by

$$h = h(u, v, H) = \lambda u + (1 - \lambda)v,$$

where $\lambda = (av - b)/(av - au) \in (0, 1)$. The radial subdivision (see, e.g., [8], [10]) of the simplex $S$ with respect to the point $h$ yields the two $n$–simplices

$$S_1 = \operatorname{conv}(V(S) \setminus \{u\} \cup \{h\}) \text{ and } S_2 = \operatorname{conv}(V(S) \setminus \{v\} \cup \{h\})$$

satisfying $S_1 \cup S_2 = S$, $\operatorname{int} S_1 \cap \operatorname{int} S_2 = \emptyset$.

**Definition 2** *The radial subdivision of $S$ in $h$ is called a bisection of $S$ with respect to $u$, $v$ and $H$, or short a bisection with respect to $h$.*

The following **Algorithm 1** applies bisections on a given $n$ –simplex $S$, until a given hyperplane $H$ redundant or irredundant for $S$ is redundant for *every* generated subsimplex:

Iteration 0**:** Set $\mathbb{L} \leftarrow \{S\}, \; \mathcal{M}_{\mathcal{S}}^- \leftarrow \emptyset, \; \mathcal{M}_{\mathcal{S}}^+ \leftarrow \emptyset, \; \| \leftarrow \infty$.

Iteration $k$:

$k.1:$ If $\mathcal{L} = \emptyset$, then stop.

$k.2:$ Choose $S_k \in \mathcal{L}$ and set $\mathcal{L} \leftarrow \mathcal{L} \setminus \left\{\mathcal{S}_{\|}\right\}$.

$k.3:$ If $n^+(S_k) = 0$, set $\mathcal{M}_{\mathcal{S}}^- \leftarrow \mathcal{M}_{\mathcal{S}}^- \cup \left\{\mathcal{S}_{\|}\right\}$ and go to step $k.6$.

$k.4:$ If $n^-(S_k) = 0$, set $\mathcal{M}_{\mathcal{S}}^+ \leftarrow \mathcal{M}_{\mathcal{S}}^+ \cup \left\{\mathcal{S}_{\|}\right\}$ and go to step $k.6$.

$k.5:$ Choose $u \in V^-(S_k)$, $v \in V^+(S_k)$ and bisect $S_k$ with respect to $h(u,v,H)$ generating two $n$–simplices $S_{k_1}$, $S_{k_2}$, and set $\mathcal{L} \leftarrow \mathcal{L} \cup \left\{\mathcal{S}_{\|_\infty}, \mathcal{S}_{\|_\in}\right\}$.

$k.6:$ Set $k \leftarrow k+1$ and go to Iteration $k$.

**Proposition 1** *Let $n^+ := n^+(S)$, $n^- := n^-(S)$, $n^= := n^=(S)$. Then we have:*

**(i)** *Algorithm 1 stops after a finite number of iterations $i_S$. If $K := \left|\mathcal{M}_{\mathcal{S}}^- \cup \mathcal{M}_{\mathcal{S}}^+\right|$ is the number of simplices generated, then $i_S = 2K+1$.*

**(ii)** *$\mathcal{M}_{\mathcal{S}} := \mathcal{M}_{\mathcal{S}}^- \cup \mathcal{M}_{\mathcal{S}}^+$ forms a simplicial partition of $S$, $\mathcal{M}_{\mathcal{S}}^+$ forms a simplicial partition of $\overline{S \cap H^+}$ and $\mathcal{M}_{\mathcal{S}}^-$ forms a simplicial partition of $\overline{S \cap H^-}$.*

**(iii)** *$K = |\mathcal{M}_{\mathcal{S}}| = \binom{n^+ + n^-}{n^+}$, $K_+ := \left|\mathcal{M}_{\mathcal{S}}^+\right| = \binom{n^+ + n^- - 1}{n^-}$ and $K_- := \left|\mathcal{M}_{\mathcal{S}}^-\right| = \binom{n^+ + n^- - 1}{n^+}$, where we understand $\binom{n}{k} = 0$ for $k > n$.*

It is an easy task to derive from Algorithm 1 a small recursive procedure, which builds for a given pair $(S,H)$ the simplicial partition $\mathcal{M}_{\mathcal{S}}^-$ of the polytope $P^{\leq} := \overline{S \cap H^-}$. Regarding the number of simplices, we get the following result:

**Theorem 1** *Let $\mathcal{D} = \left\{\mathcal{S}_\rangle : \rangle \in \mathcal{I}\right\}$ be any simplicial partition of $P^{\leq}$. If $V(S_i) \subseteq V(P^{\leq})$ $\forall i \in I$, then $|I| = \left|\mathcal{M}_{\mathcal{S}}^-\right|$. Moreover, $\left|\mathcal{M}_{\mathcal{S}}^-\right|$ is the minimal cardinality of any simplicial partition of $P^{\leq}$.*

**Application to Concave Minimization**

In this section, we will derive a finite Branch and Bound algorithm for the problem

$$\min_{x \in P} f(x) \tag{1}$$

of minimizing a function $f : \mathbb{D} \to \mathbb{R}$ concave on a suitable set $\mathbb{D} \subseteq \mathbb{R}^n$, where $\dim P = n$ and $P \subseteq \mathbb{D}$, based on the subdivision procedure of Section . Assume

$$P = \{x \in \mathbb{R}^n : a_i^T x \leq b_i,\ i \in I\}\ , \tag{2}$$

where $I$ is a finite index set, and $a_i \in \mathbb{R}^n$, $b_i \in \mathbb{R}$, $i \in I$.

Lower Bounds

Let $S = [v^0, \ldots, v^n]$ be an $n$–simplex, $P$ a polytope given in the form (2) and let $f : S \longrightarrow \mathbb{R}$ be concave on $S$. Then to calculate a lower bound for $f^\star(S) := \min\{f(x) : x \in S \cap P\}$, we propose to compute either

$$\mu_1(S) := \min_{0 \leq j \leq n} f(v^j) \tag{3}$$

or, at the expense of solving a linear program,

$$\mu_2(S) := \min_{\lambda \in \mathbb{R}^{n+1}} \left\{ \sum_{j=0}^{n} \lambda_j f(v^j) : \lambda \geq 0,\ e^T\lambda = 1,\ a_i^T V \lambda \leq b_i,\ i \in I_S \right\}\ , \tag{4}$$

where in (4), $e := (1, \ldots, 1)^T \in R^{n+1}$, $V := (v^0, \ldots, v^n) \in R^{n \times n+1}$ is the matrix containing the vertices of $S$ a columns,

$$I_S := \left\{ i \in I : \exists\, j \in \{0, \ldots, n\} : a_i^T v^j > b_i \right\}\ , \tag{5}$$

and $\mu_2(S) = \infty$ if (4) has no feasible solution.

**Proposition 2** *Both $\mu_1(S)$ and $\mu_2(S)$ as defined above define valid lower bounds for $f^\star(S)$. If $\tilde{S}$ is an $n$–simplex containing $S$, then $\mu_j(S) \geq \mu_j(\tilde{S})$, $j = 1, 2$. Moreover, $\mu_2(S) \geq \mu_1(S)$ and, if $I_S = \emptyset$, then $\mu_1(S) = \mu_2(S) = f^\star(S)$.*

Upper Bounds

For any simplex $S$ generated in the course of solving problem (1), let $Q(S) := V(S) \cap P$ be the set of feasible vertices of $S$ . If using (4) for lower bounding, then add the feasible optimal solution obtained when calculating $\mu_2(S) < \infty$ to $Q(S)$. Obviously, the (possibly infinite) number $\gamma(S) := \min\{f(x) : x \in Q(S)\}$ yields an upper bound for $f^\star(S)$, and

$$\gamma := \min_{S \in \mathcal{S}} \gamma(S) \tag{6}$$

is an upper bound for $\min f(P)$, if $\mathcal{S}$ denotes the set of all generated simplices.

Subdivision of Simplices, Deletion by Infeasibility

Let $S = [v^0, \ldots, v^n]$ be an $n$–simplex with $\mu(S) < \gamma$, where $\mu(S)$ is given by either (3) or (4). Then $\mu(S) < \gamma(S)$ and $I_S \neq \emptyset$. Choose $i \in I_S$ and subdivide $S$ with respect to the cutting plane $H_i := \left\{x \in \mathbb{R}^n : a_i^T x - b_i = 0\right\}$ into the $n$–simplices contained in the set $\mathcal{M}_S^- = \mathcal{M}_S^-(\mathcal{H}_\rangle)$. Let $\mathcal{M}_S^- = \{\mathcal{S}_\infty, \ldots, \mathcal{S}_\ell\}$, where $\ell = K_-$ is given by Proposition ( 1)(iii). Then we propose to replace $S$ by $\{S_1, \ldots, S_\ell\}$. Note that, if for some $i_0 \in I_S$ one has $V(S) \subseteq H_{i_0}^\geq$, the corresponding set $\mathcal{M}_S^-$ is empty. In this case, we have $\ell = 0$, and we propose to eliminate $S$ from the set of simplices under consideration without any further subdivision. Applying this (implicit) deletion rule, one eliminates partition sets $S$ with $S \cap P \subseteq \partial P$, i.e. one cuts off at most randpoints of $P$.

Algorithm 2:

Iteration 0**:** Determine an initial $n$ –simplex $S_0 \supseteq P$, the lower bound $\mu(S_0)$ and the set $Q(S_0)$ . Set $Q_0 \leftarrow Q(S_0)$, $\gamma_0 \leftarrow \min\{f(x) : x \in Q_0\}$ and choose $y_0 \in Q_0$ satisfying $f(y_0) = \gamma_0$, if $Q_0 \neq \emptyset$. Set $\mathcal{P}_\prime \leftarrow \{\mathcal{S}_\prime\}$, $\mu_0 \leftarrow \mu(S_0)$, $k \leftarrow 1$.

Iteration $k$:

$k.1$ : If $\gamma_{k-1} = \mu_{k-1}$, then stop. ($y_{k-1}$ is an optimal solution to Problem (1) with optimal function value $\gamma_{k-1}$)

$k.2$ : Select $S_k \in \mathcal{P}_{\|-\infty}$ satisfying $\mu(S_k) = \mu_{k-1}$.

$k.3$ : Choose $i_k \in I_{S_k}$ and compute the set $\mathcal{M}_{\mathcal{S}_\|}^-$ with respect to the cutting plane $H_{i_k}$ (as described in ). Let $\ell := \left|\mathcal{M}_{\mathcal{S}_\|}^-\right|$, $\mathcal{M}_{\mathcal{S}_\|}^- = \left\{\mathcal{S}_{\|_\infty}, \ldots, \mathcal{S}_{\|_\ell}\right\}$. Compute the lower bounds $\mu(S_{k_j})$ for $1 \leq j \leq \ell$.

$k.4$ : Set $Q_k \leftarrow Q_{k-1} \cup \bigcup_{j=1}^{\ell} Q(S_{k_j})$, $\mathcal{P}_\| \leftarrow \mathcal{P}_{\|-\infty} \setminus \left\{\mathcal{S}_\|\right\} \cup \bigcup_{|=\infty}^{\ell} \left\{\mathcal{S}_{\|_|}\right\}$, $\gamma_k \leftarrow \min\{f(x) : x \in Q_k\}$ and $\mu_k \leftarrow \min\left\{\mu(S) : S \in \mathcal{P}_\|\right\}$. If $\gamma_k < \infty$, choose $y_k \in Q_k$ satisfying $f(y_k) = \gamma_k$.

$k.5$ : Set $\mathcal{P}_\| \leftarrow \mathcal{P}_\| \setminus \left\{\mathcal{S} \in \mathcal{P}_\| : \mu(\mathcal{S}) \geq \gamma_\|\right\}$. If $\mathcal{P}_\| = \emptyset$, set $\mu_k \leftarrow \gamma_k$.

$k.6$ : Set $k \leftarrow k + 1$ and go to Iteration $k$.

**Theorem 2** *In Problem (1), let $f$ be continuous on $P$, where the polytope $P$ is given in the form (2) with $|I| = m$. Then algorithm 2 stops in iteration $K + 1$ yielding an exact solution $y_K$ with $\mu_K = \gamma_K = f(y_K)$. An upper bound for the iteration steps is given by $K \leq \frac{M^m - 1}{M - 1}$, the total number $N$ of simplices generated satisfies $N \leq \frac{M^{m+1} - 1}{M - 1}$, and the maximal size of any partition $\mathcal{P}_\|$ is bounded by $\left|\mathcal{P}_\|\right| \leq M^{m-1} + M - 1$, where $M := \binom{n}{\lfloor \frac{n}{2} \rfloor}$.*

**Application to Volume Computation**

If $S = [v^0, \ldots, v^n]$ is an $n$–simplex, then the volume of $S$ can be computed by

$$\mathrm{vol}(S) = \frac{1}{n!} \left| \det \left( v^1 - v^0, \ldots, v^n - v^0 \right) \right| . \tag{7}$$

Therefore, perhaps the most natural approach to the problem of computing the volume of a polytope $P$ is to generate a simplicial partition $\mathcal{P}$ of $P$, what is often called a triangulation of $P$. Then compute the volumes of the individual $n$–simplices $S \in \mathcal{P}$ and add them up to find the volume of $P$. In fact, besides computing $V(S)$ for all $S \in \mathcal{P}$, the problem is how to find the triangulation $\mathcal{P}$ of a polytope. The following **Algorithm 3** does both tasks simultaneously:

Iteration 0: Determine an initial $n$ –simplex $S_0 \supseteq P$, and its volume $\mathrm{vol}\,(S_0)$ . Set $\mathcal{P}_I \leftarrow \{\mathcal{S}_I\}$ $k \leftarrow 1$.

Iteration $k$:

$k.1:$ If $I_S = \emptyset \ \forall S \in \mathcal{P}_{\|-\infty}$ then compute $\mathrm{vol}\,(P) = \sum_{S \in \mathcal{P}_{\|-\infty}} \mathrm{vol}\,(S)$ and stop.

$k.2:$ Select $S_k \in \mathcal{P}_{\|-\infty}$ satisfying $I_{S_k} \neq \emptyset$.

$k.3:$ Choose $i_k \in I_{S_k}$ and compute the set $\mathcal{M}^-_{\mathcal{S}_\|}$ with respect to the cutting plane $H_{i_k}$. Let $\mathcal{M}^-_{\mathcal{S}_\|} = \left\{ \mathcal{S}_{\|_\infty}, \ldots, \mathcal{S}_{\|_\ell} \right\}$. Derive the volumes $rmvol(S_{k_j})$ for $1 \leq j \leq \ell$ from $\mathrm{vol}(S_k)$. Set $\mathcal{P}_\| \leftarrow \mathcal{P}_{\|-\infty} \setminus \left\{ \mathcal{S}_\| \right\} \cup \bigcup_{|=\infty}^{\ell} \left\{ \mathcal{S}_{\|_|} \right\}$, $k \leftarrow k+1$ and go to Iteration $k$.

The crucial step is the computation of $\mathrm{vol}(S_{k_j})$ given $\mathrm{vol}(S_k)$ in step $k.3$. Here, this becomes a simple calculation which avoids the evaluation of (7). It can be incorporated into the subdivision procedure, since for every bisection of a simplex $S$ with respect to a point on an edge of $S$ it is possible to calculate the volumes of the generated subsimplices by a multiplication.

**References**


[1] Benson, H.P. and Sayin, S. (1994), A Finite Concave Minimization Algorithm Using Banch and Bound and Neighbor Generation, *Journal of Global Optimization*, 5, pp. 1–14

[2] Cohen, J. and Hickey, T.(1979), Two Algorithms for Determining Volumes of Convex Polyhedra, *Journal of the Association for Computing Machinery*, Vol. 26, Nr. 3, pp. 401–414

[3] Edelsbrunner, H. (1987), *Algorithms in Combinatorial Geometry*, Springer, New York

[4] Edelsbrunner, H. (1993), Geometric Algorithms, in *Handbook of Convex Geometry*, Vol. A, ed. by P.M. Gruber and J.M. Wills, North–Holland, Amsterdam, pp. 699–735

[5] Gritzmann, P. and Klee, V. (1993), On the Complexity of some Basic Problems in Computational Convexity II: Volume and mixed volumes, *Report Nr. 493*, University of Trier

[6] Grünbaum, B. (1967), *Convex Polytopes*, Wiley–Interscience, London

[7] Horst,R. and Thoai, N.V. (1988), Modification, Implementation and Comparison of Three Algorithms for Globally solving Concave Minimization Problems, *Computing*, Vol. 42, pp. 271–289

[8] Horst,R. and Tuy, H. (1993), *Global Optimization (Deterministic Approaches)*, 2nd Edition, Springer, Berlin

[9] Tam, B.T. and Ban, V.T. (1985), Minimization of a Concave Function Under Linear Constraints. *Ekonomika i Matematicheskie Metody* , 21, 709–714, (in Russian)

[10] Tuy, H. (1991), Effect of the Subdivision Strategy on Convergence and Efficiency of Some Global Optimization Algorithms, *Journal of Global Optimization*, Vol. 1, Nr. 1, pp. 23–36

[11] Von Hohenbalken, B. (1978), Least distance methods for the scheme of polytopes, *Mathematical Programming*, Vol. 15, pp. 1–11

[12] Von Hohenbalken, B. (1981), Finding simplicial subdivisions of polytopes, *Mathematical Programming*, Vol. 21, pp. 233–234

## NOP – a compact input format for nonlinear optimization problems

Arnold Neumaier

This paper defines a compact format NOP for specifying general constrained nonlinear optimization problems. The proposed format is a nonlinear analogue of an explicit representation of sparse matrices by means of index lists and values of the corresponding matrix entries. Thus the format abstracts from the meaning of the problem and hence does not allow names for variables or parameters, but it explicitly displays the internal structure of the problem. This is a very useful feature for global or large scale local optimization.

In contrast to the SIF input format proposed by Conn, Gould, and Toint [1] for their LANCELOT package, the amount of overhead in the formulation of smaller problems is very slight: For example, problems like Rosenbrock's function can be represented in a few lines in such a way that the least squares structure is visible in the representation. In general, partially group separable problems are as easy to code in NOP as in SIF format.

Together with an interface to GLOPT [4], a global constrained optimization code developed in Vienna, and with planned interfaces to the local optimization package MINOS (Murtagh & Saunders [3]) for large scale problems and to the modeling language AMPL (Fourer, Gay & Kernighan [2]) to allow the automatic structuring of input on a higher level, this is a promising tool for the formulation and solution of nonlinear optimization problems.

Each *NOP file* consists of a sequence of *records* describing a *constrained optimization problem* of the form

$$\begin{aligned} &\min \omega x_s \\ &\text{s.t. } x' \le x \le x'', \\ &\quad E_\nu(x), \quad \nu = 1, \dots, N, \end{aligned}$$

possibly with additional *integer or threshold constraints*. The bound constraints $x' \le x \le x''$ may have infinite bounds to allow for one-sided bounds and free variables.

The so-called *elements* $E_\nu(x)$ are constraints of one of the forms

$$\begin{aligned} &\sum_k f(a, b_k, x_{J_k}) \in [q], \\ &\sum_k f(a, b_k, x_{J_k}) + b = x_j, \end{aligned}$$

where $f$ is a so-called *element function*, $a, b_k, b$ are parameters or parameter vectors, $x_{J_k}$ is a subvector indexed by the index list $J_k$, and $[q]$ is a possibly unbounded interval, possibly restricted to integers or with zero adjoined. The contributions $f(a, b_k, x_{J_k})$ are referred to as the *pieces* of the element. (Elements containing a single piece only are, of course, permitted.)

The NOP format supports *block structure* and *multiobjective optimization* by allowing vector-valued components $x_i$ and element functions $f$.

### References

[1] A.R. Conn, N.I.M. Gould, and Ph. L. Toint, LANCELOT. A Fortran Package for large-scale nonlinear optimization. Springer, Berlin 1992.

[2] R. Fourer, D.M. Gay, B.W. Kernighan, AMPL. A modeling language for mathematical programming, Scientific Press, San Francisco 1993.

[3] B. A. Murtagh and M.A. Saunders, MINOS 5.1 user's guide, Tech. Report SOL 83-20R, Stanford Univ., Stanford, Calif. 1983, revised 1987.

[4] A. Neumaier, S. Dallwig and H. Schichl, GLOPT – a program for constrained global optimization, talk at this conference.

## Hemivariational Inequalities and Global Optimization. Numerical Search for the Optima

E.S. Mistakidis and Panagiotu D. Panagiotopoulos

Nonconvex nonsmooth energy functions lead to a new type of variational expressions, the hemivariational inequalities. They characterize all the solutions of the general type inclusion $0 \in \partial\Phi(u)$ where $\partial$ is the generalized gradient and $\Phi$ is a nonconvex and nonsmooth global (super)potential. We consider here hemivariational inequalities of the type: Find $u \in V$ such that

$$a(u, v-u) + \Phi(v) - \Phi(u) \geq (f, v-u) \forall v \in V,$$

where $a(.,.)$ is a bilinear function, $\Phi$ is nonconvex and nonsmooth energy function, $f \in V'$ is a linear function and $V$ is an appropriately defined Banach space. A method is proposed for the numerical treatment of the problem consisting in the replacement of the nonconvex problem by a sequence of quadratic energy problems. These subproblems are effectively treated by quadratic programming algorithms. Finally a method is presented for the classification of all solutions of the equilibrium equations which are e.g. local and global minima and saddle points. The method of the paper is illustrated by numerical applications.

### References

[1] P.D. Panagiotopoulos, Inequality Problems in Mechanics and Applications. Convex and Nonconvex Energy Functions (Birkhäuser, Boston-Basel, 1985); Russian translation (MIR, Moscow, 1989).

[2] P.D. Panagiotopoulos, Hemivariational inequalities. Applications in mechanics and engineering. Springer Verlag New York - Berlin 1993.

## Continuous Approaches to Discrete Optimization Problems

Panos M. Pardalos

Discrete (or combinatorial) optimization problems, that is, problems with a discrete feasible domain and/or a discrete domain objective function, model a large spectrum of applications in computer science, operations research and engineering.

Solution methods for discrete optimization problems can be classified into combinatorial and continuous approaches. A typical combinatorial approach generates a sequence of states, which represent a partial solution, drawn from a discrete finite set. Continuous approaches for solving discrete optimization problems are based on different equivalent characterizations in a continuous space. These characterizations include equivalent continuous formulations, or continuous relaxations, that is, embeddings of the discrete domain in a larger continuous space

There are many ways to formulate discrete problems as equivalent continuous problems or to embed the discrete feasible domain in a larger continuous space (relaxation). The surprising variety of continuous approaches reveal interesting theoretical properties which can be explored to develop new algorithms for computing (sub)optimal solutions to discrete optimization problems.

We are going to discuss continuous approaches to several discrete problems, including the maximum clique problem, graph coloring and the satisfiability problem.

