INTERNATIONAL WORKSHOP ON GLOBAL OPTIMIZATION

FIRENZE, 26/09 - 02/10/1999

Global Optimization

G

69...
We would like to welcome you as a participant to GO • 99, the International Workshop on Global Optimization, hosted by the University of Florence (Italy).

Global Optimization, the field including theory, methods and applications of optimization techniques aimed at detecting a global optimum for difficult mathematical programming problems in which many local optima might exist, is a rich area of research. The subject is now reaching a mature state, with many high quality papers published in qualified scientific journals and books, a journal and a series of monographs explicitly dedicated to the field.

The workshop follows successful previous conferences held in Sopron and Szeged (Hungary), Princeton (U.S.A.) and Trier (Germany), where the most qualified representatives of this important field of research were brought together. This workshop has been organized in single stream sessions, in order to give all participants the opportunity to enjoy each of the presentations. Accordingly, only a limited number of contribution has been accepted.

This workshop might not have been possible without the supporting help of many people and organizations. Among these, we would like to express our sincere thanks to the contribution of:

- University of Florence
- Provincia di Firenze
- EPT (Ente Provinciale del Turismo)
- ATAF

and, of course, of the precious work of the Organizing Committee:

Immanuel Bomze University of Wien (Austria)
Reiner Horst University of Trier (Germany)
Marco Locatelli University of Firenze (Italy)
Panos Pardalos University of Florida (U.S.A.)
Fabio Schoen University of Firenze (Italy)
A covering method for d.c. optimization.

Applications

Rafael Blanco
Dept. de Estadística e Investigación Operativa
Universidad de Sevilla
Tarfía S.N.
41012 Sevilla (SPAIN)
Email: rblanque@cica.es

Emilio Carrizosa
Dept. de Estadística e Investigación Operativa
Universidad de Sevilla
Tarfía S.N.
41012 Sevilla (SPAIN)
Email: ecarriz@cica.es

In this talk we consider a covering algorithm for optimizing a d.c. function over a polytope. This method, based on the classical Piyavskii’s covering algorithm, makes use of a d.c. decomposition of the objective to provide an easily updated convergent covering sequence, even in the multidimensional case, without any assumption about the differentiability of the objective.

Breiman-Cutler algorithm and its generalization by Baritompa-Cutler appear as particular cases of this method for given d.c. decompositions, and all of them are outer approximation methods.

The choice of the d.c. decomposition may have a great influence on the convergence of the algorithm. In fact, we show that an optimal representation (in a given sense) can be computed for the one-dimensional case. The computational results obtained by using this decomposition confirm the importance of a good choice. In particular, by applying these results to the problem of estimating the location parameter of the statistical Cauchy Distribution, we obtain a resolution algorithm improving previously published methods.

We also provide other applications of the d.c. optimization to the fields of Location Theory, Multiple Criteria Decision Making and NonLinear Regression. All of them are based on the fact that a d.c. representation can be computed for the composition of a norm with a vector valued d.c. function.
References


On nonconvexity and Optimal Design of Experiments

Eric P.J. Boer
Subdepartment of Mathematics
Wageningen University
Dreijenlaan, 4
6703 HA Wageningen
Email: eric.boer@wts.wk.wau.nl
Eligius M.T. Hendrix
Email: eligius.hendrix@oa.wk.wau.nl

Optimal Design of Experiments is a topic in statistics concerning the question "Where to measure, how many times?". In this paper it is shown, how easy to formulate design questions may lead to complicated nonconvex optimization problems. Illustrative examples are given from linear, polynomial and nonlinear regression. Furthermore it is shown, how a traditional algorithm in this field proceeds.

1 Introduction

The eagerness of mankind to understand environmental processes and phenomena in (bio-)chemistry, geosciences, biology, electrical/mechanical engineering, economics etc., gives rise to the use of mathematical models. Analysis of models leads to a better understanding of the observable world of a researcher. Model validation is - to put it in a simple and non-mathematical way - a matter of comparing the calculated, theoretical model results with measured values. There are two important optimization problems in model validation (Hendrix, 1998).

1. Model calibration or parameter estimation involves the search for the "best" parameter values in a descriptive model with respect to measurement values.

2. Optimal experimental design concerns the question which measurements should be performed to increase the reliability of the estimation of the parameters given a model and a set of parameters.