# Optimal Renewal Policy for Slowly Degrading Systems

András Pfening and Miklós Telek

Preventive maintenance is considered to be one of the key strategies to increase system availability and performance. In general, preventive maintenance consists of periodically stopping the system, and restarting it after doing proper maintenance, that reduces the probability of failure and increases system performance. Some cost is unavoidable since the system has to be stopped and it is unavailable during the maintenance. The arising research problem is to find the optimal maintenance policy, the policy that minimizes a certain cost function. While preventive maintenance concepts have been usually applied to mechanical systems, they can also be effectively applied to the field of software reliability. Thus fault tolerant software can became an effective alternative to virtually impossible fault-free software.

*Huang et. al.* have suggested a technique which is preventive in nature. It involves periodic maintenance of the software so as to prevent crash failures, they call it *Software Rejuvenation* [2]. *Garg et. al.* [1] have improved Huang's model by allowing deterministic regeneration time and provided an optimal rejuvenation policy for the studied class of systems, regarding crash failures.

But monitoring real applications showed that software "ages" when it is run, i.e. its performance decreases and the probability of failure increases instead of suffering crash failures. Memory bloating, unreleased file-locks, data corruption are the typical causes of slow degradation which could lead to crash failure as well if it is not taken care of. Software rejuvenation involves periodically stopping the system, cleaning up, and restarting it from its peak performance level. This "renewal" of software prevents (or in the least postpones) a crash failure and increases performance.

## Problem Statement

In the paper we address the problem of soft failures of a server software, i.e. the preventive maintenance is done to increase system performance, while crash failures are not considered. The server software serves jobs arriving to the system with slowly degrading performance. The problem is to determine the rejuvenation time interval, if the probability distribution of the interarrival times and the service times are known. It should be performed to optimize the cost of the rejuvenation, consisting of the costs paid for the lost jobs that arrived during the rejuvenation and costs paid for the jobs that were queued waiting for service when rejuvenation started, since these jobs are lost. We also take into account the run time of the system, since the same cost paid in case of a longer run is preferred.

In the paper two systems are analyzed, they differ in the applied queuing policy. The first studied system does not allow buffer overflow (we will refer to it as *no buffer overflow case*) by stopping and rejuvenating the system when the buffer is full and a new job arrives to the system. It may be the case when the buffer is supposed to be large enough to accommodate all the arriving jobs, or when the system operator does not want to loose jobs during the system operation. The second scenario (*buffer overflow case*) allows buffer overflow during operation, however the cost caused by the lost jobs must be reflected in the overall cost function.

## No Buffer Overflow Case

In the first studied system Poisson arrival process of jobs is assumed with parameter $\lambda$, and the service rate is decreasing with time: $\lim_{t\to\infty} \mu(t) = \mu_\infty$ and $\mu(t) \geq \mu(t+\Delta)$ if $\Delta \geq 0$. With the above assumptions the system states can be described by two variables for the no buffer overflow case, the time spent since the last rejuvenation, and the number of jobs waiting for service in the system. Our goal is to find a *policy*, that determines for each state of the system whether to stop the system and rejuvenate it, or to continue the service.

If we discretize the system variable that describes the time spent since the last rejuvenation, we arrive to a *Markov Decision Process (MDP)*. In the framework of MDP theory an algorithm (we will refer to it as *MDP algorithm*) is provided to approximate the *optimal policy*, the policy that yields the minimum expected cost [3].

In the paper we derive a condition for the cost function (the function that defines the cost of stopping the system in the current state) when the optimal policy can be approximated by applying the results for MDPs.

In addition to the general results we investigated the special case when the cost function has the simple form of

$$C(b, t, stop) = \frac{b + \lambda T_R}{t + T_R},$$

where $b$ is the number of jobs waiting in the queue, $t$ is the time spent since the last rejuvenation, and $T_R$ is the rejuvenation time. If Poisson arrivals are assumed, $\lambda T_R$ is the mean value of number of jobs arrived during the rejuvenation period, i.e. this cost function is simply the average number of lost jobs per time unit. We show that the MDP algorithm converges to the optimal policy. Two simple rules are also derived determining a buffer content dependent upper limit of time $t_{UB}(b)$, for which if $t \leq t_{UB}(b)$, the optimal policy will continue the service, and a buffer independent lower time limit $t_{LB}$, for which if $t > t_{LB}$ the optimal policy will choose system rejuvenation ($t_{LB} \geq t_{UB}(b)$).

In practical applications if the cost differences in the time interval $(t_{UB}(b), t_{LB})$ are not significant, it is not necessary to use the MDP algorithm. The applied policy can be simply based on the derived upper and lower time bounds.

**Buffer Overflow Case**

In the buffer overflow case the system can be characterized by three variables, since we have to consider the number of lost jobs to find the minimal cost decision.

Similar analysis steps are accomplished like in the no buffer overflow case. The concerning MDP algorithm is defined, and conditions for the cost function are derived to assure the convergence to the optimal policy.

The analyzed simple cost function is modified to

$$C(b, t, L, stop) = \frac{b + \lambda T_R + L}{t + T_R},$$

where $L$ r.v. denotes the number of lost jobs in $[0, t]$. The convergence of the MDP algorithm is not guaranteed, however a simple rule for $t_{UB}(b, L)$ can be derived also for this case.

**Numerical Example**

The theoretical results are illustrated by numerical examples. The MDP algorithm is implemented in *Mathematica 2.0* for the no buffer overflow case, and the results of the MDP algorithm and the derived simple rules are compared in some figures.

**References**


[1] S. Garg, A. Puliafito, M. Telek, and K. S. Trivedi. Analysis of software rejuvenation using markov regenerative stochastic petri net. In *Sixth International Symposium on Software Reliability Engineering'95 Toulouse, France*, 1995.

[2] Y. Huang, C. Kintala, N. Kolettis, and N. D. Fulton. Software rejuvenation: Analysis, module and applications. In *To Appear in Proc. Of 25Th Symposium on Fault Tolerant Computing*, June 1995.

[3] S. M. Ross. *Applied Probability Models with Optimization Applications.* Dover Publications, Inc., New York, 1992.

## CGU: A Global Optimization Algorithm for Protein Structure Prediction

Andrew T. Phillips, J. Ben Rosen and Ken A. Dill

This talk will discuss the CGU (convex global underestimator) algorithm which is a global optimization algorithm designed to predict the native state of protein molecules, given their primary amino acid sequences. Computing the native state of a protein requires: (1) a suitable potential energy function, and (2) a conformational search method that can find the global minimum of the energy function. Neither of these problems is yet solved. Finding the global minimum is difficult because reasonable physical energy functions typically have many local minima, the number of which grows exponentially with polymer size. We have a reasonable energy function, described below, for which we know that the bottleneck is the search strategy rather than the energy function. In this talk, we focus on the global optimization rather than on the potential function problem.

We have developed a global optimization method that we have already tested on finding global optima of homopolymer conformations. For that purpose we have used the energy function:

$$U = U_b + U_{ba} + U_t + U_{nl}$$

where $U_b$ = bond stretching energy, $U_{ba}$ = bond angle energy, $U_t$ = torsion angle dependent energy, $U_{nl}$ = non-local pairwise interaction energy. The first three terms are explicitly given in terms of internal coordinates, but $U_{nl}$ is given in terms of the pairwise distances between amino acids $a_i$ and $a_j$. A major part of the computational effort in the energy minimization is devoted to computing $U_{nl}$, and its derivatives, in terms of the internal coordinates. We used a Lennard-Jones potential for $U_{nl}$ in our initial investigation of the new global optimization algorithm [1].

Our algorithm for computing the global minimum of a polymer potential energy function is based on the iterative use of global underestimators to localize the search in the region of the global minimum. This new method, developed and implemented by J. Ben Rosen and Andrew T. Phillips [1], is designed to fit all known local minima with a convex function which underestimates all of them, but which differs from them by the minimum possible amount in the discrete $L_1$ norm. The minimum of this underestimator is used to predict the global minimum for the function, allowing a localized conformer search to be performed based on the predicted minimum. A new set of conformers generated by the localized search then serves as a basis for another convex underestimation. This new algorithm combines some aspects of both quasi-Newton methods and genetic algorithms. The effectiveness of this algorithm has been shown by its ability to compute global minima for n-mer chains, as summarized next.

The use of an underestimating function allows the translation of a very complex function into a simple underestimator. If the underestimator is well suited to the problem (i.e. provides accurate predictions for the global minimum), the global solution can be found in very few iterations. This new technique has already been successfully used to determine the three dimensional molecular structures for n-mer hydrocarbon homopolymer chains with as many as $n = 30$ beads. While there are estimated to be $O(3^n)$ local minima for a chain of length $n$, this method requires only $O(n^4)$ computing time on average. In fact, for all hydrocarbon chains of length $n \leq 12$, the predicted structures obtained by applying this algorithm have been confirmed as the global energy minimizers of the potential function used. Moreover, it has also been shown that the global energy minimum values, as a function of chain length $n$, lie on a smooth curve that can be approximated very closely by a simple concave quadratic function. This important property of the very complicated function $U$, which seems not to have been observed previously, permits estimation of the global minimum energy for larger molecular chains, and can also be used to accelerate the global minimization algorithm.

While the algorithm developed has been shown to require $O(n^4)$ time on average, it is also highly amenable to massively parallel computation. In fact, for homopolymers ranging in size from $n = 14$ through $n = 22$, the results show that the algorithm is scalable with increased problem size using a parallel heterogeneous computer system involving the Cray Y-MP C90 and 128 node Cray T3D at the University of Minnesota Supercomputer Institute. Furthermore, it is expected that as the problem size increases, this new method will continue to make full use of the computational power of this system. The solution to the 22 bead hydrocarbon problem required approximately 36 minutes (wall clock time using only 8 processors).

Based on the success of the homopolymer tests described above, we have already begun the process of applying the CGU method to more realistic protein-like models. This same iterative global underestimating algorithm can be used with a potential function corresponding to the more complicated chain of residues in a realistic model of proteins. Ken Dill's group at the University of California, San Francisco, has developed simple lattice models to better understand the physics of protein folding ([2],[3],[4]). These models are based on evidence that the hydrophobic interaction is the dominant force in protein folding, and that, to first-order approximation, a protein can be modeled as a specific sequence of hydrophobic (H) and hydrophilic (P) monomers. While such simple models show many of the secondary and tertiary structural features of real folded proteins (including a-helices and b-sheets), the restriction to lattices is too artificial to allow realistic folding, but the Dill group has recently developed a more realistic off-lattice model.

An even more realistic, yet still simplified, potential function has been developed by S. Sun [5]. It is an off-lattice model in which the mainchain is represented by spherical beads centered at C alpha carbons, and sidechains are also single spherical beads. In that work, interactions are modelled based on protein database derived potentials. However Sun, now working in Dill's group, has developed a much simpler potential function that has been used in a restricted search problem. Their new potential function is simple but physical, and does not derive from a protein database. Sun and Dill [6] have shown that the use of the Sun chain representation, along with only a hydrogen bond energy term and a single hydrophobic interaction energy term, when searched using simulated annealing and genetic algorithms, reproduces relatively well the chain folds of 10 small proteins, provided helices and strands are fixed in their known native conformations (to limit the conformational searching time). Remarkably, despite the simplicity of the potential function, the limiting problem is again found to be the conformational search strategies, not the potential function: the energies of the true known native structures are lower than the best computed structures in 9 out of 10 cases. Hence the aim of this joint work is to apply the CGU global optimization strategy can do better, using the same potential function. Results from this most recent work will be presented.

**References**


[1] A.T. Phillips, J.B. Rosen and V.H. Walke: Molecular Structure Determination by Convex Global Underestimation of Local Energy Minima, Journal of Global Optimization (in press)

[2] H.S. Chan and K.A. Dill, The Protein Folding Problem, Physics Today, 1993, 24-32

[3] K.A. Dill, Dominant Forces in Protein Folding, Biochemistry, 1990, 29, 7133-7155

[4] K. Yue and K.A. Dill, The Forces of Tertiary Structural Organization in Globular Proteins, Proceedings of National Academy of Science (in Press)

[5] S. Sun, Reduced Representation Model of Protein Structure Prediction: Statistical Potential and Genetic Algorithms, Protein Science, 1993, 2, 762-785

[6] S.Sun and K.A. Dill, A Simple Protein Folding Algorithm Using Binary Code and Secondary Structure Constraints, Protein Engineering (submitted)

# LGO: An Implementation of a Lipschitzian Global Optimization Procedure

János D. Pintér

Decision problems are frequently modelled by optimizing the value of an objective function under stated feasibility constraints. Specifically, we shall consider the following global optimization problem (GOP)

$$\min f(x) \quad \text{subject to} \quad x \in D \subset \mathbb{R}^n. \tag{1}$$

It is supposed that in (1) $f : D \to R$ is a continuous function, and $D$ is a bounded, robust subset ('body') in the Euclidean $n$-space $\mathbb{R}^n$. In addition, the Lipschitz-continuity of $f$ on $D$ will also be postulated, when appropriate.

The above assumptions define a fairly general class of optimization problems. Essentially, they reflect a decision paradigm in which a rather vaguely defined, possibly quite 'large' and/or 'complicated' search region is given on which a (potentially) multiextremal—possibly 'black box'—objective function is minimized. For reasons of analytical tractability, it will also be supposed that the (non-empty) set of global solutions $X^* \subset D$ is, at most, countable. We shall apply the notation $f^* = f(x^*)$, for $x^* \in X^*$.

To solve (1), a general family of branch-and-bound type adaptive partition strategies can be introduced: for comprehensive discussions, consult, e.g., Horst and Tuy (1990), Pintér (1992a, 1995), Hansen and Jaumard (1995), with extensive lists of additional references. Necessary and sufficient convergence conditions can be established: these lead to a unified view of numerous GO algorithms, permitting their straightforward generalizations and various extensions, to handle specific cases of the general GOP (1).

An implementation of LGO, a Lipschitzian global optimization procedure (integrated with other solvers) is discussed briefly below. PC and workstation related experience, numerical test results and several applications are highlighted. Application areas include, among others, the following (Pintér, 1992b, 1995):

- general (Lipschitzian) nonlinear approximation
- systems of nonlinear equations and inequalities
- calibration (parametrization) of descriptive system models
- data classification
- generic hierarchical configuration design
- aggregation of negotiated expert opinions
- product/mixture design
- optimized design/operation of 'black box' (engineering, environmental, etc.) systems.

## The LGO Program System

Scope of Application

The LGO program system serves for finding—that is, numerically approximating—the best solution, or (theoretically) all solutions, of the possibly multiextremal optimization problem in its standardized, box-constrained form:

$$\min_{a \le x \le b} f(x). \tag{2}$$

In (2) $a$, $x$, $b$, are finite $n$-vectors, and $f$ is Lipschitz-continuous on $[a, b]$. Note that Lipschitz-continuity is an obvious 'minimal' requirement, in order to assure a guaranteed approximation of the optimal objective function value $f^*$, on the basis of a finite set of sample points from $D$. Notwithstanding, GOPs having a purely continuous structure are also of practical interest, and can be numerically handled within the framework of LGO.

Let us observe that a significantly more general class of (continuous or Lipschitz) GOPs of the form

$$\min \; f_0(x) \quad f_i(x) \leq 0, \quad i = 1, \ldots, m \quad a \leq x \leq b \tag{3}$$

(in which $f_0 := f$, $f_i$, $i = 0, \ldots, m$ are continuous or Lipschitz) can be numerically approximated in the form (2), following a penalty transformation based incorporation of the constraints $f_i$, $i = 1, \ldots, m$ into the objective function. (Consult, e.g., Fletcher's (1983) review on such techniques.) The penalty multipliers associated with the constraints can be adaptively chosen, to enforce the (approximate) feasibility and optimality of the solution found.

Other solver options to handle the general GOP (3)—in the framework of LGO—are under elaboration.

Solution Methodology

The current program system includes—in an integrated fashion—the following optimization procedure options:

- Lipschitz global optimization by adaptive partition and search (LGO)
- pure random search on the interval [a,b] (RS)
- local search (Fletcher-Reeves-Polak-Ribière method, FRPR)
- local search (Powell-Brent method, PB).

A few comments related to these solvers are in order. As known, RS is a very simple, 'folklore' approach to solve GOPs. In the present context, it can be applied, for instance, by novice users who want to explore the feasible set, and/or as a preliminary search phase for possibly reducing the initially chosen search interval. The Lipschitzian solver is far more sophisticated, and—enabled with proper parametrization—it should provide a reasonably close approximation of the global solution(s), before the LGO system is switched to local search (solution refinement). The classical local optimization approaches FRPR and PB (for a more recent practical discussion, consult Press, Teukolsky, Vetterling and Flannery, 1992) have been modified for this implementation. Further implementation details are discussed by Pintér (1995); additional options to solve (2) are under development.

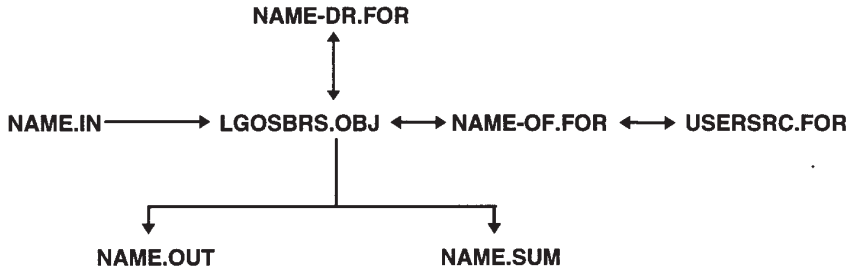
**Structure of LGO Application Programs**

The following abbreviations will be used (reflecting a Fortran environment on PCs):

- NAME – name of LGO application (given by user)
- NAME-DR.FOR – source code of main driver (prepared/adapted by user)
- NAME-OF.FOR – source code of objective function $f$ (by user)
- NAME.IN – input parameter file (by user)
- USERSRC.FOR – additional source code (optional, by user)
- LGOSBRS.OBJ – object file of LGO program system

- NAME.OUT – more detailed output file generated by LGO
- NAME.SUM – summary output file generated by LGO.

Commented templates of NAME-DR.FOR, NAME-OF.FOR and NAME.IN are provided, to assist users. The principal structure of LGO application program systems is shown below.




Observe that USERSRC can also be given as an object file, hence enabling the operation of LGO on true 'black box' applications.

### Current System Requirements and Problem Size Limitations

Hardware and Software Requirements

PC Environment

- IBM PS/2 Intel 386 & 387, 486 or Pentium processor based computer (or compatible)
- minimum 4 Mbyte (recommended: 8 Mbyte, or more) RAM
- appropriate hard drive space (when running LGO in virtual memory mode)
- professional Fortran development environment
- appropriate memory management tools.

Workstation Environment

No special restrictions (most systems, on which a professional Fortran is installed, are appropriate). Installation of LGO in other environments and programming languages is also possible.

Problem Size Limitations

- The present (explicitly declared) array structure in LGO supports the formulation and solution of GOPs up to 50 variables.
- Up to 10000 (currently active) partition subintervals can be simultaneously present in the LGO solution mode.

Note that both of these limitations can be easily relaxed—if necessary—but memory requirements and computational times will correspondingly increase.

### Using LGO in an Interactive Environment

The current PC version of LGO can be activated and used in an interactive fashion, using a menu-based interface under DOS. (A Windows version is also under development; on workstations X-Windows readily provides an interactive multitasking environment.) The application menu includes the following integrated options:

- introduction (general LGO information)

- demonstration program information
- edit application files
- compile LGO application programs
- run LGO applications
- view optimization output files
- view graphical summary of results
- quit (return to operating system).

### Numerical Experience and Applications

Test problems of realistic complexity (up to 50 variables) have been solved using LGO; a few examples are presented in Pintér (1995). Additionally, problems in numerical analysis, industrial and environmental engineering—having one to over 60 variables—have been solved, using different versions of LGO. For details on several more recent numerical studies and applications, consult, e.g., Pintér (1990a,b, 1991), Pintér and Pesti (1991), Hendrix and Pintér (1991), Pintér (1992b), Csendes and Pintér (1993), Van der Molen and Pintér (1993), Finley, Pintér and Satish (1994), Pintér, Fels, Lycon, Meeuwig and Meeuwig (1995), Stortelder and Pintér (1995). Numerous further prospective applications are collected in Pintér (1995).

Acknowledgments

This research has been supported by the Hungarian Scientific Research Fund (OTKA) and by the Dutch Technology Foundation (STW).

### References

[1] Csendes, T. and Pintér, J. (1993). The impact of accelerating tools on the interval subdivision algorithm for global optimization. *European Journal of Operational Research* 65, 314–320.

[2] Finley, J.R., Pintér, J. and Satish, M.G. (1994). Aquifer model calibration applying global optimization. Working Paper 94-05, Department of Industrial Engineering, Technical University of Nova Scotia, Halifax.

[3] Fletcher, R. (1983). Penalty functions. In: *Mathematical Programming: The State of the Art,* (ed. A. Bachem, M. Grötschel and B. Korte), pp. 87–114. Springer, Berlin.

[4] Hansen, P. and Jaumard, B. (1995). Lipschitz optimization. In: *Handbook of Global Optimization,* (ed. R. Horst and P.M. Pardalos), pp. 407–493. Kluwer Academic Publishers, Dordrecht.

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

[6] Horst, R. and Tuy, H. (1990). *Global Optimization – Deterministic Approaches.* Springer, Berlin. (2nd ed., 1993).

[7] Pintér, J. (1990a). Globally optimized calibration of environmental models. *Annals of Operations Research* 25, 211–22.

[8] Pintér, J. (1990b). Solving nonlinear equations via global partition and search: Some experimental results. *Computing* 43, 309–323.

[9] Pintér, J. (1991). Stochastic modelling and optimization for environmental management. *Annals of Operations Research* 31, 527–544.

[10] Pintér, J. and Pesti, G. (1991). Set partition by globally optimized cluster seed points. *European Journal of Operational Research* 51, 127–135.

[11] Pintér, J. (1992a). Convergence qualification of partition algorithms in global optimization. *Mathematical Programming* 56, 343–360.

[12] Pintér, J. (1992b). Lipschitzian global optimization: Some prospective applications. In: *Recent Advances in Global Optimization,* (ed. C.A. Floudas and P.M. Pardalos), pp. 399–432. Princeton University Press.

[13] Pintér, J. (1995). *Global Optimization in Action* (*Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications*), Kluwer Academic Publishers, Dordrecht, 1995.

[14] Pintér, J., Fels, M., Lycon, D.S., Meeuwig, D.J. and Meeuwig, J.W. (1995). An intelligent decision support system for assisting industrial wastewater management. *Annals of Operations Research* 58, 455–477.

[15] Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B. (1992). *Numerical Recipes in FORTRAN: The Art of Scientific Computing.* Cambridge University Press.

[16] Stortelder, W.J.H. and Pintér, J. (1995). Numerical approximation of elliptic Fekete point sets: A global optimization approach. (In preparation.)

[17] Van der Molen, D.T. and Pintér, J. (1993). Environmental model calibration under different problem specifications: An application to the model SED. *Ecological Modelling* 68, 1–19.

# Numerical Approximation of Elliptic Fekete Point Sets: A Global Optimization Approach

Walter J.H. Stortelder and János D. Pintér

The objective of this work is to provide a numerical approximation of elliptic Fekete point sets, applying Lipschitz global optimization. This problem is of obvious practical interest in scientific modelling; consult, for instance, [1, 3, 4].