In this paper focus is on the experimental design problem. It is shown why the optimization problem of experimental design is interesting from a global
optimization point of view. We will give a flavour of the problem without going into detail. Some terminology is introduced first.

2 Terminology and examples

First of all we should introduce the concept of an experimental design. One concept is that of a so-called discrete design denoted as

\[
\begin{pmatrix}
  x_1 & x_2 & \ldots & x_r \\
  p_1 & p_2 & \ldots & p_r 
\end{pmatrix}
\]

(1)

where \( x_i \) are points where to measure and \( p_i \) denotes a measurement weight, \( \sum p_i = 1 \); how much to measure at point \( x_i \).

From an optimization point of view, we would like to find the best values for \( p_i \) and \( x_i \). Notice however, that in some situations it also may not be known how many measurement points \( r \) should be taken. A more practical definition is that of an exact design \( (N) \):

\[
\begin{pmatrix}
  x_1 & x_2 & \ldots & x_r \\
  n_1 & n_2 & \ldots & n_r 
\end{pmatrix}
\]

(2)

in which \( n_i \), \( \sum n_i = N \) denotes the number of measurements at point \( x_i \) and \( N \) is the available number of measurements. At first instance one would consider this a mixed continuous-integer optimization problem, as the values of \( n_i \) are integer. However the design can also represented as

\[
\begin{pmatrix}
  x_1 & x_2 & \ldots & x_r \\
  1 & 1 & \ldots & 1 
\end{pmatrix}
\]

(3)

which may imply an increase in the number of decisions in the design problem. Moreover one should take into account that \( x_1 \leq \ldots \leq x_i \leq \ldots \leq x_N \).

The optimality of a design depends on the model \( \eta(x, \theta) \) with \( \theta = (\theta_1, \ldots, \theta_m) \), \( x \in X \subseteq \mathbb{R}^k \), under consideration. In many cases research focuses on models linear in the parameters \( \eta(x, \theta) = \theta^T f(x) \). Moreover performance depends on the specific criterion which tells us how to deal with the variance. Without going into detail, often the determinant of the so-called information matrix is used:

\[
M[r(N)] = \sum_{i=1}^r p_i f(x_i) f'(x_i)
\]

(4)

Another important concept which makes the design problem more easy to understand from a graphical point of view is that of the dispersion function.

\[
d(x, r(N)) = f'(x) M^{-1} f(x)
\]

(5)

of which examples are given in Figure 1. It represents -to put it in a non-mathematical way- the variance of the resulting prediction. In some cases
(for discrete design), finding the maximum of \( |M(e)| \) is equivalent to finding \( \min \max d(x, e) \). From such observations one can derive easily that finding the optimal design in linear regression (Figure 1) relates to concave optimization (usually the experimental region \( X \) is convex).

![Figure 1: Dispersion functions for linear and quadratic regression (Atkinson and Donev, 1992).](image)

Figure 1 shows examples of \( d(x, e) \). One can observe that \( \min \max d \) leads, for the linear case, to an extreme point design. For polynomial regression the objective function is certainly not necessarily concave (Figure 1, quadratic regression). Specific algorithms have been derived for optimal design problems. One of the algorithms for models linear in parameters proceeds as follows (Figure 1): put more weight at that point where \( d(x, e) \) reaches a maximum, until all maxima are equal (based on analytical mathematical results, Fedorov; 1972). Intuitively this is clear, but also implies problems such as finding the maximum (local maxima) possibly in a multidimensional (\( k > 1 \)) situation.

### 3 Challenges

Several other challenging problems exist in this field.

- Figure 2 illustrates a case of nonlinearity in the parameters for the Michaelis-Menten function (Boer et al, 1999). It shows the nonconvexity of the criterion function.

- As shown by Rasch (1997) the optimal design problem becomes a combinatorial optimization problem when we do not allow repetitions (replications) of measurements at one point and we have to select the measurements from a predefined set of candidate points.

- Interesting examples of multipurpose designs can be found in spatial statistics (Müller, 1998). In this field we like to model the spatial variability (covariance
function) which leads to clustered measurements and simultaneously (given the covariance function) we look for a space filling design.