Let us consider an $n-$tuple $x = (x_1, \ldots, x_n)$ in which $x_i \in \mathbb{R}^3$; we are interested in the global maximum of

$$f_n(x) = \prod_{1 \le i < j \le n} \| x_i - x_j \|, \quad \text{s.t.} \quad \| x_i \| = 1, \;\; i \in \{1, \ldots, n\}, \tag{1}$$

in which $\| \cdot \|$ denotes the Euclidean norm. It is known that problem (1) has a non-polynomially increasing number of local optima and saddle points; its analytical solution is unknown, and it apparently poses a nontrivial numerical challenge. The points $x_1, \ldots, x_n$ which give the global maximum of (1) are called the elliptic Fekete points of order $n$.

In order to solve this problem efficiently, first we will discuss a straightforward transformation to spherical coordinates. After briefly discussing some analytical results concerning the problem, we shall apply the Lipschitzian global optimization strategy LGO (consult [2]) to provide numerical approximations to elliptic Fekete point sets. LGO is a combination of Lipschitzian adaptive search, pure random search, and gradient-free local optimization.

*Acknowledgments:* This research has been supported by the Dutch Technology Foundation (STW) and the Hungarian Scientific Research Fund (OTKA).

## References

[1] P.M. Pardalos, An open global optimization problem on the unit sphere, *Journal of Global Optimization,* 6, 1995, p. 213.

[2] J. Pintér, *Global Optimization in Action* (*Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications*), Kluwer Academic Publishers, Dordrecht, 1995.

[3] M. Schub and S. Smale, Complexity of Bezout's theorem. III: Condition number and packing, *Journal of Complexity,* 9, 1993, 4–14.

[4] M. Tsuji, *Potential Theory in Modern Function Theory,* Maruzen Co., Tokyo, 1959.

# Reducing the Problem of Organization Structure Adaptation to Optimization Problem in Boolean Space

Olga Yu. Polyakova

The situation which is considered is: there is a hierarchical structure $S$ which describe the nature and the state of production system. When hindrances influence on system or the environment of the system have changed the structure of system must be changed too. So the structure adaptation problem become the actual one.

Let the structure $S$ is described by such set of parameters:

$$S = (X, Y, F_X, F_Y).$$

Here, $X$ is the set of structure elements, $Y$ is the set of connections between elements, $F_X$ is the set of elements' characteristics, $F_Y$ is the set of connections' characteristics.

The problem formulation of structure adaptation problem considered in some literature is based on the set of possible variants of structure and possibility of looking over all those variants. However such approach isn't expedient one for production system structure because of large dimension of the set of elements. It doesn't allow to synthesise an effective algorithm for suluting this problem.

From the other side the clustering algorithms which are used for synthesis of organization structure of production system allow to solve a large-dimensioned problem. Minimum of clusters number (when a threshold value is fixed) is the criterion for structure synthesis problem. This criterion describe management characteristics of structure, not it's stability characteristics.

To study the stability quality of synthesized structure the stability function is used. It may be used as a criterion for the synthesis problem or as an additional criterion. However the stability function describe static characteristics of structure and tell nothing about it's potential changing possibilities when environment is changing.

Another stability function can be proposed as a criterion which allow to study the dynamic stability of structure. Let assumpt that the single-elements cluster is absolutely stable one. Such partitioning may be considered for each problem when no another criterion used. Let define the freedom degree of cluster as a value which depend on number of elements included in cluster, stability of each element, degree of connections between elements and cluster's diameter.

$$F_l = \frac{N_l}{N} \left[ \sum_{i,j=1}^{N_l} u_{ij} \right]^{-1} diam(Q_l), \tag{1}$$

where

$$u_{ij} = \begin{cases} \text{degree of connection between elements } i,j \\ \text{stability of the element } i \end{cases}$$

$N_l$ is the number of elements included in cluster $Q_l$, $N$ is the number of all elements.

Let for absolutely stable partitioning stability function (II) is equal 1 and for any other partitioning stability of it's cluster:

$$U_l = \frac{1}{F_l}, \ l = 1, ..., M$$

$M$ is the number of clusters. (We suppose that all elements are in unit cube).

When hierarchical structure is considered the beginning elements for next level synthesis are clusters of the previous level.

Now the synthesis problem for the organisation structure of production system can be represent as a two-criterion optimization problem:

$$\begin{aligned} U(S) &\to \max \\ dim(X) &\to \min \end{aligned} \tag{2}$$

with restrictions:

$$\begin{aligned} & X = X_0^0 \cup X_1^0 \cup ... \cup X_n^m, \\ & X_i^k \cap X_j^k = \emptyset, \\ & \textstyle\sum_{i=1}^{N_j} x_{i,j}^k = 1, \\ & x_{ij} = \begin{cases} 1, & X_i^k \in X_j^{k+1} \\ 0 \end{cases} \end{aligned} \tag{3}$$

$m$ is the number of hierarchy level. The stability function (II) $U(S)$ can be represented in the form:

$$U(S) = \frac{1}{m-1} \left[ \sum_{k=0}^{m-1} \frac{1}{N_k} \left[ \sum_{j=1}^{N_{k+1}} \sum_{j=1}^{N_k} u_i^k x_{ij}^k \right] \right] \tag{4}$$

The complexity of using this function as a criterion is that neither number of hierarchy level no number of obtained clusters are known. However the nature of problem let suppose that the function has a global maximum when it is considered in boolean space $B_n$. The dimension of the space is unknown for the synthesis problem but we consider a restriction for the structure adaptation problem

$$n < dim(X)$$

and take $n = dim(X)$ as a beginning value.

The structure adaptation problem is transformed to the form

$$U(S^*) \to \max_{B_n}$$

$$dim(S^*) \to \min_{B_n} \tag{5}$$

with restrictions

$$U(S^*) > U(S) \tag{6}$$

Restriction (6) is like restriction (3)

$$\| S^* - S \| < \delta \tag{7}$$

$S$ is the known structure, $S^*$ - unknown structure, $\delta$ - the assumed measure of difference between structures. It is suppose that

$$dim(X_0 \cap X_0^*) \gg dim(X_0 \triangle X_0^*)$$

in other case the solving adaptation problem is unexpedient.

The speed of solving adaptation problem will depend on choice of beginning point. The beginning vector in boolean space must ensure the condition (7) and let save a part of previous structure. Such vector is easy found.

The synthesis of algorithm is based on the assumption about existence of global maximum and the one-to-one correspondence between some components of vector from boolean space which describe a new structure.

An algorithm of soluting structure adaptation problem will allow to get a new structure with the more little expenditures then it is need for solving production system synthesis problem because of saving a part of old structure.

# A Multistart Linkage Algorithm Using First Derivatives

Chris J. Price

This paper describes an optimization algorithm for estimating various acoustic parameters of an ocean environment. The parameter estimation proceeds as follows: a continuous wave single frequency source in the ocean generates a sound field. This field is measured at a number of points in the ocean. A normal mode model (see eg. [1]) incorporating various sea floor parameters such as density, sound speed, and attenuation is used to predict the measured field values. The parameters are chosen to minimise the 2–norm of the difference between the measured and predicted field values. It is possible to modify the normal mode model to calculate the objective function and its gradient for approximately twice the computational cost of calculating the objective function alone. The following algorithm is designed to exploit the cheap gradient information, and for convenience is recast in the following form: maximize $f(x)$, over $x \in \mathcal{F}$, where $\mathcal{F} \subseteq R^n$. Herein $f \in C^2$, and $\mathcal{F}$ is of the form $\ell \leq x \leq u$.

The algorithm is a variant of multistart, using a clustering process called Stochastic Process Single Linkage (SPSL) which is similar to Multi-Level Single Linkage (MLSL) [3]. SPSL uses a stochastic model together with the gradient of the objective function to assign a reliability value to each link. Unlike MLSL, SPSL does not put a link between every pair of points that are sufficiently close together. This suggests that SPSL's performance will not degrade when used with non-uniform distributions of sample points. An adaptive method for generating sample points is described next, followed by a description of SPSL.

The sample points are generated in batches. The first batch consists of random or quasi-random points. Points in subsequent batches are generated by perturbing existing sample points (referred to as branch off points). The perturbations are drawn from a uniform distribution on $[-1, 1]^n$ and scaled to produce small perturbations most of the time.

The branch-off points are chosen using a statistical model. The existing sample points are first grouped into $m$ layers $L_1, \ldots, L_m$, where $x \in L_i$ iff $\ell_{i-1} < f(x) \leq \ell_i$. Here $\ell_0 < \ell_1 \leq \cdots \leq \ell_{m-1} < \ell_m$, $\ell_0 = -\infty$, and $\ell_m = \infty$. A transition matrix $T$ is then formed, where $T_{ij}$ is the perceived probability that perturbing a randomly chosen sample point in layer $j$ will produce a point in layer $i$. Each branch-off point is chosen randomly from within each layer, and the layer is chosen according to the probabilities given in the vector $q$, where the $i^{\text{th}}$ element of $q$ is the probability of choosing layer $i$, and where $q$ is parallel to the solution of

$$\max_{\eta} e_m^T T \eta \quad \text{such that} \quad \eta \geq 0 \quad \text{and} \quad \eta^T W^{-1} \eta = 1.$$

Here $W = \text{diag}(w_1, \ldots, w_m)$ and $w_i$ is the number of sample points in $L_i$. Thus $q$ is a vector which increases the probability that the next sample point will be in $L_m$.

The advantage of a scheme of this form for generating sample points is that it allows the algorithm to automatically vary between randomly chosen sample points, and sample points generated by perturbing existing sample points which are regarded as better than average. The former would be more effective on a function with many widely spaced maxima which are nearly global maxima, whereas the latter would be preferable on, say, a hump shaped function with small ripples which create many local maxima.

It can be shown that the sequence of points is dense in $\mathcal{F}$ with probability 1, and that provided the maximum permitted link length $\ell_{\max}$ goes to zero as the number of sample points increases indefinitely, the algorithm will find all global maxima with probability 1.

It can be shown that if the layer structure and $T$ eventually remain fixed, but $W$ continues to be updated, then at any iteration an *a priori* distribution for the sample points exists in the limit $N \to \infty$, and is independent of the sample points already generated. The limiting

distribution's mean is strictly positive, and if the branch off points are always chosen randomly from the existing sample points, then this mean is constant over $\mathcal{F}$.

**The stochastic process based single linkage**

Let $x_0$ and $x_1$ be sample points. A link from $x_0$ to $x_1$ exists iff the objective function is monotonically increasing along the line segment from $x_0$ to $x_1$. Let $F(t) = f(x_0 + t(x_1 - x_0))$. The second derivative of the objective function on the line segment is modelled by a stochastic process $B(t)$ $t \in [0,1]$. This model is first constructed without reference to the known function and gradient values at the end points; these are included afterwards. By assumption, $F''$ is continuous, but no other information is known about it. Hence, any model for $F''$ should be invariant under translations, and under an interchange of the endpoints. It is reasonable to make the following assumption:

**Assumption 1** *$\forall t_1, t_2, t_3, t_4 \in [0,1]$ satisfying $t_1 < t_2 \leq t_3 < t_4$, the random variables $B(t_2) - B(t_1)$ and $B(t_4) - B(t_3)$ are independent.*

Under these assumptions it can be shown that $B = B_0 + B_1 + D$ where D is a random variable, and where $B_0$ and $B_1$ are independent Brownian motion processes rooted at 0 and 1 respectively. $B_0$ and $B_1$ have zero mean and a variance constant $\sigma^2$. If $D$ is independent of $B_0$ and $B_1$ then the covariance function for $B_0 + B_1 + D$ is translation invariant inside the unit interval. From now on this independence will be assumed.

Let $F_0$, $F_1$, $F_0'$, and $F_1'$ be the known function values and first derivatives of $F$ at $t = 0$ and $t = 1$ respectively. $F(t)$ is modelled by the stochastic process

$$S(t) = F_0 + F_0' t + \int_{s=0}^{t} \int_{u=0}^{s} B(u)\, du\, ds$$

Information about $\sigma^2$ is obtainable from $F_1$ and $F_1'$ by considering

$$F_1' = F_0' + D + \alpha_1 \quad \text{and} \quad F_1 = F_0 + F_0' + \tfrac{1}{2}D + \alpha_2$$

$$\text{where } \alpha_1 = \int_{u=0}^{1} B_0(u) + B_1(u)\, du \quad \text{and} \quad \alpha_2 = \int_{s=0}^{1} \int_{u=0}^{s} B_0(u) + B_1(u)\, du\, ds$$

Defining $Q_{\text{err}} = 2F_1 - 2F_0 - F_0' - F_1' = 2\alpha_2 - \alpha_1$, this Gaussian random variable is independent of $D$, has zero mean, and a variance of $\sigma^2/15$. It can be used to yield a crude estimate of $\sigma^2$ on its own, or combined with estimates from other links under the assumption that there is some sort of uniformity between the covariance constants for different links.

The link's reliability is calculated by estimating the probability that a line local minimum exists along the line segment between the two sample points. This estimate is formed by first estimating the probability distributions for $S'(1/3)$ and $S'(2/3)$ using the estimate of $\sigma^2$. The reliability of each link can then be computed as follows:

1. If $F_1 < F_0$, if $F_0' < 0$, or if $\|x_1 - x_0\| > \ell_{\max}$ then stop as there is no link. Here $\ell_{\max}$ is the maximum permitted link length given by equation (35) in [2].

2. Calculate the probabilities $p_1$ and $p_2$ that $S'(1/3) > 0$ and $S'(2/3) > 0$ respectively.

3. Estimate the reliability of the link as follows: If $F_1' \leq 0$ then the reliability is $1 - (1 - p_1)p_2$, otherwise the reliability is $p_1 p_2$. The link exists only if the reliability exceeds a minimum value.

The global optimisation algorithm can now be stated:

1. Generate initial batch of sample points.

2. Set the number of layers as $\min(N/20, 5)$ and choose the layer boundaries so that there are approximately an equal number of points in each layer. Calculate $q$.

3. Find all points in the top layer which are not linked to any higher point.

4. If the stopping conditions are satisfied, do a local search from each unlinked sample point in the top layer, otherwise generate the next batch of points, adjust $\ell_{\max}$, adjust the perturbation scale factor, and go to step 2.

**Preliminary numerical results and discussion**

The algorithm was tested on a number of standard test functions (see eg [4]), where all sample points were generated randomly in each run. All global maxima were located for each problem and results for these are listed in table 1.

Two additional test functions were also used. For these two problems the number of sample points was kept small in order to simulate what happens in a small part of a large feasible region. The first has a unique global maximum surrounded by a ring of connected local maxima:

$$\max_x f(x) \text{ subject to } x \in [0,1]^n$$

$$\text{where } f(x) = \begin{cases} \cos(4\pi\|x - \frac{1}{2}\|) - \|x - \frac{1}{2}\|^2 & \text{if } \|x - \frac{1}{2}\| \le \frac{1}{2} \\ -\|x - \frac{1}{2}\|^2 & \text{otherwise} \end{cases}$$

The number of successes out of 10 trials were: for $n = 2$ and 50 sample points, MLSL (9), SPSL (10); for $n = 2$ and 13 sample points, MLSL (0), SPSL (10); and for $n = 3$ and 50 sample points, MLSL (1), SPSL (10). In all of these runs all sample points were linked.

The second test problem has two global maxima close together, and is designed to test an algorithm's ability to resolve two close maxima.

$$\textbf{Problem T:} \quad \max_x -\frac{\|x - \frac{1}{2} - 0.15e_1\|^2 \|x - \frac{1}{2} + 0.15e_1\|^2}{(1 + \|x\|^2)^2} \text{ subject to } x \in [0,1]^2$$

The results for this problem are listed in table 2. In each run a fixed number of sample points were generated randomly. The results show that SPSL achieves the same resolution as MLSL with half as many sample points. Results for the Rastrigan function also support this: using 250 random sample points MLSL found the global maximum on 4 runs out of 10, whereas SPSL succeeded 8 times out of 10 using 125 random sample points.

The results show that SPSL is a viable method. If the gradients are only calculated for points in the top layer, then the results for problem T and the Rastrigan function show that SPSL can be more efficient than MLSL, especially if the gradient is significantly cheaper than the worst case for reverse automatic differentiation. The results for the acoustic parameter estimation, and other problems will be included in the final paper.

**References**


[1] Brekhovskikh, L. M. and Yu. P. Lysanov, *Fundamentals of ocean acoustics* 2nd *edition*, Springer-Verlag, ©1991.

[2] Rinnooy Kan, A. H. G. and G. T. Timmer, *Stochastic global optimization methods part I: clustering methods*, Math. Prog. 39, pp 27–56, 1987.

[3] Rinnooy Kan, A. H. G. and G. T. Timmer, *Stochastic global optimization methods part II: multi level methods*, Math. Prog. 39, pp 57–78, 1987.

| problem | local searches | local maxima | sample points |
|---|---|---|---|
| Branin | 3 | 3 | 250 |
| Goldstein-Price | 2 | 2 | 150 |
| Camel | 5 | 3 | 350 |
| Hartman 3 | 2 | 2 | 50 |
| Hartman 6 | 8 | 2 | 500 |
| Shekel 5 | 3 | 3 | 150 |
| Shekel 7 | 5 | 4 | 450 |
| Shekel 10 | 11 | 8 | 500 |
| Rastrigan | 27 | 27 | 500 |

Table 1: Numerical results for the standard test problems.

| sample points | 50 | 75 | 100 | 125 | 150 | 175 | 200 |
|---|---|---|---|---|---|---|---|
| MLSL | 0 | 2 | 8 | 6 | 7 | 9 | 10 |
| SPSL | 6 | 8 | 10 | 10 | 10 | 10 | 10 |

Table 2: Results for problem T.

[4] Törn A. and A. Žilinskas, *Global Optimization*, Lecture notes in Computer Science no. 350, Springer Verlag, 1989.

# Tracking Elementary Particles near their Primary Vertex: A Combinatorial Approach

Jean-François Pusztaszeri, Paul E. Rensing and Thomas M. Liebling

We report on the successful implementation of a discrete optimization algorithm to reconstruct tracks generated by charged elementary particles produced by the LEP accelerator at CERN. This method solves a five-dimensional assignment problem, and is now used to reconstruct data produced by the ALEPH collaboration. It is shown to perform substantially better than the local search methods it is replacing. We believe this represents one of the first successful attempt at applying a combinatorial optimization method to tracking in High Energy Physics.

## Problem definition

Most of High Energy Physics is devoted to the study of fundamental interactions produced by colliding beams of elementary particles, as provided by the LEP storage ring located at CERN, near Geneva. These interactions are observed by means of large detectors which are made of concentric cylindrical shells of electronic arrays, such as the ALEPH compound detector. The inner shells are made of ionization chambers and solid-state devices whose purpose is to record space-points along the flight path of every charged particle produced by the interaction (generally a $Z^0$ decay). Only when such a track is fully reconstructed from its distribution of space-points can the properties of the particle which generated the track be known.

To determine this distribution, one traditionally uses discrete linear filtering methods [5]. These methods return insufficiently precise results if the hit density is large enough and the separation between tracks decreases. We This situation is more commonly encountered near the point of origin (or *vertices*) of these curves, where the precision in track parameters is needed most. This paper presents an algorithm which improves the precision of the track fit in ALEPH, by solving a global assignment of tracks to hits produced by the detector nearest to the point of collision, the Vertex Detector (VDET).

The latter is made of two concentric layers of silicon wafers inlayed with aluminium trigger strips. Wafers overlaps ensure that the entire vertex region (i.e., the collision region) is surrounded. The wafer ideally generates two orthogonal hits for each charge particle which crosses it. From these two hits, a three-dimensional point can be reconstructed.

The detector is immersed in a magnetic field of constant magnitude along the beam (cylinder) axis, which deflect charged particles along a helical path towards the end plates of the cylindrical assembly. The track fitting part of the reconstruction is done by means of a discrete linear filter method (a Kalman filter with a spatial dependence of the track parameters). Dense material present in the detector (walls or wires) often induce perturbations on the tracks however, which requires an increase in extrapolation error in the vertex region.

Throughout this paper, we rely on the availability of a set of outer partial tracks reconstructed from outer tracking devices, where tracks are better separated, and where the pattern recognition problem is solved without ambiguities. We perform a global assignment of outer partial tracks to an equal number of VDET signals in two layers (inner and outer) and in two views (the $z$ direction, along the beam axis, and the $r - \phi$ direction, perpendicular to it).

Track extrapolation errors are represented as cones which cover more than one cross-hit association, and which may overlap each other, hence the intransic ambiguity of this problem. The elliptic intersection of the cones with the wafer planes, together with the hits they cover, are the inputs to a global assignment. In this formulation, a given track may be assigned no hit (when it misses the detector altogether) and up to eight hits, when its extrapolation footprint covers overlap regions in the two layers of the detector, defining a set partioning problem.

**Combinatorial Formulation**

Input

Each element of a set $I$ of outer partial tracks is associated to elements of four disjoint sets of hits in either view and either direction of the VDET, denoted by $J$, $K$, $L$ and $M$. The *pulse-height* of a hit (equivalent to the amplitude of its signal) is used to determine whether that hit may be used more than once in a track assignment. Those which cannot are called single hits.

To reduce the dimension of this problem, we fuse pairs of real hits which lie in the same view and in the same layer, but lie on *different* wafers, and consider them as a single candidate for assignment to a track. Doing so allows us to bring the formulation down from a generalized set partioning problem, with each set containing exactly one track and from one to eight hits, to a five-dimensional assignment problem. This requires a formalism to account for null assignments (i.e., noise and inefficiencies). We introduce therefore a "null" hit (denoted by zero) for each view and each layer. Likewise, we define a noise track (track "0") to collect hits which may have been left unused by the association. The input to our problem is therefore a set of $(n_i+1)$ outer tracks, together with $(n_j+1)$ $\rho-\phi$ hits and $(n_k+1)$ z hits on the outer layer, and $(n_l+1)$ $\rho-\phi$ hits and $(n_m+1)$ z hits on the inner layer.

If $UV_j$, $UW_k$, $LV_l$ and $LW_m$ represent two orthogonal pairs of hits on the outer and lower layers of the vertex detector respectively, and $T_i$ is the track which is to be matched to these hits, the following decision variable

$$x_{ijklm} = \begin{cases} 1 & \text{if } T_i \rightarrow \{\ UV_j, UW_k, LV_l, LW_m\} \\ 0 & \text{otherwise} \end{cases}$$

will determine the association of every outer track $T_i$.

Further reduction in problem size has proved to be experimentally unwieldy: in the absence of correlation between the sets, we may be tempted to formulate our problem as four instances of a *two-dimensional* assignment problem (i.e., a track and a hit in either layer and view), and use polynomial algorithms to solve each one. As experience showed us, we lost our most useful criterion for assignment in doing so, namely the angular comparison between a segment made of two VDET cross-hits on different layers and the angle of incidence of the tracks. For the same reason, a *three-dimensional* assignment for each layer is uninteresting. A plausible alternative would be a three-dimensional assignment formulation *across the views*. Angular information is preserved.

The cost of assignment of each combination is calculated by finding the optimal estimators of the track parameters found by adding the VDET hit candidates to the hit distribution of the outer track. These optimal estimators are found by what amounts to a least square minimization and a goodness-of-fit test for which a $\chi^2$ value may be calculated, but due to the presence of dense material (silicon wafers and their support frames) in the region of interest, it is likely that the particle trajectories undergo substantial local deviations from their ideal helix model. The dynamics of these interactions is modeled by a discrete linear dynamic system, and a Kalman filter algorithm is used to calculate recursively the individual costs of assignment of each pattern. This is expressed in terms of the filtered residuals $\vec{r}_k$ and their covariance matrix $R_k$, and the random variable,

$$\chi^2_k = \chi^2_{k-1} + \vec{r}^{\,T}_k R^{-1}_k \vec{r}_k \qquad (1)$$

will follow a $\chi^2_k$ distribution ($\chi^2_0 = 0$). Given n data points in our original distribution, equation 1 yields the $\chi^2_n$ value for the fit. The $\chi^2$ value of the fit corresponding to some $x_{ijklm} = 1$ assignment is denoted by $C_{\chi^2,ijklm}$

Objective function

We consider here a linear analog to the nonlinear objective function obtained from a normalized $\chi^2$ test. Given the simplified case where the probabilities of the $x_{ijklm} = x^*_{ijklm}$ assignments are assumed to be independent, the hypothesis $x_{ijklm} = x^*_{ijklm}$, for a real track $i$, will have a probability

$$p_{\mathrm{cond},ijklm} = \int_{C_{\chi^2,ijklm}}^{\infty} p_{\chi^2,\mathrm{NDOF}_{\mathrm{vdet}^{(i)}}}(t)dt \tag{2}$$

where $\mathrm{NDOF}_{\mathrm{vdet}}(i)$ is the number of hits in the Vertex Detector which have been assigned to $i$. This probability is conditional to having observed $N_{null,ijklm}$ missing hits in this pattern (one or more indices different from i is equal to zero), so if we assume the local inefficiency of the detector to follow a binomial distribution with probability $\epsilon$, we obtain the individual costs of assignment $c_{ijklm} = -\log p_{\mathrm{cond},ijklm} - \log(\gamma_{null,\epsilon})(N_{null,ijklm})$ . Likewise, considering detector noise as a Poisson process, we may write $c_{0jklm} = -\log p_{\mathrm{poisson}}(\mathrm{INT}(\lambda))$ where $\lambda$ is the expected number of noise hits in the event. These cost coefficients define the following objective function

$$Z(\vec{X}) = \sum_{i=0}^{n_i}\sum_{j=0}^{n_j}\sum_{k=0}^{n_k}\sum_{l=0}^{n_l}\sum_{m=0}^{n_m} c_{ijklm}x_{ijklm} \tag{3}$$

which needs to be minimized.

Constraints

Tracks and hits may not be used freely in the AP5 assignment: having introduced null hits in our formulation, the tracks must always be assigned to *some* hit pattern which can be made of real hits and null hits alike. This gives

$$\sum_{j=0}^{n_j}\sum_{k=0}^{n_k}\sum_{l=0}^{n_l}\sum_{m=0}^{n_m} x_{ijklm} = 1,\ \forall i \in \{1,\ldots,n_i\} \tag{4}$$

The noise track is subject to an inequality constraint. The constraints which apply to real hits in the vertex detector are symmetric in each layer and each view. We take, as an example, all real hits in the $\rho-\phi$ view of the outer layer, which are indexed by $j$. Hit constraints in other views and other layers are identical to those found below, save for the order of the indices. For every real hit $UV_j$, we know an integer $M_j \geq 0$ which indicate how many times that hit is part of a fused "logical" hit located on the overlap region of the wafers. If that number is non-zero, the vector $G_j(M_j)$ contains the indices of the logical overlap hits (in general, one real hit may belong to more than one overlap pair).

We want single hits to be used exactly once, whether assigned to a real or a noise track. If a hit is part of an overlap combination, we require a mutual exclusion between the hit and its overlap parent in the assignment. We have therefore,

$$\sum_{i=0}^{n_i}\sum_{k=0}^{n_k}\sum_{l=0}^{n_l}\sum_{m=0}^{n_m}\left(x_{ijklm} + \sum_{m=1}^{M_j} x_{iG_j^{(m)}klm}\right) = 1 \quad \forall j \in \{1,\ldots,n_{jreal,single}\}$$

Inequality constraints which apply to undecided real hits are expressed in a similar fashion: keeping in mind that it is extremely rare for a hit to be used more than twice, given the fine

resolution of the Vertex Detector, we set the right-hand side of the inequality to two. Hence,

$$1 \leq \sum_{i=0}^{n_i} \sum_{k=0}^{n_k} \sum_{l=0}^{n_l} \sum_{m=0}^{n_m} \left( x_{ijklm} + \sum_{m=1}^{M_j} x_{iG_j^{(m)}klm} \right) \leq 2$$
$$\forall j \in \{n_{jreal,single} + 1, \ldots, n_{jreal}\}$$

The overlap hits are subject to the constraints which apply to their components. No constraints have been placed on the null hits, which are used freely.

**Algorithms**

Preprocessing

Tracks tend to bunch together in subsets which are themselves well separated, so one may successfully apply a partitioning to the main assignment in order to produce several logically independent subproblems of smaller sizes. This is tantamount to finding all connected components in a graph constructed with the tracks as nodes, connected by edges of non-zero weights if and only if any two track are in competition for at least one hit. Together with this component-generation step, we remove entirely diagonal subproblems, i.e., those for which the set of local optimal patterns of each track is in fact the global optimal solution.

This is followed by a heuristic to remove from the component tracks with an extrapolation error which is much larger with respect to other elements of the set. Such tracks, which have been either poorly fitted in the outer tracking region, or have undergone a large smearing due to multiple scattering, may produce patterns with an excellent $\chi^2$, and are therefore more likely to contribute to an optimal solution with little physical significance. By simply removing such tracks from the component, we render this algorithm more robust while reducing component size. Edge weights are calculated using

$$w_{ij} = \frac{\Delta(|A_i - A_j|)}{\max(A_i, A_j)} \tag{5}$$

where $A_i$ and $A_j$ are the average area of intersection between the detector wafer and the extrapolation cones of tracks $T_i$ and $T_j$ respectively, and perform an iteration over the edges in order of non-decreasing length. If the successor of the edge being considered is less than half of its own, we remove this edge from the list and check whether the graph has been disconnected. This latter step is in $O(n)$, where $n$ is the number of nodes in the graph [1]. Once the graph has been disconnected, we reapply the algorithm to the independent components until we are left with a set of dense subproblems. If the end problems are all singletons, then we have a purely sequential assignment algorithm.

Main algorithm

Once preprocessing has completed, the irreducible AP5 subproblems are solved by means of a branch & bound algorithm using linear programming relaxation. This scheme follows the conventional structure of commercial mixed integer programming solver, with some problem-specific steps implemented into the generic structure. Among these is the use of track momentum to generate partial lists of variables while selecting branching variables. If, for a given active node, any two variables with different track indices are in competition, the variable corresponding to the track with the higher momentum, and therefore the smaller extrapolation error, will be branched upon first. Because the footprint of this track will be relatively small with respect to others, so will in general be the number of patterns available for it. For variables all involving the same track index, the ordering will put priority on patterns which involve two pairs of real cross-hits (all four hit indices are greater than zero), and if this is not available, only then one

pair (corresponding to the cases where either hits j,l *or* hits k,m are non-zero). Ordering within these categories is arbitrary. If no cross-hit pairs can be found, then the next candidate in the list is the fully null variable, $X_{i0000}$. The goal of this ordering is to set as early as possible in the algorithm the assignment of tracks for which very little ambiguity exist.

Node ordering among the list of active nodes follows a depth-first search plus backtracking scheme, with an arbitrary left son selected first, as described in Nemhauser & Wolsey [4]. If the node is pruned, the next active node is determined by finding the node for which the lower bound is minimal.

## Results

### Computation

The application of the logical partitioning and clustering steps accounted for a decrease in the average number of tracks per event from 32 to 6, for a sample of a thousand events. This came with no major increase in the *number* of problems to solve, mainly because disconnecting the component graph generated in turn diagonal problems which were solved by a linear time algorithm.

Remaining subproblems of non-trivial size were directed to the branch & bound algorithm. The average number of variables and constraints in the sample were 160 and 31 respectively, for an average component density (node to edge ratio) of 0.5. While these problems are small in average, it is important that the algorithm solves *all* problems encountered, as the larger problems contain generally different physics information than the small ones. Most noticeable are the multiple-prongs tau decay interactions, which consistently reach a thousand variables.

To date, all problems could be solved to optimality by the branch & bound algorithm described earlier, interfaced with a public domain, dual simplex-based linear programming solver called LP_SOLVE, written by Michel Berkelaar from the Eindhoven University of Technology [2]. Compute time for these problems never exceeded a second of CPU time on a DEC ALPHA 3000/300 RISC workstation. The compute time of the entire procedure was in fact dominated by its initialization phase, and most noticeably by the several smoothing iterations required by the Kalman filter while generating the cost matrix. Time spent in preprocessing and post-optimal hit interchange was negligible. No event observed so far has taken more than ten seconds to solve on our computer platform, all phases included.

### Physics results

The method described above has been implemented into the main reconstruction algorithm used by ALEPH. Since it started running at the end of May 1995, this program has reconstructed 10'000 hadronic $Z^o$ events from the 1994 year of production. This section provides a comparison of results obtained by this algorithm and the method it has replaced, called JULIA, which is using a Kalman filter and a greedy search.

We considered instead the physics problem of identifying events which contain b-quarks. This so-called "hadron tagging" is performed by identifying the signature of the event, the long lifetime and large mass of the hadron containing b-quarks, and is fully described in [3]. This method uses what is known as the *impact parameter* of the tracks in the event, defined as the distance of closest approach between a track and the primary vertex. The probability that a measured impact parameter is consistent with zero is computed using a resolution function which can be measured directly from the data. Then, probabilities from the tracks are combined in a two step process to generate a probability that the event contains a long-lived hadron.

Fig. 1 shows the resolution histogram based on the events reprocessed thus far by ALEPH with the help of our algorithm. This plot shows the impact parameter divided by its error for both the old sequential (JULIA) and the new branch & bound based method. The peak of

this distribution represents an error-free pattern recognition, while the tail area reveal a poor resolution of the pattern recognition algorithm. The log scale used for this histogram tends to amplify the measurement errors, but also reveals that the method presented in this paper is doing substantially better when compared to a greedy sequential assignment.

### Conclusion

The method presented in this paper is currently being used to reprocess the 4.5 million events which have been collected by ALEPH since 1990. Improvement observed following this implementation, first on simulated events and later on real data, provided the incentive for starting this effort. An online implementation of this method should follow in early 1996, requiring substantial improvements in the combinatorial algorithms, to accommodate both the near-real-time requirements on compute-time bounds, and an expected tenfold increase in problem size. This work may serve as a base to broaden the use of combinatorial method to other experiments in High Energy physics.

### References


[1] Ahuja R.K., Magnanti T.L., Orlin J.B., *Network Flows*, Prentice Hall, 1993.

[2] Berkelaar M.R.C.M., **LP_SOLVE 2.0**, Eindhoven University of Technology, Eindhoven, The Netherlands. Package available via anonymous ftp at ftp.es.ele.tue.nl/pub/lp_solve/

[3] Brown D., *Tagging b Hadrons using Track Impact Parameters*, ALEPH Note 92-135, CERN, 1992.

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

[5] Particle Data Group, *Review of Particle Properties*, Phys. Lett. B, **239**, 1990.

# An LP-Based Branch and Bound Algorithm for the Quadratic Assignment Problem

K.G. Ramakrishnan, M.G.C. Resende and P.M. Pardalos

The quadratic assignment problem (QAP), first proposed by Koopmans and Beckmann [5], can be stated as

$$\min_{p\in\Pi} \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}b_{p(i)p(j)},$$

where $\Pi$ is the set of all permutations of $\{1,2,\ldots,n\}$, $A=(a_{ij})\in \mathbb{R}^{n\times n}$, $B=(b_{ij})\in \mathbb{R}^{n\times n}$.

Though a wide range of heuristics has been applied to find approximate solutions of large quadratic assignment problems, exact solution approaches have been limited to instances of dimension $n \leq 20$.

Most exact methods for the QAP are based on branch and bound. Lower bounds are key to the computational efficiency of branch and bound algorithms. Recently, Resende, Ramakrishnan, and Drezner [11] used an efficient implementation of an interior point algorithm to compute lower bounds for the QAP by solving the linear programming (LP) relaxation of a classical integer programming formulation of the QAP. The linear program, that has $n^2(n-1)^2/2+n^2$ variables and $2n^2(n-1)+2n$ constraints, is large-scale, by today's standards, even for quadratic assignment problems of small dimension. The linear programs were solved with ADP, an implementation of a dual interior point algorithm, with centering, that uses a preconditioned conjugate gradient algorithm to compute the directions taken at each iteration by the interior point method [4]. Attempts at using the simplex and (direct factorization based) interior point codes of CPLEX [1] were successful only for the smallest instances. That study also showed that the quality of the lower bounds produced was good, suggesting that they be incorporated in a branch and bound algorithm. Pardalos, Ramakrishnan, Resende, and Li [9] describe a branch and bound algorithm used to study the effectiveness of a variance reduction based lower bound [7].

In this paper, we use an extension of the branch and bound algorithm described in [7] in conjunction with the LP-based lower bound described in [11]. We report on preliminary results that show how this lower bound performs when implemented in a branch and bound algorithm for the QAP.

## Implementation

We next outline the branch and bound implementation used in this study. The system is composed of four components: a branch and bound control module; a greedy randomized adaptive search procedure (GRASP) module to produce the initial upper bound; an AMPL modeling language [2] module to manage the linear programming models; and the linear programming solver ADP to produce the lower bounds for the branch and bound algorithm.

The branch and bound control module controls the solution process. It inputs the problem data, calls the GRASP to produce an initial upper bound, and manages the search of the branch and bound tree. At each node of the tree, it spawns an AMPL process that generates the linear program to be solved at that node. The linear program, along with the current best upper bound and the fixed cost associated with the partial assignment of the node, are passed to the LP solver ADP. The optimal permutation is output by the branch and bound module.

The GRASP [8, 10] is called before any search of the branch and bound tree begins so that an initial upper bound can be produced. The branch and bound module passes to the GRASP the problem data and the number of GRASP iterations and gets back the best permutation found over the GRASP iterations and its corresponding cost (the upper bound).

[1] CPLEX is a Registered Trademark of CPLEX Optimization, Inc.

At each node of the branch and bound tree ADP iterates on the LP, producing a sequence of lower bounds, until the dual objective function is greater than the current upper bound minus the fixed cost of the partial assignment, in which case the branch rooted at the current node of the branch and bound tree can be pruned, or if the relative improvement of the dual objective function falls below a specified tolerance, in which case the branch needs to be further explored. If the data is integer, as is the case for all of the QAPLIB instances considered in this paper, the dual interior solution can be rounded up to check for termination. This is done in our code. A detailed description of the ADP code as used in this application is given in [11].

**Computational Results**

In this section, we present preliminary experimental results with our code. The new branch and bound algorithm is tested on a set of standard QAP instances from the QAPLIB. We compare the new algorithm with an algorithm that is identical, except that it uses the Gilmore-Lawler lower bound in place of the LP-based bounds.

The experiments were conducted on a Silicon Graphics (SGI) Challenge (150 MHz MIPS R4400 processor, 1920 Mbytes of main memory, 16 Kbytes of data cache, and 16 Kbytes of instruction cache). The branch and bound control module and the GRASP are implemented in Fortran and compiled with the `f77` compiler using compiler flags `-O2 -Olimit 800`. The interior point solver ADP is written in C and was compiled with the `cc` compiler using compiler flags `-O2 -Olimit 800`.

We tested the codes on several instances from the QAP library QAPLIB. Table 1 summarizes the runs on both algorithms. For each instance it displays the name and dimension of the problem, as well as the solution times and number of branch and bound search tree nodes examined by each of the algorithms. The ratio of CPU times is also displayed.

The number of GRASP iterations was set to 100,000 for all runs.

We make the following remarks regarding the computational results.

- The code solved all 24 instances of QAPLIB of dimension less than or equal to 15.
- Compared with the branch and bound algorithm using the Gilmore-Lawler lower bound, the number of branch and bound search tree nodes examined by the algorithm is small and grows slowly as a function of the dimension $n$ of the QAP.
- Because no level 0 nodes (the QAP linear programming relaxation for the original problem) were done, the minimum number of nodes examined by the algorithm is $n$, the number of level 1 nodes in the tree. Level 0 nodes were solved for all the instances solved in this paper in [11]. In 10 of the 24 problems solved in this paper (`nug05`, `nug06`, `nug07`, `esc08c`, `esc08f`, `chr12a`, `chr12b`, `chr12c`, `chr15b`, and `chr15c`) the level 0 lower bound produced was tight and thus all but one of the problems could be solved at level 0, since the initial upper bounds produced were optimal. The only exception is `nug05` for which the GRASP upper bound initially produced is not optimal.

**Concluding Remarks**

In this paper, we presented implementation details and computational results of a new branch and bound algorithm for solving the quadratic assignment problem.

Our implementation successfully solved to optimality all instances of QAPLIB [1] with dimension $n \leq 15$. The main observation is that the lower bounds are good, resulting in the search of very few branch and bound search tree nodes compared to the same branching scheme using the classical Gilmore-Lawler lower bound [3, 6, 9]. The number of nodes scanned grew less than a cubic function of $n$, the dimension of the QAP. The instance with the largest number of nodes

Table 1: QAP test instances: LP-based vs. GLB-based B&B algorithms

| | | LP-based B&B | | GLB-based B&B | | time | nodes |
|---|---|---|---|---|---|---|---|
| problem | dim | nodes | time | nodes | time | ratio | ratio |
| nug05 | 5 | 12 | 11.7 | 44 | 0.1 | 117.0 | 3.7 |
| nug06 | 6 | 6 | 9.5 | 82 | 0.1 | 95.0 | 13.7 |
| nug07 | 7 | 7 | 16.6 | 115 | 0.1 | 166.0 | 16.4 |
| nug08 | 8 | 8 | 35.1 | 895 | 0.2 | 175.5 | 111.9 |
| nug12 | 12 | 220 | 5238.2 | 49063 | 14.6 | 358.8 | 223.0 |
| nug15 | 15 | 1195 | 87085.7 | 1794507 | 912.4 | 95.4 | 1501.7 |
| scr10 | 10 | 19 | 202.1 | 1494 | 0.6 | 336.8 | 78.6 |
| scr12 | 12 | 252 | 5118.7 | 12918 | 4.8 | 1066.4 | 51.3 |
| scr15 | 15 | 228 | 3043.3 | 506360 | 274.7 | 11.1 | 2220.9 |
| rou10 | 10 | 52 | 275.7 | 2683 | 0.8 | 344.6 | 51.6 |
| rou12 | 12 | 152 | 2715.9 | 37982 | 12.3 | 220.8 | 249.9 |
| rou15 | 15 | 991 | 30811.7 | 4846805 | 2240.3 | 13.8 | 4890.8 |
| esc08a | 8 | 8 | 37.4 | 57464 | 7.0 | 5.3 | 7183.0 |
| esc08b | 8 | 208 | 491.1 | 7352 | 0.7 | 701.6 | 35.3 |
| esc08c | 8 | 8 | 42.7 | 2552 | 0.3 | 142.3 | 319.0 |
| esc08d | 8 | 8 | 38.1 | 2216 | 0.3 | 127.0 | 277.0 |
| esc08e | 8 | 64 | 251.0 | 10376 | 1.0 | 251.0 | 162.1 |
| esc08f | 8 | 8 | 37.6 | 1520 | 0.3 | 125.3 | 190.0 |
| chr12a | 12 | 12 | 312.0 | 672 | 0.7 | 445.7 | 56.0 |
| chr12b | 12 | 12 | 289.4 | 318 | 0.6 | 482.3 | 26.5 |
| chr12c | 12 | 12 | 386.1 | 3214 | 1.5 | 257.4 | 267.8 |
| chr15a | 15 | 15 | 1495.9 | 413825 | 235.5 | 6.4 | 27588.3 |
| chr15b | 15 | 15 | 1831.9 | 396255 | 217.8 | 8.4 | 26417.0 |
| chr15c | 15 | 15 | 1908.5 | 428722 | 240.0 | 8.0 | 28581.5 |

(`nug15` of dimension $n = 15$) required the solution of 1195 linear programs. Other instances of the same size required much fewer (as few as 15) nodes. As the size of the QAP grew the CPU time ratio of the time taken by the new code to the time taken by the GLB based code decreased dramatically. However, for problems of the dimensions considered, it is still faster to use the GLB-based branch and bound algorithm.

The aim of our ongoing research on branch and bound algorithms for the quadratic assignment problem is to make extensions to the algorithm to produce examples for which the new approach, besides scanning much fewer nodes than Gilmore-Lawler lower bound based methods, is also faster than those methods. Since linear programs on different branches of the search tree are essentially independent of each other, it is possible to solve them in parallel. We are implementing a distributed algorithm that solves different linear programs on different processors, broadcasting the value of a new upper bound whenever one is found. In the version of the ADP code used in this paper, the algorithm does not have the capability to do warm starts, i.e. start from an advanced solution. The linear programs are always solved from the start, even when one LP varies from the other by a single column in the dual program. We are implementing a version of ADP that can start from a warm solution. With this, we expect to speed up the solution process significantly. During the tree search small instances of QAPs are induced. The code described in this abstract computes the lower bounds associated with these QAPs with linear programming. The new code will enumerate or use dynamic programming to solve small dimension QAPs. Finally, stronger LP formulations can be derived, producing better lower

bounds.

**References**


[1] R. Burkard, S. Karisch, and F. Rendl, *QAPLIB – A quadratic assignment problem library*, European Journal of Operational Research, 55 (1991), pp. 115–119. Updated version – Feb. 1994.

[2] R. Fourer, D. Gay, and B. Kernighan, *AMPL – A modeling language for mathematical programming*, The Scientific Press, South San Francisco, CA, 1993.

[3] P. Gilmore, *Optimal and suboptimal algorithms for the quadratic assignment problem*, J. SIAM, 10 (1962), pp. 305–313.

[4] N. Karmarkar and K. Ramakrishnan, *Computational results of an interior point algorithm for large scale linear programming*, Mathematical Programming, 52 (1991), pp. 555–586.

[5] T. Koopmans and M. Beckmann, *Assignment problems and the location of economic activities*, Econometrica, 25 (1957), pp. 53–76.

[6] E. Lawler, *The quadratic assignment problem*, Management Science, 9 (1963), pp. 586–599.

[7] Y. Li, P. Pardalos, K. Ramakrishnan, and M. Resende, *Lower bounds for the quadratic assignment problem*, Annals of Operations Research, 50 (1994), pp. 387–410.

[8] Y. Li, P. Pardalos, and M. Resende, *A greedy randomized adaptive search procedure for the quadratic assignment problem*, in Quadratic assignment and related problems, P. Pardalos and H. Wolkowicz, eds., vol. 16 of DIMACS Series on Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, 1994, pp. 237–261.

[9] P. Pardalos, K. Ramakrishnan, M. Resende, and Y. Li, *Implementation of a variance reduction based lower bound in a branch and bound algorithm for the quadratic assignment problem*, tech. rep., AT&T Bell Laboratories, Murray Hill, NJ 07974, 1994.

[10] M. Resende, P. Pardalos, and Y. Li, *FORTRAN subroutines for approximate solution of dense quadratic assignment problems using GRASP*, ACM Transactions on Mathematical Software, (To appear).

[11] M. Resende, K. Ramakrishnan, and Z. Drezner, *Computing lower bounds for the quadratic assignment problem with an interior point algorithm for linear programming*, Operations Research, (To appear).

# An unsolved problem of Fenchel[1]

Tamás Rapcsák

Fenchel (1953, p. 115, Roberts and Varberg, 1973, p. 271) drew up the following problem of level sets: "What "nice" conditions on a nested family of convex sets will ensure that it is the family of level sets of a convex function?" Rapcsák (1991) gave an explicit formulation of the gradient of the class of the smooth pseudolinear functions (both pseudoconvex and pseudoconcave). This result means an extension of the Cauchy functional equation and the solution of the Fenchel's problem in the case of a nested family of convex sets whose boundaries are hyperplanes which define an open convex set, and where this family of convex sets corresponds to the equality level sets of pseudolinear functions. We have to point out, however, that the pseudolinear functions are more general than the convex functions. The following theorem was proved for characterizing the gradient of the smooth pseudolinear functions:

**Theorem 1** *[2] Let a three times continuously differentiable function $f$ be defined on an open convex set $A \subseteq R^n$ and assume that $\nabla f(\mathbf{x}) \neq \mathbf{0}$, $\mathbf{x} \in \mathbf{A}$. Then, $f$ is pseudolinear on $A$ iff there exist continuously differentiable functions $l(\mathbf{x})$, $\eta_i(f(\mathbf{x}))$, $i = 1, \ldots, n$, $\mathbf{x} \in \mathbf{A}$, such that the following conditions are satisfied:*

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x_i}} = l(\mathbf{x})\eta_{\mathbf{i}}(\mathbf{f(x)}), \quad \mathbf{i = 1, \ldots, n}, \quad \mathbf{x \in A}.$$

In order to construct pseudolinear functions, the next theorem can be useful.

**Theorem 2** *[2] If the functions $l(\mathbf{x})$, $\eta_i(f(\mathbf{x}))$, $i = 1, \ldots, n$, $\mathbf{x} \in \mathbf{A}$, have continuous derivatives in all arguments on an open set $A \subseteq R^n$ and if they satisfy the compatibility conditions*

$$\frac{\partial l(\mathbf{x})}{\partial \mathbf{x_j}}\eta_i(f(\mathbf{x})) + \mathbf{l^2(x)}\frac{\mathbf{d}\eta_{\mathbf{i}}(\mathbf{f(x)})}{\mathbf{df}}\eta_{\mathbf{j}}(\mathbf{x}) = \frac{\partial \mathbf{l(x)}}{\partial \mathbf{x_i}}\eta_{\mathbf{j}}(\mathbf{f(x)}) + \mathbf{l^2(x)}\frac{\mathbf{d}\eta_{\mathbf{j}}(\mathbf{f(x)})}{\mathbf{df}}\eta_{\mathbf{i}}(\mathbf{x}),$$

*$i, j = 1, \ldots, n$, $\mathbf{x} \in \mathbf{A}$, then a uniquely determined solution of the system*

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x_i}} = l(\mathbf{x})\eta_{\mathbf{i}}(\mathbf{f(x)}), \qquad \mathbf{i = 1, \ldots, n}, \qquad \mathbf{x \in A},$$

*exists in a neighbourhood of every point of $A$ as soon as the value of the function $f$ is prescribed at some point of the neighbourhood.*

The purpose of the paper is to solve Fenchel's problem if the boundaries of the nested family of convex sets are given by $n$-dimensional differentiable manifolds which determine an open convex set with nonempty interior in $R^{n+1}$.

## References

[1] Fenchel, W., Convex cones, sets and functions, (mimeographed lecture notes), Princeton University Press, Princeton, New Jersey, 1953.

[2] Rapcsák, T., On pseudolinear functions, European Journal of Operational Research 50 (1991) 353-360.

[3] Roberts, A. W. and Varberg, D. E., Convex functions, Academic Press, New York, London, 1973.

[1] This research was supported in part by the Hungarian National Research Foundation, Grant No. OTKA-T016413 and OTKA-T017241

# Techniques for Gap-Treating and Box-Splitting in Interval Newton Gauss-Seidel Steps for Global Optimization

D. Ratz

For a twice continuously differentiable function $f : D \to I\!R$ and $D \supseteq [x] \in I\!I\!R^n$, we address the problem of finding all points $x^*$ in the interval vector $[x]$ such that

$$f(x^*) = \min_{x \in [x]} f(x).$$

We are interested in both the global minimizers $x^*$ and the minimum value $f^* = f(x^*)$.

We use the interval branch-and-bound method described in [2] and [3] with several modifications. The method subdivides the initial box $[x]$, stores the subboxes in a list $L$, and discards subintervals which are guaranteed not to contain a global minimizer, until the desired accuracy of the intervals in the list is achieved. The tests we use to discard or to prune pending subboxes are cut-off test, monotonicity test, concavity test, and the extended interval Newton Gauss-Seidel step.

The latter we apply to the nonlinear system $\nabla f(y) = 0$ with $y \in [y]$. The subbox $[y]$ is a candidate box for enclosing a minimizer $x^*$, for which we assume $\nabla f(x^*) = 0$. One step of the extended interval Newton Gauss-Seidel method shall improve the enclosure $[y]$ by formally solving the system

$$b = [A] \cdot (c - y),$$

where $b = R \cdot \nabla f(c)$, $[A] = R \cdot \nabla^2 f([y])$, $c = m([y])$, and $R \in I\!R^{n \times n}$ is some preconditioner matrix. Then, we compute $N'_{\mathrm{GS}}([y])$ according to

$$\left.\begin{aligned} [z] &:= [y] \\ [z]_i &:= \left(c_i - \left(b_i + \sum_{\substack{j=1 \\ j \neq i}}^{n} [A]_{ij} \cdot ([z]_j - c_j)\right) \Big/ [A]_{ii}\right) \cap [z]_i, \qquad i = 1, \ldots, n \\ N'_{\mathrm{GS}}([y]) &:= [z] \end{aligned}\right\}.$$

If $0 \in [A]_{ii}$ for some $i$, extended interval arithmetic is applied, which allows division by an interval containing zero. In this case, a gap can be produced in the corresponding component $[z]_i$ of $[z]$. Therefore, if $0 \in [A]_{ii}$ for several components $i$, the extended interval Newton Gauss-Seidel step possibly produces several gaps in the actual box $[y]$ and it may result in the union of several boxes $N'_{\mathrm{GS}}([y]) = [V]_1 \cup \ldots \cup [V]_p$, where $[V]_i \in I\!I\!R^n$, $i = 1, \ldots, p$, that is $[V] \in I\!I\!R^{p \times n}$.

We investigate the impact of different techniques for gap-treating and box-splitting which can be applied resulting in different values for $[V]$ and $p$.

In the main optimization algorithm, different subdivision direction selection rules can be applied to determine "optimal" components for bisection of the current box $[y]$ (see [1] and [4]). Each of these rules selects a direction $k$ with $D(k) = \max_{i=1}^{n} D(i)$, where $D(i)$ is determined by the given rule.

We investigate some of these rules in connection with the interval Gauss-Seidel step, where we use them to compute a sorting vector $s = (s_1, s_2, \ldots, s_n)$ with $s_i \in \{1, \ldots, n\}$ and $s_i \neq s_j$ for $i \neq j$, which satisfies $D(s_i) \geq D(s_{i+1})$, $i = 1, \ldots, n-1$ for the corresponding direction selection rule $D(\ldots)$. Then, we study the performance of a *sorted* interval Newton Gauss-Seidel step given by

$$\left.\begin{aligned} [z] &:= [y] \\ [z]_{s_i} &:= \left(c_{s_i} - \left(b_{s_i} + \sum_{\substack{j=1 \\ j \neq s_i}}^{n} [A]_{s_i j} \cdot ([z]_j - c_j)\right) \Big/ [A]_{s_i s_i}\right) \cap [z]_{s_i}, \qquad i = 1, \ldots, n \\ N'_{GS}([y]) &:= [z] \end{aligned}\right\}$$

incorporating different splitting techniques.

We propose strategies which improve the overall efficiency of the interval Newton Gauss-Seidel step and therefore of global optimization methods. We present results of computational experiments with standard global optimization problems.

**References**


[1] Csendes, T., Ratz, D.: *Subdivision Direction Selection in Interval Methods for Global Optimization.* SIAM Journal of Numerical Analysis, accepted for publication, 1995.

[2] Hammer, R., Hocks, M., Kulisch, U., Ratz, D.: *Numerical Toolbox for Verified Computing I – Basic Numerical Problems.* Springer-Verlag, Heidelberg, New York, 1993.

[3] Ratz, D.: *Box-Splitting Strategies for the Interval Gauss-Seidel Step in a Global Optimization Method.* Computing **53**, 337–353, Springer-Verlag, Wien, 1994.

[4] Ratz, D., Csendes, T.: *On the Selection of Subdivision Directions in Interval Branch-and-Bound Methods for Global Optimization.* Journal of Global Optimization, accepted for publication, 1995.

## Analysis of threshold accepting global optimization methods

Marco Locatelli and Fabio Schoen

Among the best performing algorithms for global optimization of reasonably smooth multimodal functions over simple feasible sets (e.g. hypercubes) there are quite a few approaches in which a proper mix of random sampling and local search is performed. One quite well known such an approach is Multi-level Single-Linkage ([3], [4]), where local searches are started from selected points in a random uniform sample of the feasible region. Although there exist many other approaches based upon the idea of mixing random sampling and local searches, (one of the most interesting of which being [2]), it is felt by the authors (and confirmed by numerical experiments), that the criterion used in Multi-level Single-linkage (MLSL for short) is a particularly sensible one. The basic idea of MLSL is that of starting a local search from a sampled point whenever the distance between such a point and the nearest higher-valued point (in case of maximization) is greater than a threshold. As the threshold used in MLSL is decreasing, points which are judged "near" in the early stages of the algorithm, might become "far" later on; this fact precludes the possibility of sequential sampling, being the process of revising previous decisions on sampled points too costly from a computational point of view. The authors of MLSL thus propose a batch sampling strategy, by which at each stage of the algorithm a bunch of points, say $N > 0$, is sampled and the decision whether to start or not a local search from points in the sample is taken once every stage.

In this paper we present and analyze a new class of stochastic global optimization algorithms which

- possess all of the theoretical characteristics of MLSL – namely:
  1. almost sure convergence to the global optimum;
  2. observation of the global optimum after a finite number of iterations with probability 1;
  3. probability of starting a local search decreasing to 0;
  4. expected number of local searches performed, even if the algorithm is never stopped, finite with probability 1;
- samples sequentially – thus avoiding the necessity of pre-specifying a batch size $N$;
- enables the start of local searches only from the most recently sampled point – thus avoiding the time-consuming analysis of the whole sample needed in MLSL in order to take account of the decreasing threshold;
- differently from what is required in the theoretical analysis of MLSL, lets local searches start even from points sampled on the boundary of the feasible region – which becomes particularly interesting for problems of concave minimization.

### Simple Linkage

In [1] a family of algorithms all possessing the above properties was introduced, where the decision whether to start or not a local search from the current point was based on a randomized criterion. Here we restrict the attention to threshold based methods in which we decide to start a local search at iteration $k$, based on a sample $X_1, \ldots, X_k$, if

$$\alpha_k \leq \min_j \{\|X_k - X_j\| : f(X_k) \leq f(X_j) + \epsilon\}$$

where $f$ is a real valued function from $[0,1]^d$ in $R$ to be maximized, $\epsilon$ is a small positive constant, $\alpha_k$ is a non-negative sequence.

It was proven in [1] that the probability of starting a local search decreases to 0 if and only if

$$\alpha_k k^{1/d} \to \infty.$$

Let

$$\alpha_k := \frac{1}{\sqrt{\pi}} \left( \frac{\sigma}{\Gamma(1+d/2)} \frac{\log k}{k} \right)^{1/d}$$

where, $\sigma$ is a constant to be chosen by the user. It was proven in [1] that, provided that $\sigma > 1$ and that no local search is started from within a prescribed distance from the boundary of the feasible region, then the algorithm, while being able to find with probability 1 the global optimum, will perform a number of local searches whose expectation is finite even if the algorithm were never stopped.

Actually, threshold accepting algorithms may be much more general than this one, both by using more general thresholds and by letting local searches start from the boundary, but here, in order to be able to make significant comparisons, we will restrict the analysis to this special instance. We adopt the convention of calling "Simple Linkage" this algorithm.

For what concerns MLSL, a positive constant $\sigma_M$ and an integer constant $N > 0$ are given, and sampling proceeds in batches. The decisions about starting or not local searches can be taken only after $N$ points have been drawn — so that the decision epoches are $N$, $2N$, .... At each decision epoch, say $h \geq 1$, a threshold is computed

$$\delta_{hN} = \frac{1}{\sqrt{\pi}} \left( \frac{\sigma_M}{\Gamma(1+d/2)} \frac{\log hN}{hN} \right)^{1/d}. \tag{1}$$

Given this threshold, the whole sample of $hN$ points is reconsidered and a local search is started from $X_i$, $i \in \{1 \ldots, hN\}$ if and only if

$$\min_{j \leq hN, j \neq i} \{ \|X_i - X_j\| : f(X_j) \geq f(X_i) \} \geq \delta_{hN} \tag{2}$$

Let us briefly denote with $Y_k^{(i)} := \min\{\|X_i - X_j\| : j \leq k, j \neq i\}$ the random variable corresponding to the minimum distance between a sample point $X_i$ and the first $k$ points in the sample. Then a local search from $X_i$ is started in SL if and only if $Y_i^{(i)} > \alpha_i$

while it is started in MLSL from the same point if and only if $\exists\, h \geq \left\lceil \frac{i}{N} \right\rceil : Y_{hN}^{(i)} > \delta_{hN}$.

**Theoretical comparison between MLSL and SL**

We plan to compare the behaviour of the proposed algorithm and MLSL when the sample is the same. We assume that there exists a constant $\beta > 1$ such that $\frac{\sigma_M}{\sigma} = \beta$. In the applications this is the most common situation: it has been proven in fact that a finite expected number of local searches in MLSL is obtained for $\sigma_M > 4$, while the same holds for SL when $\sigma > 1$. It is thus sensible, for comparison to choose, for example, $\beta = 4$. In a forthcoming paper results will be given for the general case.

Let us denote with $S_i$ and $M_i$ the events that, respectively, SL and MLSL decide to start a local search from $X_i$. We look for bounds on

$$P(S_i \mid M_i) \qquad P(\neg S_i \mid \neg M_i)$$

i.e. on the probability that a local search is started in SL given that, *sometimes after the i–th iteration* also MLSL has decided to start a local search from the same point and the probability that, if MLSL decides never to start a local search from some point, then also SL decides the same.

The main results can be stated as follows:

$$P(S_i \mid M_i) \gtrsim \frac{\beta^{\beta\sigma} - 1}{\beta^{\beta\sigma-1} + \beta} \qquad P(\neg S_i \mid \neg M_i) \sim 1$$

Here the symbol $\gtrsim$ is used to denote "asymptotic minorization", i.e., if $a_k \gtrsim b_k$ then, for every $\epsilon > 0$ $\exists \bar{k}$ such that $k \geq \bar{k}$ implies

**Conclusions**

Simple Linkage seems to be an extremely simplified version of MLSL which, while retaining all of the good theoretical properties of the latter, is much easier to implement and order of magnitude faster. It is also to be noticed that SL is one out of an infinite family of algorithms all possessing the same strong theoretical properties. It is a subject of current research to try to understand which algorithm in the family can be proposed as the "best" one.

It is to be observed also that all of the properties enjoyed by MLSL (and by SL) are asymptotic ones. The analysis carried out in [1] shows that, asymptotically, there is no advantage in building "chains" of improving points (as MLSL does); moreover, since the publication of MLSL, apparently it was never explicitly noticed that, asymptotically, function values become irrelevant, all of the decisions being taken only the basis of relative distances between points in the sample. This is the key point for being able to derive accurate asymptotic results. But what happens in the first stages? The analysis carried out both for MLSL and for SL tells nothing about the finite time behaviour of those algorithms; it seems plausible to assume that results in this direction, although much welcome, will be very hard to obtain.

In conclusion, we know very little on the actual behaviour of both MLSL and SL (although computational experiments tend to support the evidence that they perform the same in terms of number of function evaluations and local searches performed). Asymptotically, MLSL and SL behave the same: SL, being much easier and more efficient, seems to be a practical alternative to MLSL.

**References**


[1] Locatelli, M. and Schoen, F., "Random Linkage: a family of acceptance/rejection algorithms for global optimisation", submitted, 1995 (available in pre-print form through the `www` page of the second author: `http://www-dsi.ing.unifi.it/~schoen/home.html`).

[2] Lucidi, S. and Piccioni, M., "Random Tunneling by Means of Acceptance-Rejection Sampling for Global Optimization"I, *J.O.T.A.*, **62**, 255–276, 1989.

[3] Rinnooy Kan, A. H. G. and Timmer, G.T., "Stochastic Global Optimization Methods. Part I: Clustering Methods", *Mathematical Programming*, **39**, 27–56, 1987.

[4] Rinnooy Kan, A. H. G. and Timmer, G.T., "Stochastic Global Optimization Methods. Part II: Multi Level Methods", *Mathematical Programming*, **39**, 57–78, 1987.

# An Algorithm for Minimizing Functions with Lipschitz Derivatives

Yaroslav D. Sergeyev

A new algorithm for minimizing one-dimensional functions having Lipschitz first derivatives is presented. The method does not require a priori knowledge of the exact Lipschitz constant but adaptively estimates the local ones in different sectors of the search region in the course of minimization. Convergence conditions of the method are investigated. Some numerical examples are also presented.

Let us consider the following global optimization problem

$$x^* = \text{argmin}\{f(x) : x \in [a, b]\}, \tag{1}$$

Many algorithms have been proposed for solving this problem under different assumptions (see [1]–[11]). In this paper it is supposed that the objective function has a finite number of the local minima and its first derivative $f'(x)$ satisfies the Lipschitz condition, i.e.

$$| f'(x_1) - f'(x_2) | \leq L | x_1 - x_2 |, \forall x_1, x_2 \in [a, b], \tag{2}$$

where the constant $0 < L < \infty$ is called the global Lipschitz constant.

For solving the problem (1) under assumption (2) two methods have been proposed independently. In the first one Breiman & Cutler [2] (see also [1]) consider the case the constant $L$ from (2) is a priori known. Gergel [5] proposes another approach estimating $L$ in the course of optimization. Both the methods use the global Lipschitz constant $L$ (or its upper estimate) to construct support functions for $f(x)$ over the interval $[a, b]$.

The algorithm proposed here constructs an auxiliary function $\varphi(x)$ also. The difference with the previous approaches consists in the fact that $\varphi(x)$ is constructed using estimates $\mu_i$ of the *local* Lipschitz constants $L_i$ of intervals $[x_{i-1}, x_i] \subset [a, b]$ where $x_i, 1 \leq i \leq k$, are trial points previously produced by the algorithm (the term *trial* means evaluating $f(x)$ and its derivative $f'(x)$ at a point $x$). Thus, the method executes a local tuning on the behavior of the objective function over different subintervals in $[a, b]$. It has been demonstrated [10] for some global optimization algorithms (which do not use derivatives) that using local information can accelerate the search significantly.

## The algorithm

Two initial trials are performed at the points $x^1 = a$ and $x^2 = b$. The point $x^{k+1}, k \geq 2$, of the current $(k+1)$th iteration is chosen by the following rules.

**Step 1.** Order the points $x^1, \ldots, x^k$ of previous $k$ trials by increasing their coordinates, i.e.

$$a = x_1 < x_2 < \ldots < x_i < \ldots < x_k = b. \tag{3}$$

Underline, that the record $x^k$ means that this point has been produced in the course of the $k$th iteration of the method. On the other hand the equality $x^k = x_i$ shows the position of $x^k$ in the series (3).

**Step 2.** Calculate the auxiliary values $v_i$ being low bounds for the local Lipschitz constants $L_i$ of the intervals $[x_{i-1}, x_i]$, where $x_i, 1 < i \leq k$, are from (3)

$$v_i = \max\{\tau(x) : x \in [x_{i-1}, x_i]\}, \tag{4}$$

$$\tau(x) = 2\frac{| z_i - z_{i-1} + z'_i(x - x_i) - z'_{i-1}(x - x_{i-1}) |}{(x - x_i)^2 + (x - x_{i-1})^2}$$

and $z_{i-1} = f(x_{i-1}), z_i = f(x_i), z'_{i-1} = f'(x_{i-1}), z'_i = f'(x_i)$.

**Step 3.** Calculate estimates $\mu_i$ of the local Lipschitz constants $L_i$ for the intervals $[x_{i-1}, x_i], 1 < i \le k$:

$$\mu_i = r \max\{\lambda_i, \gamma_i, \xi\}, \tag{5}$$

where $r > 1, \xi > 0$ are parameters of the method and the values $\lambda_i, \gamma_i$ are computed using the values $v_i, 1 < i \le k$, from (4):

$$\lambda_i = \max\{v_j : 1 < j \le k, i-1 \le j \le i+1\}, \tag{6}$$

$$\gamma_i = M(x_i - x_{i-1})/X^{\max}, \tag{7}$$

$$M = \max\{v_i : 2 \le i \le k\}, \quad X^{\max} = \max\{x_i - x_{i-1} : 2 \le i \le k\}.$$

**Step 4.** Compute for each interval $[x_{i-1}, x_i], 1 < i \le k$, the *characteristic*

$$R(i) = \min\{z_{i-1}, z_i, z_{i-1} + z'_{i-1}(\hat{x}_i - x_{i-1}) - 0.5\mu_i(\hat{x}_i - x_{i-1})^2\}, \tag{8}$$

where $\mu_i$ is from (5) and

$$\hat{x}_i = \frac{-z_i + z_{i-1} + z'_i x_i - z'_{i-1} x_{i-1} + 0.5\mu_i(x_i^2 - x_{i-1}^2)}{\mu_i(x_i - x_{i-1}) + z'_i - z'_{i-1}} \tag{9}$$

**Step 5.** Find among the intervals $[x_{i-1}, x_i], 1 < i \le k$, an interval $[x_{t-1}, x_t]$ such that

$$R(t) = \min\{R(i) : 1 < i \le k\}; \tag{10}$$

**Step 6.** Execute the new trial at the point $x^{k+1} = \hat{x}_i$, where $t$ is from ( 10) and $\hat{x}_t$ is calculated according to ( 9).

A complete theoretical basis of the algorithm will be presented in the full paper. Here we only note that the value $\mu_i$ estimates the local Lipschitz constant $L_i$ over the interval $[x_{i-1}, x_i]$. The values $\lambda_i$ and $\gamma_i$ spy respectively on the local and the global information obtained in the course of the previous $k$ iterations. When the interval $[x_{i-1}, x_i]$ is small then due to (7) $\gamma_i$ is small also and we use the local information represented by $\lambda_i$. When the interval $[x_{i-1}, x_i]$ is very wide reliability of the local information is low and the global estimate $\gamma_i$ plays the main role. Taking in consideration the local estimates $\mu_i$ permits to construct an auxiliary function $\varphi(x)$ (the characteristic $R(i)$ is its minimum over $[x_{i-1}, x_i]$) which is closer to $f(x)$ than that ones which use only global Lipschitz constant $L$ or its estimates. Underline, that the local information is used over the *whole*

**Theorem.** *Let $x^*$ be a global minimizer of $f(x)$ and $[x_{i-1}, x_i], i = i(k)$, be an interval containing this point in the course of the $k$-th iteration of the algorithm. If there exists an iteration number $s$ such that for all $k \ge s$ for the value $\mu_i$ from (5) the inequality*

$$\mu_i \ge L_i$$

*is true then, the point $x^*$ is a limit point of the sequence of trial points generated by the algorithm.*

Theorem demonstrates that to have the global convergence it is not necessary to estimate correctly the *global* Lipschitz constant (it may be underestimated) but it is enough to have a correct notion only about the local Lipschitz constant for the subinterval containing the point $x^*$ in the course of the $k$th iteration.

**Numerical examples**

To illustrate performance of the new algorithm (NA) in Tab.1 we compare it with the methods of Breiman & Cutler (BC) and Gergel (GM) on that test functions from [6] where there exist subregions with local Lipschitz constants less than the global one. The parameters of the algorithms have been chosen as follows : $r = 1.1$ for GM and NA. The parameter $\xi$ for NA has been taken equal to $10^{-9}$. We have used the exact values of global Lipschitz constants for $f'(x)$ in BC. We stopped our experiments when the interval to place a new trial was less than $\epsilon = 10^{-4}(b-a)$, where $a, b$ are from (1). Global minima have been found by all the methods for all the functions.

Table 1: Number of trials done by the methods before satisfaction of the stopping rule.

| Problem | BC | GM | NA |
|---|---|---|---|
| 3 | 103 | 98 | 90 |
| 13 | 264 | 89 | 47 |
| 15 | 80 | 47 | 29 |
| 16 | 88 | 75 | 34 |
| 17 | 67 | 65 | 46 |
| Average | 120.4 | 74.8 | 49.2 |

**References**


[1] Baritompa W. (1994), Accelerations for a variety of global optimization methods, *J. of Global Optimization*, **4**(1), 37–45.

[2] Breiman, L. and A. Cutler (1993), A deterministic algorithm for global optimization, *Math. Programming*, **58**, 179–199.

[3] Csendes, T. (1989), An interval method for bounding level sets of parameter estimation problems, *Computing*, **41**, 75–86.

[4] Evtushenko, Yu.G., M.A. Potapov and V.V. Korotkich (1992), Numerical methods for global optimization, *Recent Advances in Global Optimization*, ed. by C.A. Floudas and P.M. Pardalos, Princeton University Press, Princeton.

[5] Gergel, V.P. (1992), A global search algorithm using derivatives, *Systems Dynamics and Optimization*, N.Novgorod University Press, N.Novgorod, 161–178.(In Russian).

[6] Hansen, P., B. Jaumard and S.-H. Lu (1992), Global optimization of univariate Lipschitz functions: 1–2, *Math. Programming*, **55**, 251–293.

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

[8] Kostrowicki J. and H.A. Scheraga (1995), Simple global minimization algorithm for one–variable rational functions, *J. Global Optimization*, **6**, 293–311.

[9] Pintér, J. (1992), Convergence qualification of adaptive partition algorithms in global optimization, *Math. Programming*, **56**, 343–360.

[10] Sergeyev, Ya.D. (1995), An information global optimization algorithm with local tuning, to appear in *SIAM J. Optimization*.

[11] Strongin, R.G. (1989), The information approach to multiextremal optimization problems, *Stochastics & Stochastics Reports*, **27**, 65–82.

# Global Optimization Requires Global Information

Chris Stephens and William Baritompa

There are many global optimization algorithms which use global information. For instance, Lipschitz algorithms, bounded second derivative algorithms, interval methods, pure adaptive search and simulated annealing. All of these algorithms share the properties that they require global information in the form of a parameter (e.g. Lipschitz constant, bound on second derivative, functional form, level sets or a cooling schedule) and they are guaranteed to find the global optimum. One criticism of these algorithms is that this information, being of a global nature, is hard to obtain (or simply may not be available).

Thus there is a desire to design algorithms which avoid the need for global information. A number of algorithms have been proposed with this in mind. For example, the DIRECT algorithm of Jones, Perttunen and Stuckman, Strongin's algorithm, algorithms of Gergel and Sergeyev, adaptive search and simulated annealing as used in practice. These algorithms often perform very well in empirical tests, and can even outperform their global counterparts. However they do have inherent theoretical limitations.

Hansen, Jaumard and Lu [1] found a class of functions for which Strongin's algorithm fails to converge. Törn and Žilinskas [2] showed all deterministic algorithms which use only function values at sample points converge to the global optimum on all continuous functions if and only if they search a dense set.

Introducing a stochastic element into algorithms is often seen as a way to overcome these limitations, (so that no function can be found that will definitely fail). In this paper we extend Törn and Žilinskas's results to include stochastic algorithms, as well as to algorithms which use other local information, such as derivatives, in addition to function values. We describe other classes of functions for which these algorithms will fail.

Our results show that any algorithm, including stochastic algorithms, using only local information will succeed frequently on all functions (in certain classes) if and only if all points in the domain are frequently seen. That is, all algorithms which use local information only, must use brute force if convergence is guaranteed.

Furthermore, we show that attempts to localize the global optima on all functions with such algorithms will always fail. Given any algorithm (including stochastic algorithms) we show the existence of a function for which the probability of the sample points converging to the global optima (or a subset of them) is arbitrarily small.

An important example is simulated annealing. "Standard" simulated annealing localizes if the cooling schedule is slow enough. It has been shown (in the deterministic setting) a necessary and sufficient condition on the cooling schedule depends on the depth of the lowest local minimum. This is clearly a global parameter. In the continuous case where gradients are used, our results show the cooling schedule must also depend on global properties. So, attempts to find a suitable (or optimize an existing) cooling schedule by pre-sampling or adjusting the cooling schedule on the run using the new sample points are doomed to failure. Our results show that there are always functions for which the probability of success of such a scheme is arbitrarily small.

In practice algorithms must stop after a finite time and hence do not look everywhere. So, if no global information about the problem is utilized, the function tried may be one on which the algorithm fails. We cannot have mathematically justified confidence in the results.

However, many of these algorithms do have empirical and heuristic justification. They are often designed for certain real-world problems and perform well when tested on these and similar problems. Indeed, these global optimization heuristics are often far more practical than general algorithms run until the mathematically proven stopping criteria are satisfied. These real-world

and test functions must have nicer characteristics, than "randomly chosen" functions from a formal class.

There are two areas for future work, which may prove fruitful. Firstly, by our results, the "niceness" of real-world and test functions (when it exists) must be a global characteristic. This illustrates the need to quantify this "niceness" into a useful global parameter. Success of such an undertaking would result in algorithms with the practical usefulness of current heuristics with the addition of mathematically justified confidence in the results.

The second area is to design algorithms which use both local and global information without losing guaranteed convergence. Lipschitz and second derivative methods suffer when there are large first or second derivatives remote from the global optima. Algorithms which use local information only can often adapt to local conditions as they progress. Meewella and Mayne were able to obtain local Lipschitz constants adaptively by using interval arithmetic and the global functional form. It is hoped that similar modifications to second derivative methods will yield an efficient guaranteed algorithm.

**References**

[1] P. Hansen, Jaumard B., and Lu S-H. On using estimates of Lipschitz constants in global optimization. *Journal of Optimization Theory and Applications*, 75:195–200, 1992.

[2] Aimo Törn and Antanas Žilinskas. *Global Optimization*, volume 350 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin Heidelberg, 1989.

# On Global Search in Non-Convex Optimal Control Problem

Alexander S. Strekalovsky and Igor L. Vasiliev

We consider non-convex (w.r.t. the state) optimal control problems. A global search algorithm based on global optimality conditions is proposed. Several test problems are solved.

It is known, the sufficient optimality conditions as the dynamic programming and the Krotov's conditions have some disadvantages which are not characteristically of the Pontryagin maximum principle. First of all it refers to the numerical methods constructed on their basis [1], [2],[3].

In this paper we propose an algorithm allowing to transcend stationary, i.e. to escape from the process $(x(\cdot), u(\cdot))$ verifying Pontryagin maximum principle, if $(x(\cdot), u(\cdot))$ is not global solution.

The approach proposed here is based on global optimality conditions presented in [4], [5] and develops the results from [6], [7].

## Problem Formulation

Consider the following control system:

$$\left.\begin{array}{l} \dot{x}(t) = A(t)x(t) + f(u(t), t), \\ x(t_0) = x^0; \;\; t \in ]t_0, t_1[ \stackrel{\triangle}{=} T, \\ -\infty \;<\; t_0 \;<\; t_1 \;<\; +\infty; \;\; u(\cdot) \in \mathcal{U}, \\ \mathcal{U} = \{u \in L^r_\infty(T) : u(t) \in U \;\; \stackrel{0}{\forall}\, t \in T\}; \end{array}\right\} \tag{1}$$

where $x(t) = (x_1(t), ..., x_n(t))^T$, $u(t) = (u_1(t), ..., u_r(t))^T$, $x^0 \in R^n$ — an initial state, $U$ is a compact from $R^r$, and the designation $\stackrel{0}{\forall}$ means "for almost every in the sense of Lebesque measure". Other assumptions are usual for optimal control. Denote by $x(t, u)$ the solution of the differential equations system (1) corresponding to a control $u(\cdot) \in \mathcal{U}$.

Let we have to maximize the functional

$$J(u) \stackrel{\triangle}{=} g(x(t_1)) \to \max, \tag{2}$$

where $g : R^n \to R$ is a convex function.

It can be readily seen that the problem (1)-(2) is non-convex, and this nonconvexity is generated by the objective functional.

Hence, Pontryagin maximum principle is not sufficient condition for a control to be global optimal.

## Global Search Algorithm

First of all let us describe the algorithm step by step. Suppose a control $\bar{u}^1(\cdot)$ belong to $\mathcal{U}$.

Let a control $\bar{u}^k(\cdot) \in U$ and a sequence $\{\epsilon_k\}$, $\epsilon_k > 0$, $k = 1, 2, ...$, $\epsilon_k \downarrow 0$ $(k \to \infty)$, be given.

**Step 1** *Let* $\left(x^k(\cdot), u^k(\cdot)\right)$, $x^k(\cdot) = x\left(\cdot, u^k\right)$ *be* $\epsilon_k$*- stationary process obtained by a local search algorithm beginning at the control* $\bar{u}^k(\cdot)$ *:*

$$J\left(\bar{u}^k\right) \stackrel{\triangle}{=} g\left(x\left(t_1, \bar{u}^k\right)\right) \leq J\left(u^k\right) \stackrel{\triangle}{=} g\left(x^k(t_1)\right) = \zeta_k.$$

**Step 2** *Construct an approximation*

$$\Re_k = \Re\left(u^k, \epsilon_k\right) = \left\{\bar{y}^1, ..., \bar{y}^{N_k} / \; g(\bar{y}_i) = g\left(x\left(t_1, u^k\right)\right)\right\}$$

*of the level surface*

$$S(g,\zeta_k)=\{y\in R^n:\ g(y)=\zeta_k\}.$$

**Step 3** $\forall\, i=1,...,N_k$ *find a control* $v^i(\cdot)\in\mathcal{U}$ *s.t.*

$$\left\langle g'\left(\bar{y}^i\right),x\left(t_1,v^i\right)\right\rangle\geq\sup_{u\in\mathcal{U}}\left\langle g'\left(\bar{y}^i\right),x(t_1,u)\right\rangle-\epsilon_k.$$

**Step 4** $\forall i=1,...,N_k$ *construct points* $y^i,\ g(y^i)=\zeta_k$ *s.t.*

$$\left\langle g'\left(y^i\right),x\left(t_1,v^i\right)-y^i\right\rangle+\epsilon_k\geq\sup_y\left\{\left\langle g'(y),x\left(t_1,v^i\right)-y\right\rangle/\ g(y)=\zeta_k\right\}. \tag{3}$$

**Step 5** *Set*

$$\eta_k:=\left\langle g'\left(y^j\right),x\left(t_1,v^j\right)-y^j\right\rangle=\max_{1\leq i\leq N_k}\left\langle g'\left(y^i\right),x\left(t_1,v^i\right)-y^i\right\rangle.$$

**Step 6** *If* $\eta_k>0$, *then set*

$$\bar{u}^{k+1}(t):=v^j,\ t\in T,\ k:=k+1,$$

*and loop to Step 1.*

**Step 7** *If* $\eta_k\leq 0$, *where* $\epsilon_k>\chi$, *is a given tolerance, then stop. If* $\epsilon_k>\chi$, *loop to Step 1, with*

$$\bar{u}^{k+1}:=u^k,\ k:=k+1.\qquad\sharp$$

**Remark 1** *When* $\eta>0$ *(Step 6) we have due to the convexity of* $g(\cdot)$

$$\begin{gathered}0<\langle g'(y^j),x(t_1,v^j)-y^j\rangle\leq g(x(t_1,v^j))-g(y^j)=\\=g(x(t_1,v^j))-g\left(x\left(t_1,u^k\right)\right).\end{gathered}$$

*Thus, the control* $v^j(\cdot)\in U$ *is better than the stationary control* $u^k(\cdot)\in U$, *so that we have transcended stationary.*

**Test Examples**

Consider the norm maximization

$$\|x(t_1)\|^2\to\max \tag{4}$$

for the following simplest control system:
**Example 1** $\dot{x}_i=u_i,\ x_i(0)=0,\ i=1,...,n,\ -2\leq u_i(t)\leq 1,\ t\in[0,1]$. Each vertex of the reachable parallelogram $(x_i=x_i(t_1,u))P=\{x\in R^n:\ -2\leq x_i\leq 1,\ i=1,...,n\}$ is stationary.
$\dot{x}_1=x_2,\ x_1(0)=-1,\ \ \dot{x}_2=-x_1+u,\ x_2(0)=0,\ \ -1\leq u(t)\leq 1,$
$t\in[0,\pi]$. There are two stationary controls, but only one is global.
**Example 3** $\dot{x}_1=x_2+u_1,\ x_1(0)=3,\ \ \dot{x}_2=u_2,\ x_2(0)=-1,\ \ t\in[0,2],\ -1\leq u(t)\leq 1$. There exist three control $u(t)\equiv(1,1)^T$, $\hat{u}(t)\equiv-\tilde{u}(t)\equiv(1,-1)^T$, verifying maximum principle, but only $u(\cdot)$ is global optimal.

**Numerical Test**

We present here the numerical result of solving the above examples by global search algorithm from Sect.2. The global search was beginning after solving the linear problem:

$$\langle c, x\,(t_1, u)\rangle \to \max,\ \ u \in \mathcal{U}, \tag{5}$$

with a vector $c \in R^n$.

Let $J\,(u^0)$ and $J\,(u^*)$ be the initial and final values of objective functional, St be the number of stationary controls obtained, LP – the number of linearized problems solved during the numerical experiments and finally Time is the solving time in min:sec:subsec. All numerical tests were implemented using IBM PC/AT-386.

**Table 1**

| c | $J(U^0)$ | $J(U^*)$ | St | LP | Time |
|---|---|---|---|---|---|
| Example 2 | | | | | |
| $(1,1)^T$ | 6.5396 | 8,9987 | 2 | 18 | 1:18:36 |
| $(-1,1)^T$ | 1.00 | 8.9899 | 1 | 10 | 1:05:80 |
| Example 3 | | | | | |
| $(1,-3)^T$ | 10.00 | 26.00 | 3 | 12 | 0:39:25 |
| $(1,1)^T$ | 26.00 | 26.00 | 1 | 6 | 0:31:25 |
| $(-1,-1)^T$ | 18.00 | 26.00 | 2 | 10 | 0:32:12 |

For Example 1 we took $c = (1,1)^T$ and the following results have been obtained.

**Table 2**

| n | $J(U^0)$ | $J(U^*)$ | St | LP | Time |
|---|---|---|---|---|---|
| 5 | 5 | 20 | 6 | 27 | 0:11:21 |
| 10 | 10 | 40 | 11 | 77 | 1:20:46 |
| 20 | 20 | 80 | 21 | 252 | 12:24:95 |

**References**


[1] Pontryagin L.S. and al. *Mathematical Theory of optimal processes.* Moscow, Nauka, 1976.

[2] Moiseev N.N. *Numerical methods in optimal system theory.* Moscow, Nauka, 1971.

[3] Krotov V.F., Gurman V.I. *Methods and Problems of Optimal Control.* Moscow, Nauka, 1973.

[4] Strekalovsky A. *Extremal problems on complements of convex sets.* Translated from Kibernetika i Sistemmyi Analis. No. 1, pp. 113-126, 1993 Plenum Publishing Corporation.

[5] Strekalovsky A. *On non-convex optimal control problems.* Vestnik of Moscow University, seria "Computational Mathematics and Cybernetics". 1993, No 1, pp. 9-13.

[6] Strekalovsky A. *On Global Maximum of a Convex Terminal Functional in Optimal Control Problems.* Journal of Global Optimization (to appear).

[7] Strekalovsky A. *The Search for a Global Maximum of a Convex Functional on an Admissible Set.* Comput. Mathematics and Math. Physics, vol. 33, No. 3, pp. 315-328, 1993, Pergamon Press Ltd.

## Global Optimization (Systematic Approach Employing Peano Mappings)

Roman G. Strongin

The approach under consideration is based on reducing multidimensional multiextremal optimization problems to those of one dimension by applying space-filling curves mapping a unit interval on the real axis onto a multi-dimensional hyperinterval. The scheme employs several somehow conjugate (joint) Peano-type scannings which conduct the metric property of nearness of points in many dimension sufficiently better than in a case with a single Peano curve.

Following this approach a single multidimensional nonlinear program with Lipschitz (multiextremal) left-hand sides of constraints and of an objective function is reduced to a system of somehow connected one-dimensional problems admitting Holder continuations in a unit interval. The proposed technique for these one-dimensional constrained problems does not make use of penalties and less tedious. Each iteration in the suggested scheme involves the successive calculation of left-hand sides of the constraints. This calculation either interrupted by the occurrence of the constraint violation, or terminated with the objective function estimation (in the case of an admissible point).

The unknown Holder coefficients are adaptively estimated using the computed running lower bounds for the divided differences (with some idemfactor).

In the case when multiprocessor system is available, each one-dimensional problem is solved on a separate processor and the processors exchange the results of iterations. This corresponds to parallelization of the main problem.

A new method for scalarizing a multicriteria problem is also proposed so that the set of points of the absolute minimum of the scalar problem is identical to the Slater set of the initial problem.

Convergence conditions for all above cases are examined.

# A Method for Solving an Utility Function Program in Multiple Objective Nonlinear Optimization

Nguyen Van Thoai

Let $U : \mathbb{R}^k \to \mathbb{R}$ be an utility function according to a multiple objective programming problem of the form max $c_i(x) = z_i$ $(i = 1, \cdots, k)$, s.t. $x \in X \subset \mathbb{R}^n$. Consider the *utility function program* $\max\{U(z) : c_i(x) = z_i \ (i = 1, \cdots, k),\ x \in X\}$. Assuming that the utility function $U$ has the monotonocity property in a sense that $U(z^1) \leq U(z^2)$ for $z^1 \leq z^2;\ z^1, z^2 \in A$, where $A$ is a suitably large set containing the feasible region in criterion space of the multiple objective programming problem, we establish an algorithm of branch and bound type for solving the above utility function program. The algorithm is implemented for several types of objective functions $c_i$ and feasible set $X$ of the multiple objective programming problem under consideration.

## Quadratic Programming with Box Constraints

Gerardo Toraldo and Panos M. Pardalos

Quadratic programming problems with box constraints have the following form:

$$\min_{x \in I\!R^n} \frac{1}{2} x^T Q x + c^T x \ \text{ s.t. } \ l \leq x \leq u. \tag{1}$$

Here $Q$ is an $n \times n$ symmetric matrix, $c$, $l$ and $u$ are known $n$-vectors. Problems of this form have numerous applications in engineering and include as special cases binary quadratic programming and linear complementarity problems. In addition, quadratic programming with box constraints is a fundamental subroutine in many nonlinear optimization packages (such as the LANCELOT package).

In the nonconvex case, several types of algorithms have been proposed. Nonconvex problems may have an exponential number of local solutions and stationary points, and from the complexity point of view are NP-hard.

Since every nonconvex quadratic function can be decomposed as a sum of a convex and a concave function, algorithms for the convex case can be used for computing approximate solutions of the general case. Two main approaches have been proposed for the problem (1) when $Q$ is positive semidefinite, the projected-gradient active set approach, and the interior point approach. Both of these methods seem to be more efficient than the classical active set method and are suitable for sparse and large scale problems.

In this talk we will review some of the more recent theoretical results and computational algorithms for the general (concave and indefinite) box constrained quadratic programming problem.

# Reverse Convex Programming. Theory and Algorithms.

A.S. Strekalovsky and Ider Tsevendorj

We consider a non-convex mathematical programming problem with a reverse convex constraint. A global search algorithms based on global optimality conditions is proposed. One test problem of high dimension is solved.

Global Optimization (GO) has remained marginal for a long time, this is still the case to a certain extend (Horst and Tuy 1990). On the other hand Optimality Conditions are one of the cornerstones in Convex and Local Optimization Theory and of paramount importance in the construction of corresponding algorithms. But in GO the situation with Global Optimality Conditions (GOC) was hopeless until the works J.-B. Hiriart-Urruty (1989) and of the Strekalovsky (KiSA 1993, IFIP 1993). Now the crucial question is how to use the obtained GO Conditions for constructing GO Algorithms. This paper deals with so called reverse convex problem (Horst and Tuy 1990):

$$f(x) \to \min, \quad x \in S, \quad g(x) \geq 0 \tag{P}$$

where $g(\cdot)$ is a convex function over $R^n$, $f : R^n \to R$ and $S$ may also be convex. It is clear, that feasible region of $(P)$ is not convex, even in the case $S$ is convex. As a result, problems with such constraints generally have local optima, which are not the global ones.

Having a wide class of applications (Horst and Tuy 1990) the problem $(P)$ remains unsolved till now.

Here we display the results of using the approach based on Global Optimality Conditions (Strekalovsky KiSA 1993), and the $R$-algorithm for $(P)$, presented for the first time in (Strekalovsky IFIP 1993).

## $R-$Algorithms

This algorithms is based on the notion of resolving set (Strekalovsky IFIP 1993, CMaMP 1993), which is related to using the level surface

$$LS(g) = \{y \in R^n : \ g(y) = 0\}$$

in the (GOC) for $(P)$ (Strekalovsky KiSA 1993).
Let an admissible point $x^0 \in S, \ g(x^0) \geq 0, \ k := 0.$

1. Beginning from $x^k$ obtain a $\varepsilon_k-$stationary point $z^k \in S, \ g(z^k) = 0.$

2. In order to decide whether a $\varepsilon_k-$stationary point $z^k$ is a global solution, take a finite number of points from $LS(g)$ $(\varepsilon_k \downarrow 0)$ :

$$R_k = \{v^1, ..., v^N \ / g(v^i) = g(z^k), \ \ i = 1, .., N, \ \ N = N(z^k)\}$$

   instead of using all points from $LS(g)$, what is obviously impossible.

3. Solve $\forall i = 1, .., N$, following linearized problems:

$$\langle g'(v^i), x\rangle \to \max, \quad x \in S, \quad f(x) \leq f(z^k), \tag{$P_i$}$$

   which are convex, when $f$ and $S$ are convex, while the problem $(P)$ remains non-convex even under the assumptions above. Let $u^i$ be a $\delta_k$-solution of $(P_i)(\delta_k \downarrow 0)$.

4. Solve so-called level problem $(\forall i = 1, .., N)$

$$\langle g'(v), u^i - v\rangle \rightarrow \ max, \ \ g(v) = g(z^k), \qquad (PL_i)$$

Let $w^i$ be a $\delta_k$-solution of $(PL_i)$.

5. Consider the number

$$\eta_k = \langle g'(w^j), u^j - w^j\rangle = max_i\{\langle g'(w^i), u^i - w^i\rangle \ / \ i = 1, .., N\}.$$

6. If $\eta_k > 0$ , then set $x^{k+1} := u^j$. And go to step 1.

7. If $\eta_k \leq 0$ and $\quad \varepsilon_k, \delta_k \leq \chi \quad$ ,where $\chi$ is suitable tolerance, then stop.

The global convergence of the algorithm has been proved for example for the case of quadratic $g(\cdot)$ under assumption that $R_k$ is resolving set $\forall k = 0, 1, 2...$,i.e. if the inequality

$f(z^k) > f_* + \varepsilon_k$

(where $f_* = inf\{f(x) \ / \ x \in S, \ \ g(x) \geq 0\}$) implies two following inequalities:

$(i) \quad \eta_k > 0,$

$(ii) \quad \eta_k + \delta_k \geq \ sup_{x,v}\{\langle g'(v), x - v\rangle \ / \ x \in clcoS, \ f(x) \leq f(z^k), \ g(v) = g(z^k)\}$

**Numerical Tests**

Now consider the problem similar to this one from (Gurlitz and Jacobsen 1991) (in order to obtain a comparative example):

$$f(x) \stackrel{\triangle}{=} \frac{1}{2}\|x - y\|^2 \rightarrow \min, \qquad (1)$$

$$x \in \Pi \stackrel{\triangle}{=} \{x \in R^n \ / \ -1 \leq x_i \leq 1, \ i = 1, .., n\} \qquad (2)$$

$$g(x) \stackrel{\triangle}{=} \|x\|^2 - (n - 0.5), \qquad (3)$$

$y = (-0.25, 1, ..., 1)^\top \in R^n.$

It's easy to see that $x_* = (-\sqrt{0.5}, 1, ..., 1)^\top$ is the solution of the problem with the value $f(x_*) = 0.104$. The worse initial point is $x^0 = (1, -1, ..., -1)^\top$. As displayed in (Gurlitz and Jacobsen 1991) the H. Tuy cuts method is not able to solve this problem beyond the dimension 10, what is not practical.

We present here the first results of numerical solving the problem (1)-(3) of the dimension till 400.

But before it should be said, that the stationary search and the solving the problem $(P_i)$ have been done by the simplest procedure of optimization, taking into account the data nature of the problem (1)-(3).

We intend to ameliorate this part of the programme. It was not very simple, while the solution of the level problem $(PL_i)$ has been obtained analytically (Strekalovsky CMaMP 1993).

Since it is not yet proved the resolving set for (1)-(3), during the tests we used the following level set approximations:

$R1 = \{v^i = (z_1, ..., z_{i-1}, -z_i, z_{i+1}, ..., z_n)^\top \ , \ i = 1, ..., n\},$

$$R20 = \left\{ \begin{array}{c} v^i = (z_1, ..., z_{i-1}, -z_i, -z_{i+1}, z_{i+2}, ..., z_n)^\top \ , \ i = 1, ..., n-1, \\ v^n = z - 2 \cdot f'(z) \cdot \langle f'(z), z \rangle \ / \ \|f'(z)\|^2 \end{array} \right\}.$$

Let $n$ be dimension of the problem, $f(x^0)$ the initial value of the function, $R$-the choice of level set approximation, $f(x^m)$-the obtained function value, $St$-the number of obtained stationary points, from what we managed to exit, $LP$-the number of the linearized problem $(P_i)$ solved during the process, and finally $T$ be the time of solving (min:sec). The tests have been done using the PC/AT IBM-386.

| $n$ | $f(x^0)$ | $R$ | $f(x^m)$ | $St$ | $LP$ | $T$ |
|---|---|---|---|---|---|---|
| 20 | 38.781 | $R1$ | 0.104 | 21 | 230 | 00:14.01 |
| | | $R20$ | 0.104 | 11 | 120 | 00:08.68 |
| 40 | 78,781 | $R1$ | 0.104 | 41 | 860 | 01:01.36 |
| | | $R20$ | 0.104 | 21 | 440 | 00:37.30 |
| 60 | 118.781 | $R1$ | 0.104 | 61 | 1890 | 02:39.45 |
| | | $R20$ | 0.104 | 31 | 960 | 01:35.89 |
| 80 | 158.781 | $R1$ | 0.104 | 81 | 3320 | 05:27.30 |
| | | $R20$ | 0.104 | 41 | 1680 | 03:18.23 |
| 100 | 198.781 | $R1$ | 0.104 | 101 | 5150 | 09:33.97 |
| | | $R20$ | 0.104 | 51 | 2600 | 05:47.46 |
| 150 | 298.781 | $R20$ | 0.104 | 76 | 5775 | 16:27.12 |
| 200 | 398.781 | $R20$ | 0.104 | 101 | 10200 | 34:02.52 |
| 300 | 598.781 | $R20$ | 0.104 | 151 | 22800 | 1hr:44:34.80 |
| 400 | 798.781 | $R20$ | 0.104 | 201 | 25136 | 3hr:58:18.65 |

**References**


[1] Gurlitz, T.R. and Jacobsen, S.E. (1991) On the use of Cuts in Reverse Convex Programs., Journal of Optimization Theory and Applications, v. 68,pp.257-274.

[2] Hiriart-Urruty, J.-B. (1989) From convex optimization to non-convex optimization. Part 1: Necessary and sufficient conditions for global optimality, in Nonsmooth Optimization and Related Topics. Plenum Press, pp. 219-239.

[3] Horst, R. and Tuy, H. (1990) Global Optimization (Deterministic Approach), Springer-Verlag.

[4] Strekalovsky, A.S.(1993) Extremal problems on complements of convex sets., Translated from Kibernetika i Sistemnyi Analiz, 1, pp. 113-126. (KiSA)

[5] Strekalovsky, A.S. (1993) The search for a global maximum of a convex functional on an admissible set, Comput. Math. and Math Physics, vol. 33, No 3, pp. 315-328, Pergamon press Ltd.(CMaMP) bibitem6 Strekalovsky, A.S. (1993) Global Optimization Algorithms corresponding Global Optimality Conditions. 16th IFIP Conference on System Modelling and Optimization, July 5-9, 1993, Compiegne, France, Collection of Abstracts, vol 2, pp. 825-828, (IFIP)

# Global Optimization for Imprecise Problems

M.N. Vrahatis, D.G. Sotiropoulos and E.C. Triantafyllou

Several methods for finding the extrema of a function $f: \mathcal{D} \subset I\!R^n \to I\!R$, where $\mathcal{D}$ is open and bounded, have been proposed with many applications in different scientific fields (mathematics, physics, engineering, computer science etc.). Most of them require precise function and gradient values. In many applications though, precise values are either impossible or time consuming to obtain. For example, when the function and gradient values depend on the results of numerical simulations, then it may be difficult or impossible to get very precise values. Or, in other cases, it may be necessary to integrate numerically a system of differential equations in order to obtain a function value, so that the precision of the computed value is limited, [8, 15, 14]. On the other hand, it is necessary in many applications to use methods which do not require precise values [4, 15], as for example in neural networks training [9, 10].

In this paper a new method is presented for the computation of the global minimum $x^*$ in the box $\mathbf{X}$ such that :

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

where $f$ has continuous first and second derivatives. This method can be applied to problems with imprecise function and gradient values and it is composed of two parts.

In the first part, interval arithmetic [2, 11] is implemented for a "rough" isolation of all the extrema of $f$ in various boxes (not necessarily too small). So, although this phase of our algorithm uses extended interval arithmetic, it can be applied to problems with unavoidable inaccuracies.

Our method starts with an initial box $\mathbf{X}^{(0)} \in I\!I^n$ and it is based on the branch and bound principle. When the total number of the extrema is known, our algorithm is more efficient. One way to estimate this number is by using degree computational techniques [5]. For this purpose one can apply Picard's theorem and compute the value of the topological degree of the extended Picard's function [4]. For the computation of this value Aberth's method [1], which is an adaptation of Kearfott's method [7], can be utilized to boxes with interval arithmetic.

The stationary points of a function $f(x)$ in the box $\mathbf{X}^{(0)}$ are the zeros of the set of equations :

$$\nabla f(x) = \mathcal{O} = (0, 0, \ldots, 0), \tag{2}$$

where $\nabla f = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n}\right)$ denotes the gradient of $f$. Thus, using the information about the total number of extrema as an upper bound of how many sub–boxes must be investigated, the initial box is divided into smaller sub–boxes. In this way the given region is covered by a set of small boxes where the range of values of $f(x)$ is more precise [11].

Also our algorithm is based on Kearfott's root inclusion test [6], which assigns to each box the values "true", "false" or "unknown". Specifically, we perform one step of the extended interval Newton Gauss–Seidel method to the nonlinear system (2). When we apply the test to each sub–box three cases can be distinguished. First, if the test result is "true", then this sub–box contains a unique stationary point and is stored in a new list $\mathcal{L}$ (which is initially empty). If the test result is "false", this indicates that the sub–box contains no stationary points and we may discard it. When the test result is "unknown", we bisect this box or not, according to the length of its diameter.

In the second phase of the algorithm, a new criterion is used in order to characterize the isolated stationary points as minima, maxima or saddle points. Specifically, the concept of the characteristic $n$–polyhedron (CP) is implemented. Let us define a characteristic polyhedron by constructing the $2^n \times n$ matrices $\mathcal{M}_n$ whose rows are formed by all possible combinations of $-1$, 1. The $n$–polyhedron $\Pi^n = \langle \Upsilon_1, \Upsilon_2, \ldots, \Upsilon_{2^n} \rangle$ in $I\!R^n$ is called a *characteristic polyhedron*

*(CP) relative to* $\nabla f$ if the matrix of signs associated with $\nabla f$ and $\Pi^n$, $\mathcal{S}(\nabla f;\, \Pi^n)$, is identical with the $n$–complete matrix $\mathcal{M}_n$, [16]. In other words, the signs of the components of $\nabla f$ at the $2^n$ vertices of $\Pi^n$ obtain every combination of $\pm 1$.

If $\Pi^n$ is a CP then, under suitable assumptions on the boundary of $\Pi^n$, the value of the topological degree of $\nabla f$ at $\mathcal{O}$ relative to $\Pi^n$ is nonzero which implies the existence of a stationary point inside $\Pi^n$, [12, 13, 16].

The previous characterization of each stationary point is done according to the orientation of the characteristic $n$–polyhedron [17]. The above procedure makes use only of the algebraic sign of $\nabla f$, while derivatives of $\nabla f$ or approximations of them are not required.

Next the algorithm chooses those points characterized as minima and computes all of them to obtain the global one. To this end it uses a generalized bisection method which requires only the signs of the gradient values to be correct and thus it can be applied to imprecise problems. Also it is globally convergent method and can be applied to non–differentiable continuous functions [12, 13, 16].

This generalized bisection method, used in combination with the CP–criterion outlined above, bisects a CP, in such a way that the new refined $n$–polyhedron is also a CP. To do this, one computes the midpoint of a proper 1–simplex (edge) of $\Pi^n$ and uses it to replace that vertex of $\Pi^n$ for which the vectors of their signs are identical (see [12, 13, 16] for details). Finally, the number $B$ of characteristic bisections of the edges of a $\Pi^n$ required to obtain a new refined CP, $\Pi^n_\star$, whose longest edge length, $\Delta(\Pi^n_\star)$, satisfies $\Delta(\Pi^n_\star) \leq \varepsilon$, for some $\varepsilon \in (0,1)$, is given by :

$$B = \left\lceil \log_2(\Delta(\Pi^n)\, \varepsilon^{-1}) \right\rceil , \tag{3}$$

where the notation $\lceil \cdot \rceil$ refers to the smallest integer, which is not less than the real number quoted (see [12] for a proof).

## References


[1] Aberth O., Computation of topological degree using interval arithmetic, and applications, *Math. Comp.*, **62,** 171–178 (1994).

[2] Alefeld G. and Herzberger J., *Introduction to Interval Computations*, Translated by J. Rokne, Academic Press, New York (1983).

[3] Hansen E.R., *Global Optimization using Interval Analysis*, Marcel Dekker, Inc., New York (1992).

[4] Kavvadias D.J. and Vrahatis M.N., Locating and computing all the simple roots and extrema of a function, *SIAM J. Sci. Comput.*, in press.

[5] Kearfott R.B., An efficient degree–computation method for a generalized method of bisection, *Numer. Math.*, **32,** 109–127 (1979).

[6] Kearfott R.B., Some tests of generalized bisection, *ACM Trans. Math. Software*, **13,** 197–220 (1987).

[7] Kearfott R.B. and Novoa M., INTBIS, a portable Interval Newton/bisection package, *ACM Trans. Math. Software*, **16,** 152–157 (1990).

[8] Kupferschmid M. and Ecker J.G., A note on solution of nonlinear programming problems with imprecise function and gradient values, *Math. Program. Stud.*, **31**, 129–138 (1987).

[9] Magoulas G.D., Vrahatis M.N., Grapsa T.N. and Androulakis G.S., Neural network supervised training based on a dimension reducing method, *Ann. Math. Artif. Intel.*, in press.

[10] Magoulas G.D., Vrahatis M.N., Grapsa T.N. and Androulakis G.S., An efficient training method for discrete multilayer neural networks, *Ann. Math. Artif. Intel.*, in press.

[11] Moore R., Hansen E. and Leclerc A., *Rigourous methods for global optimization*, in: Recent Advances in Global Optimization, Ch.A. Floudas and P.M. Pardalos eds., Princeton University Press, Oxford, pp. 321–342, (1992).

[12] Vrahatis M.N., Solving systems of nonlinear equations using the nonzero value of the topological degree, *ACM Trans. Math. Software*, **14,** 312–329 (1988).

[13] Vrahatis M.N., CHABIS: A mathematical software package for locating and evaluating roots of systems of non–linear equations, *ACM Trans. Math. Software*, **14,** 330–336 (1988).

[14] Vrahatis M.N., *A generalized bisection method for large and imprecise problems*, in: Scientific Computing and Validated Numerics, G. Alefeld and A. Frommer eds., Akademie Verlag, to appear.

[15] Vrahatis M.N., Androulakis G.S. and Manoussakis G.E., A new unconstrained optimization method for imprecise function and gradient values, *J. Math. Anal. Appl.*, in press.

[16] Vrahatis M.N. and Iordanidis K.I., A rapid generalized method of bisection for solving systems of nonlinear equations, *Numer. Math.*, **49,** 123–138 (1986).

[17] Vrahatis M.N. and Triantafyllou E.C., *Locating, characterizing and computing the stationary points of a function*, in: Scientific Computing and Validated Numerics, G. Alefeld and A. Frommer eds., Akademie Verlag, to appear.

## Smoothing Transform and Continuation for Global Optimization

Jorge Moré and Zhijun Wu

We discuss the smoothing techniques for global optimization and their applications in macromolecular modeling and simulation. We focus on issues associated with the solution trajectories determined by the smoothing transform, and discuss some of our recent theoretical and computational studies that lead to better understanding of the smoothing techniques and development of more efficient global continuation algorithms.

## Complexity Analysis Integrating PAS, PRS and Simulated Annealing

Zelda B. Zabinsky and Birna P. Kristinsdottir

The complexity of random search algorithms has been studied for global optimization with the hope of understanding their behaviour and hence developing more efficient algorithms. Pure adaptive search (PAS) has been analyzed for both continuous and discrete finite global optimization [2, 4, 5, 6]. The complexity of PAS has been shown to be "linear" in the dimension of a continuous global optimization problem satisfying a Lipschitz condition [6] and has an analogous result for a finite global optimization problem [5]. Pure adaptive search is an idealistic algorithm because it is defined to generate a sequence of feasible points that are sampled according to a probability distribution that is restricted to the region of improving objective function values. This stipulation makes the method impractical at this point in time to implement efficiently. In contrast, pure random search (PRS) samples points according to a fixed probability distribution with no such restriction on improving points and is readily implementable. As shown in [4] the complexity of PRS is exponentially greater than PAS to solve an equivalent problem.

To attempt to analyze a more realistic algorithm, in this paper we examine a combination of PAS with PRS to allow the algorithm to generate both improving points as well as non-improving points. We also add a probability of accepting a non-improving point, as is commonly done in simulated annealing algorithms. We use a Markov chain analysis for a general analysis and then evaluate some special cases to gain insight into the value of generating improving points and the tradeoffs of accepting non-improving points.

### Markov Chain Analysis

We consider the following finite global optimization problem:

$$\text{minimize} \quad f(x) \tag{1}$$

$$\text{subject to} \quad x \in X \tag{2}$$

where $f(x)$ is a real valued function on a finite set $X$. We let $y_1 < y_2 < \ldots < y_M$ be distinct objective function values. Notice that there may be more than $M$ points in $X$. In keeping with the notation in [5], for $m = 0, 1, \ldots$, let the random variable $Y_m$ be the objective function value on the $m$th iteration of PRS. Note that $Y_0, Y_1, \ldots$ are independent and identically distributed. Pure random search samples the domain according to fixed probability distribution, $\mu$ on $X$. Given this sampling distribution, we define a probability measure $\pi = (\pi_1, \ldots, \pi_M)$ on the range of $f$ as follows. Let $\pi_j$ be the probability that any iteration of pure random search attains a value of $y_j$. That is $\pi_j = P(Y_0 = y_j) = \mu(f^{-1}(y_j))$ for $j = 1, 2, \ldots, M$. Throughout this paper $p_j$ denotes $\sum_{i=1}^{j} \pi_i$ the probability that PRS attains a value of $y_j$ or less.

The algorithm analyzed here has a probability $p$ of sampling according to PAS and probability $1 - p$ of sampling according to PRS, $0 \leq p \leq 1$, both based on the same arbitrary distribution. This paper always refers to "weak PAS" as stated in [5]. A similar development is possible for "strong PAS", but is not included here. In addition, we include a probability of accepting a non-improving point which is intended to be similar to simulated annealing. We define $t_{ij}$ as the probability of accepting a point with objective function value $y_j$ when sampled from $y_i$, for $i, j = 1, \ldots, M$. If $j \leq i$, we assume $t_{ij} = 1$ because we always want to accept an improving point. We also assume $t_{1j} = 0$ for $j = 2, \ldots, M$ such that we never leave the global minima.

We can now define the Markov chain to model the optimization algorithm. The states of the Markov chain represent the objective function values, $y_1, \ldots, y_M$, where state $y_1$ represents the global optimum. The initial probability distribution for the initial state is given by $\pi$. In standard Markov chain terminology [3], $y_1$ is the absorbing state of this chain and all other states are transient. We say the algorithm converges when the chain reaches the absorbing state.

We let $S$ be the one-step transition matrix that models the algorithm, so $s_{ij}$ is the probability that the algorithm moves from state $y_i$ to state $y_j$. This transition probability incorporates the probability of sampling according to PAS or PRS, as well as the probability of accepting the sampled point even if it is not improving. The matrix $S$ has the following entries:

$$s_{ij} = \begin{cases} (1-p)\pi_j t_{ij} & \text{if} \quad i < j \\ (1-p)\pi_j + p(\pi_j/p_i) + \sum_{k=i+1}^{M}(1-p)\pi_k(1-t_{ik}) & \text{if} \quad i = j \\ (1-p)\pi_j + p(\pi_j/p_i) & \text{if} \quad i > j \end{cases}$$

The expected number of iterations to absorption can be expressed in terms of the transition matrix of the Markov chain. This expected number of iterations to absorption indicates the average computational effort to sample the global optimum but not necessarily to confirm it. Let $v_i$ be the expected number of iterations until absorption, starting in state $i$, $i = 1, \ldots, M$. Then the expected number of iterations until absorption $v$, can be found by solving the system of equations

$$v = (I - Q)^{-1}e$$

where $Q$ consists of the first $(M-1)$ rows and $(M-1)$ columns of $S$, $I$ denotes an $(M-1)\times(M-1)$ identity matrix and $e$ is an $M-1$ vector of ones. The variance of the number of iterations until absorption is also obtainable from the fundamental matrix $(I-Q)^{-1}$ (see [3] page 49).

**Special Cases**

The previous section provided the expression for the expected number of iterations until convergence for a combination of PAS and PRS with an arbitrary distribution, and general acceptance probability. We now turn to two special cases. Both cases assume a uniform distribution, with $\pi_j = 1/M$ and $p_i = i/M$. The first case never accepts a non-improving point, while the second case allows non-improving points to be accepted according to a probability similar to that used in simulated annealing.

Uniform distribution and only improving points accepted:

We now consider the special case where we only accept improving points, and therefore $t_{ij} = 1$ if the point is improving, $i \geq j$, and 0 otherwise. This assumption coupled with uniform sampling leads to a simplified transition matrix. This allows us to derive an analytical expression for the expected number of iterations, and provide some simple bounds. The results are stated below without proofs.

**Theorem 1** *The expected number of iterations to converge to the global optimum starting in state $y_M$ is*

$$v_M = \frac{M}{M-1} + M\sum_{j=1}^{M-2}\frac{1}{(M-1-j)(M-j(1-p))}. \tag{3}$$

**Corollary 1** *The expected number of iterations to converge to the global optimum starting in state $y_M$ is bounded above by*

$$v_M \leq 1 + \frac{qM}{M+(q-1)}(1+\log(M-1)) \tag{4}$$

*where $q = 1/p$.*

Theorem 1 gives the exact number of iterations required to solve the global optimization problem from the worst state in terms of $p$ and $M$. An upper bound on the number of iterations required to solve the global optimization problem is stated in Corollary 1.

PAS and PRS are two extreme algorithms. PAS is the best possible algorithm we could hope to have and PRS is an inefficient "blind" algorithm. An algorithm that is a mixture of these two would have a complexity somewhere in between. The expression in Theorem 1 agrees with the expression given in [2] when $b(y)$ is chosen to reflect the combination of PAS and PRS, which is done by choosing $b(y) = (1-p)p_{j-1} + p(\frac{p_{j-1}}{p_j})$. Figure 1 shows the expected number of iterations, plotted against the number of states for various values of $p$. When $p = 0.0$ we have PRS, and when $p = 1.0$ we have PAS. From the figure we can see how the expected number of iterations required to converge to the global optimum changes as the probability we sample according to PAS changes. It is interesting to see that the expected number of iterations changes slowly as $p$ changes. This shows that one needs only a small probability of sampling in the improving region in order to dramatically improve performance. Also there is a diminishing return as $p$ exceeds 0.5 and gets closer to 1. Many practical algorithms have some probability of sampling in the improving region, which is analogous to sampling according to PAS. We hope that this analysis will be helpful in understanding why many such algorithms have good performance.

In [6] it is proven that the complexity of PAS is linear in dimension, and an analogous complexity result for finite PAS is developed in [5] on an $m$-dimensional lattice $\{1, \ldots, k\}^m$ with distinct objective function values. The domain for the lattice has $M = k^m$ number of states. The following corollary give an upper bound on the expected number of iterations to solve the lattice optimization problem with the combination of PAS and PRS, and proves that the number of iterations to solve the global optimization problem is proportional to the dimension of the problem $m$ by a constant $1/p$.

**Corollary 2** *For an $m$-dimensional lattice $\{1, \ldots, k\}^m$ with distinct objective function values, the expected number of iterations to converge to the global optimum is bounded above by*

$$v_M \leq 1 + (1/p) + (m/p) \log k. \tag{5}$$

Uniform distribution and accept non-improving points:

The second special case again assumes a uniform sampling distribution and also allows a non-improving point to be accepted. We introduce the acceptance probability $t_{ij}$ as follows:

$$t_{ij} = \begin{cases} e^{-(j-i)/T} & \text{if} \quad i < j \quad \text{not improving} \\ 1 & \text{if} \quad i \geq j \quad \text{improving} \end{cases}$$

where the constant $T$ denotes the temperature as in a standard acceptance probability for simulated annealing [1]. The transition matrix for the Markov chain analysis simplifies in this case, and we numerically solve for the expected number of iterations until convergence.

In figure 2, the expected number of iterations until absorption is graphed for fixed temperature values and $p = 0.25$. The graph illustrates how the probability of accepting a non-improving point compares to the ideal PAS situation. It is interesting that when $T$ is very close to zero, then this special case is essentially the same as the previous special case because it is never accepting a non-improving point. And as $T$ grows without bound, the complexity will also grow dramatically as compared with the ideal of PAS. Our research is continuing to explore the effects of temperature on the complexity of this type of random search algorithm. To bridge the gap to analyze more realistic algorithms, we are generalizing the framework to modify the sampling distribution, $\pi$ as a function of temperature and objective function value.

## Summary/Conclusions

We have presented a Markov chain analysis for a random search algorithm over finite global optimization problems. The random search algorithm is a combination of PAS and PRS with

a probability of accepting a non-improving point that is motivated by simulated annealing. Two special cases are examined, which provide some insight into the behaviour of this type of algorithm.

**References**


[1] Aarts, E., and Korst, J., *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*, John Wiley and Sons, New York, 1989.

[2] Bulger, David, and Wood, G.R., *Hesitant Adaptive Search for Global Optimisation*. Technical report from Central Queensland University, Australia, March 1995.

[3] Kemeny, J.G., and Snell, J.L., *Finite Markov Chains*, Springer-Verlag, New York, 1976.

[4] Patel, N.R., Smith, R.L., and Zabinsky, Z.B., "Pure adaptive search in Monte Carlo optimization," *Mathematical Programming 43* (1988) 317-328.

[5] Zabinsky, Z.B., Wood, G.R., Steel, M.A., and Baritompa, W.P., "Pure adaptive search for finite global optimization", forthcoming in Mathematical Programming.

[6] Zabinsky, Z.B., and Smith, R.L., "Pure adaptive search in global optimization", *Mathematical Programming 53* (1992), 323-338.


Figure 1: Expected number of iterations to converge to the global optimum, where $p$ is the probability of picking a point according to PAS.

Figure 2: Expected number of iterations to converge to the global optimum using p=0.25 and varying the temperature $T$.

## Global optimization and visualisation of multidimensional data

Antanas Žilinskas

Multidimensional scaling in Euclidean space means fitting distances to given dissimilarities by weighted least squares. The corresponding objective function called STRESS is generally non-differentiable and has many local minima. On the other hand STRESS is defined by a rather simple analytical formula as well as the gradient of STRESS (where it exists). We start with the general discussion on the possibilities of minimization of STRESS by various global optimization techniques. The conclusion: global technique should include a local descent subalgorithm. It is proved that local descent trajectories never cross the subsets of nondiferentiability of STRESS. Therefore, for local search it is reasonable to choose a gradient based method. A local minimization method is proposed taking into account the specific features of the constraints to the local subproblem. The global algorithm controlling the local searches is a version of evolution strategy. The pros and cons of two- and three-dimensional scaling are discussed. The use of stereoscopic techniques to visualize the results of three dimensional scaling is demonstrated. There is twofold relations between scaling and global optimization. We have discussed the application of global optimization in constructing of scaling methods. But two/three dimensional scaling is important to visualization of the global search as well.

## The Graph Partitioning Problem and the Nodal Properties of the Eigenvectors of the Laplacian

Patrizio Cintioli, Pierluigi Maponi, Donatella Ponziani and Francesco Zirilli

In recent years there has been a great deal of interest in using methods and results of continuous mathematics in discrete mathematics. The work presented here belongs to this set of ideas. We consider the *Graph Partitioning* problem and its formulation as a (0,1) constrained quadratic programming problem. It is well known that estimates on the *Graph Partitioning* problem can be obtained from the knowledge of the eigenvalues of the laplacian associated to the graph.

We deal with the problem of computing these eigenvalues. First of all we generalize to the graph contest some properties that hold for the eigenvalue problem associated to the classical laplacian on a bounded domain of $\mathbb{R}^n$ with Neumann boundary conditions. These properties include some monotonicity properties of the eigenvalues and some "nodal properties" of the eigenvectors. Using these properties some special techniques to compute the eigenvalues of the laplacian associated to the graph are proposed and tested on a significant set of test problems.

**Al-Mharmah, Hisham** Industrial Engineering Department, University of Jordan, Amman, Jordan, E-mail: —

**Berner, Sonja** Bergische Universität GH Wuppertal, Fachbereich Mathematik, Gaußstr. 20, D-42097 Wuppertal, Germany, E-mail: sonja@math.uni-wuppertal.de

**Bollweg, Wilfried** Institut für numerische und Instrumentelle Mathematik, Münster, Germany, E-mail: bollweg@math.uni-muenster.de

**Bomze, Immanuel M.** Dept. of Statistics, Operations Research and Computer Science, University of Vienna, Universitaetsstrasse 5, A-1010 Wien, Austria, E-mail: bomze@eos.smc.univie.ac.at

**Calvin, James** School of System and Industrial Engineering, Georgia Institute of Technology, Atlanta - GA 30332, USA, E-mail: calvin@isye.gatech.edu

**Carrizosa, Emilio** Facultad de Matematicas, Universidad de Sevilla, c/ Tarfia s/n. 41012 Sevilla, Spain, E-mail: ecarriz@obelix.cica.es

**Csallner, András Erik** József Attila University, Institute of Informatics, H-6720 Szeged, Árpád tér 2, Hungary, E-mail: csallner@inf.u-szeged.hu

**Csendes, Tibor** József Attila University, Institute of Informatics, H-6720 Szeged, Árpád tér 2, Hungary, E-mail: csendes@inf.u-szeged.hu

**Dallwig, Stefan** Institut für Mathematik, Universität Wien, Strudlhofgasse 4, A-1090 Wien, Austria, E-mail: sdal@cma.univie.ac.at

**Fehér, Ede** University of Veszprém, Dept. of Applied Informatics, H-8000 Veszprém, Egyetem u. 2, Hungary, E-mail: feherede@hal2000.vein.hu

**Floudas, Christodoulos A.** Department of Chemical Engineering, Princeton University, Princeton, NJ 08544-52633, USA, E-mail: floudas@titan.princeton.edu

**Friedler, Ferenc** University of Veszprém, Department of Computer Science, H-8200 Veszprém, Egyetem u. 10, Hungary, and Research Institute of Chemical Engineering, Hungarian Academy of Sciences, Veszprém, Egyetem u. 2. E-mail: friedler@almos.vein.hu

**Fülöp, János** Computer and Automation Institute, Hungarian Academy of Sciences, Budapest, Hungary, E-mail: h635ful@ella.hu

**Garcia, Inmaculada** Dpto. Arquitectura de Computadores y Electrónica, Universidad de Almería. 04120-Almería. Spain, E-mail: inma@iron.ualm.es

**Geramitchioski, Tale** Faculty of Engineering, University of Bitola, Macedonia, E-mail: —

**Gergel, Victor** Nizhni Novgorod State University, 23, Gagarin Prosp., Nizhni Novgorod, 603600, Russia, E-mail: gergel@nnucnit.nnov.su

**Hendrix, Eligius** Department of Mathematics, Wageningen Agricultural University, Wageningen, The Netherlands, E-mail: eligius@rcl.wau.nl

**Hichert, Jens** TU Ilmenau, Fakultaet für Mathematik und Naturwissenschaften, PF 327, D-98648 Ilmenau, Germany, E-mail: jens.hichert@mathematik.tu-ilmenau.de

**Hoffmann, Armin** TU Ilmenau, Fakultaet für Mathematik und Naturwissenschaften, PF 327, D-98648 Ilmenau, Germany, E-mail: armin.hoffmann@mathematik.tu-ilmenau.de

**Holmqvist, Kristina** Division of Optimization, Department of Mathematics, Linköping Institute of Technology, S-581 83 Linköping, Sweden, E-mail: krhol@math.liu.se

**Horst, Reiner** University of Trier, Department of Mathematics, D–54286 Trier, Germany, E-mail: horst@uni-trier.de

**Imreh, Balázs** József Attila University, Institute of Informatics, H-6720 Szeged, Árpád tér 2, Hungary, E-mail: imreh@inf.u-szeged.hu

**Jones, Donald** General Motors R&D Center, Warren, MI, USA, E-mail: djones@cmsa.gmr.com

**Khamisov, Oleg V.** Siberian Energy Institute, Irkutsk, Russia, E-mail: root@sei.irkutsk.su

**Korotkich, Victor** Department of Mathematics and Computing, Central Queensland University, Rockhampton, Queensland 4702, Australia, E-mail: v.korotkich@cqu.edu.au

**Kovács, Zoltán** Department of Computer Science, JGYTF, Szeged, Hungary, E-mail: kovacs@kurrah.cab.jgytf.u-szeged.hu

**Locatelli, Marco** Dip. Scienze della Informazione, via Comelico 39/41 - 20139 Milano, Italy, E-mail: locatelli@hermes.mc.dsi.unimi.it

**Madsen, Kay** Technical University of Denmark, Lyngby, Denmark, E-mail: km@imm.dtu.dk

**Martindale, J.M.** Kluwer Publisher, Boston, USA, E-mail: jmkluwer@world.std.com

**Mockus, Jonas** Institute of Mathematics and Informatics, Akademijos 4, Vilnius 2600, Lithuania, E-mail: jonas.mockus@mii.lt

**Nast, Michael** University of Trier, Department of Mathematics, D-54286 Trier, Germany, E-mail: nast@uni-trier.de

**Neumaier, Arnold** Institut für Mathematik, Universität Wien, Strudlhofgasse 4, A-1090 Wien, Austria, E-mail: neum@cma.univie.ac.at

**Ortigosa, Pilar M.** Dpto. Arquitectura de Computadores y Electrónica, Universidad de Almería. 04120-Almería. Spain, E-mail: inma@iron.ualm.es

**Panagiotopoulos, Panagiotu D.** Dept. of Civil Engineering, Aristotle University, GR-54006 Thessaloniki, Greece, Also in the Faculty of Math. and Physics, RWTH, D-52056 Aachen, Germany, E-mail: pdpana@heron.civil.auth.gr

**Pardalos, Panos M.** Center for Applied Optimization and ISE Department, University of Florida, Gainesville, FL 32611-6595, USA, E-mail: pardalos@ufl.edu

**Pfening, András** Technical University of Budapest, Department of Telecommunications, Sztoczek 2, 1111 Budapest, Hungary, E-mail: pfening@hit.bme.hu

**Phillips, Andrew T.** United States Naval Academy, 572 Holloway Road, Annapolis, MD 21402, USA, E-Mail: phillips@nadn.navy.mil

**Pintér, János D.** Pinter Consulting Services and Dalhousie University, 129 Glenforest Drive, Halifax, Canada B3M 1J2, E-mail: pinter@tuns.ca

**Polyakova, Olga Yu.** Kharkov State University, Department of Economic Cybernetics, 4 Svobody Sq., 310077, Kharkov, Ukraine. E-Mail: rena@ilt.kharkov.ua

**Price, Chris** New Zealand Defence Scientific Establishment,P. O. Box 32-901, Auckland Naval Base, Auckland, New Zealand, E-mail: chrisp@dse.govt.nz

**Pusztaszeri, Jean-François** ECP Division, European Laboratory for Particle Physics (CERN), CH-1211 Geneva 23, Switzerland, E-mail: jfp@aloha.cern.ch

**Ramakrishnan, K.G.** AT&T Bell Laboratories, Murray Hill, NJ 07974 USA, E-mail: kgr@research.att.com

**Rapcsák, Tamás** Computer and Automation Institute, Hungarian Academy of Sciences, Budapest, Hungary, E-mail: h632rap@ella.hu

**Ratz, Dietmar** Universität Karlsruhe, Institut für Angewandte Mathematik, D-76128 Karlsruhe, Germany, E-mail: ae07@rz.uni-karlsruhe.de

**Schoen, Fabio** Dip. Sistemi e Informatica, via di Santa Marta, 3, 50139 Firenze, Italy, E-mail: schoen@ingfi1.ing.unifi.it

**Sergeyev, Yaroslav D.** Institute for Systems Science & Informatics of National Research Council, c/o DEIS, University of Calabria, 87036 Rende (CS), Italy and Software Department, University of Nizhni Novgorod, Gagarin Av. 23, Nizhni Novgorod, Russia, E-mail: yaro@si.deis.unical.it

**Sotiropoulos, Dimitris Gr.** University of Patras, Department of Mathematics, GR- 261 10 Patras, Greece, E-mail: dgs@math.upatras.gr

**Stephens, Chris** Department of Mathematics and Statistics, University of Canterbury, Private Bag 4800, Christchurch, New Zealand, E-mail: c.stephens@math.canterbury.ac.nz

**Strekalovsky, Alexander S.** Mathematical Department, Irkutsk State University, Gagarin Avenue, 20, 664003, Irkutsk-3, Russia, E-mail: strekal@uni.irkutsk.su

**Strongin, Roman G.** State University of Nizhni Novgorod, 23 Gagarin Prospect, 603600 Nizhni Novgorod, Russia, E-mail: strongin@nnucnit.nnov.su

**Thoai, Nguyen Van** Department of Mathematics, University of Trier, 54286 Trier, Germany, E-mail: thoai@orsun2.uni-trier.de

**Toraldo, Gerardo** Università di Napoli "Federico II", Facolta di Agraria, Via Cintia, Complesso Monte S. Angelo, 80126 – Napoli, Italy, E-mail: toraldo@matna2.dma.unina.it

**Tsevendorj, Ider** Mathematical Department, Irkutsk State University, K. Marx Street 1, Irkutsk, 664 003, Russia, E-mail: strekal@uni.irkutsk.su

**Varga, József** University of Veszprém, Dept. of Applied Informatics, H-8000 Veszprém, Egyetem u. 2, Hungary, E-mail: vj@almos.vein.hu

**Vasiliev, Igor L.** Mathematical Department, Irkutsk State University, Gagarin Avenue, 20, 664003, Irkutsk-3, Russia, E-mail: strekal@uni.irkutsk.su

**Vrahatis, M.N.** Department of Mathematics, University of Patras, GR-261.10, Patras, Greece, E-mail: vrahatis@math.upatras.gr

**Wu, Zhijun** Argonne National Laboratory, Mathematics and Computer Science Division, Argonne, IL 60439, USA, E-mail: zwu@mcs.anl.gov

**Zabinsky, Zelda B.** Industrial Engineering, Box 352650, University of Washington, Seattle, WA 98195-2650, USA, E-mail: zelda@u.washington.edu

**Žilinskas, Antanas** Inst. Math. Informat & Vytautas Magnus Univ., Department of Informatics, 4 Akademijos str., LT-2600 Vilnius, Lithuania, E-mail: zilinskas@info.ped.lt

**Zirilli, Francesco** Dipartimento di Matematica "G. Castelnuovo", Università di Roma "La Sapienza", 00185 Roma, Italy, E-mail: apzrm@itcaspur.caspur.it

**Additional information**

**E-mail** A modest telnet facility will be built for the participants. This will allow to make character (i.e. not graphical or x-terminal) telnet connections to their home machines. In this way, they will be able to read their new e-mails, and to answer them. You can use character oriented mailer programs (such as mail, elm or pine). The connection will be open after the sections (16:30 – 20:00), in this way the workshop will not be disturbed.

**thermal bath** The hotel has its own indoor thermal bath that is available for the guests free of charge (7:00 – 20:00). Its temperature is about 32 centigrade.

**sauna** There is also a sauna available in the hotel, it costs 250 HUF (about 2 USD).

**swimming** There is a 4-pool indoor swimming facility in the neighbourhood (100 meters) of the hotel.

**phone** Each room of the hotel is equipped with a phone. You can use it also for distance calls. The costs of the calls are not covered by the participation fee (neither other services like laundry or mini bar in the rooms)— you must pay these services when you leave the hotel.

**public transportation** The city center is in walking distance (about 1.5 km) to the hotel. The old bridge is being repaired, yet open for pedestrians and for public transportation. To use the buses, trolleys or trams you can buy a ticket before the travel (36 HUF, about 0.3 USD each), and you must punch it on the bus. You can also buy ticket from the driver for a larger price (50 HUF, about 0.4 USD).

**taxi** The best is to ask the at the reception desk for a taxi. A trip within Szeged should not cost more than 600 HUF (5 USD).

**Distribution of the participants:**

| | |
|---|---|
| Australia | 1 |
| Austria | 3 |
| Canada | 1 |
| Denmark | 1 |
| Germany | 8 |
| Greece | 3 |
| Hungary | 10 |
| Italy | 5 |
| Jordan | 1 |
| Lithuania | 2 |
| Macedonia | 1 |
| The Netherlands | 1 |
| New Zeeland | 2 |
| Russia | 6 |
| Spain | 3 |
| Sweden | 1 |
| Switzerland | 1 |
| Ukraine | 1 |
| USA | 9 |
| | 60 |

# Contents

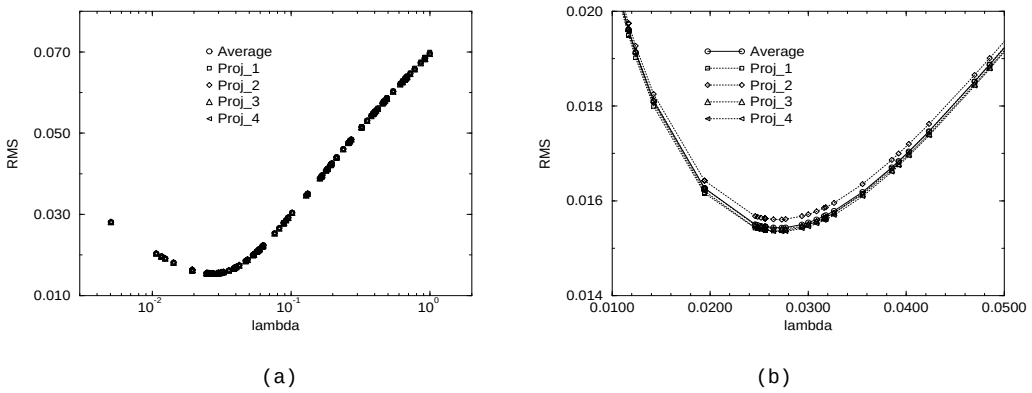



Figure 1: Values for the objective function at the trial points tested during an execution of PCRS using four different set of projection data. (b) is an enlargement window of (a) around the optimum value

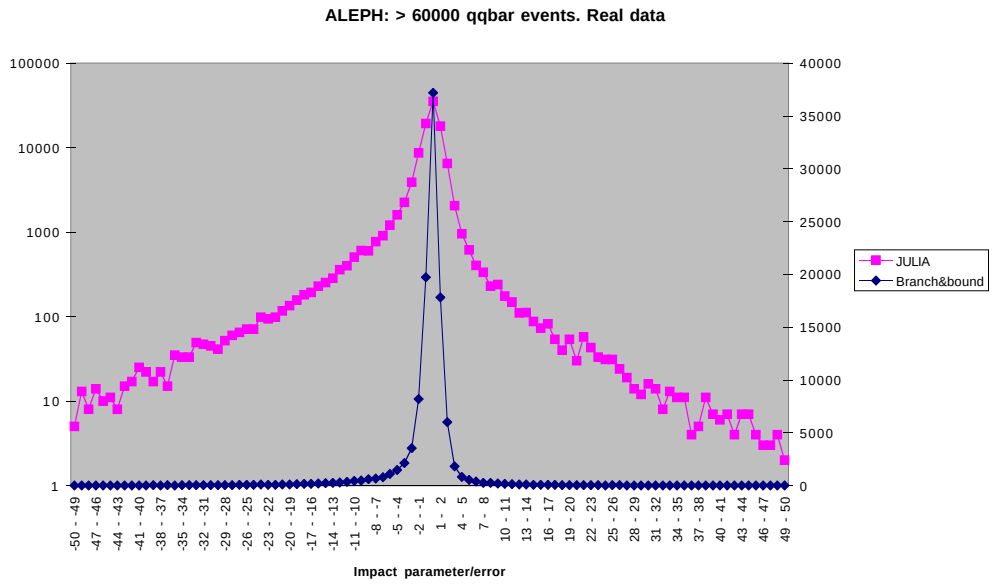



Figure 1: Impact parameter/error distributions for 10'000 hadronic $Z^0$ events (60'000 tracks), obtained with the old sequential code (JULIA) and the new combinatorial method. The new method produces a sharper peak about the impact parameter value (bin zero) and flatter tails in the outlying area