References


On copositive programming and standard quadratic optimization problems

Immanuel M. Bomze
University of Vienna
Universitätsstraße 5
A-1010 Wien, Austria
Email: immanuel.bomze@univie.ac.at

Mirjam Díir
Vienna University of Economics, Austria

Etienne de Klerk, Cornelius Roos, Arie Quist, Tamas Terlaky
Delft University of Technology, The Netherlands

A standard quadratic problem (QP) consists of finding (global) maximizers of a quadratic form over the standard simplex \( \Delta \):

\[
\max \{ x^T A x : x \in \Delta \},
\]

with \( A \) a symmetric \( n \times n \) matrix and \( \Delta = \{ x \in \mathbb{R}^n : x \geq 0 \text{ and } e^T x = 1 \} \). Here, \( e = [1, \ldots, 1]^T \in \mathbb{R}^n \) and \( ^T \) signifies transposition. Standard QPs arise quite naturally in procedures which enable an escape from local solutions to general quadratic problems, and moreover have diverse direct real-life applications like the maximum (weight) clique problem. In this paper, the usual semidefinite programming approach is enlarged to reformulate a standard QP into a linear programming problem on a cone which is the dual of the cone of copositive matrices (i.e., those which yield a non-negative quadratic form on the positive orthant).

To be more specific, consider the general form of an LP over a pointed convex cone \( \mathcal{K} \): the primal problem

\[
\max \{ c^T x : D x \leq b, x \in \mathcal{K} \},
\]

where \( D \) is \( m \times d \), \( b \) is \( m \times 1 \), \( c \) is \( d \times 1 \), and its dual

\[
\min \{ b^T y : y \geq 0, D^T y - c \in \mathcal{K}^* \}.
\]

Here \( \mathcal{K}^* = \{ s \in \mathbb{R}^d : s^T x \geq 0, \text{ all } x \in \mathcal{K} \} \) is the dual cone of \( \mathcal{K} \).

Now in semidefinite programming (SDP) we usually have \( d = \binom{n+1}{2} \); further,
the inner product is given by the duality acting on symmetric $n \times n$ matrices: if $A$ and $X$ are two such matrices, the expression $c^T x$ from above will be replaced with $\langle A, X \rangle = \text{trace}(AX)$. The cone used in SDP is that of all symmetric positive semidefinite $n \times n$ matrices,

$$K = P = \{X \text{ symmetric } n \times n : X \text{ is psd}\}$$

which is self-dual so that $P^* = P$.

Cosepositive programming (COP) uses the same inner product, but employs a different cone: rather than $P$ in SDP, here

$$K = \text{conv} \{xx^T : x \in \mathbb{R}^n, x \succeq o\}$$

is the cone of completely positive matrices, with its dual cone

$$K^* = \{S \text{ symmetric } n \times n : S \text{ is cosepositive}\}.$$

Recall that a symmetric $n \times n$ matrix $S$ is said to be cosepositive if

$$v^T Sv \geq 0 \quad \text{if} \quad v \succeq o.$$

While it is relatively easy to check whether or not a matrix is positive semidefinite, i.e., whether $X \in P$ holds, remember that to check $X \in K$ or $S \in K^*$ is NP-hard. So one difficulty is to obtain efficient shortcuts for checking primal and dual feasibility, some of which will be presented in the talk.

A central observation relating standard QPs to COP is that the extremal points of the feasible polyhedron are all of rank one: if $E = ee^T$ is the $n \times n$ matrix of all ones, then

$$\text{ext } \{X \in K : \langle E, X \rangle = 1\} = \{xx^T : x \in \Delta\}.$$

Hence every standard QP is equivalent to the COP

$$\max \{\langle A, X \rangle : X \in K, \langle E, X \rangle = 1\},$$

and its dual

$$\min \{y \in \mathbb{R} : yE - A \in K^*\}$$

is a counterpart to the eigenvalue bound problem in the SDP context $\lambda_{\text{max}}(A) = \min \{y \in \mathbb{R} : yI - A \in P\}$.

Apart from this interpretation, this formulation allows to employ several interior-point primal-dual algorithms, which are by now among the most popular methods in semidefinite programming. Furthermore, these approaches are combined with an evolutionary dynamics approach which generates primal-feasible paths along which the objective is monotonically improved until a local solution is reached.
Concave programming and numerical methods for finding all roots of nonlinear systems

V.P. Bulatov
Russian Academy of Sciences
Irkutsk, 664033 (Russia)

1. Formulation of the problem. Let \( g_1(x), \ldots, g_n(x) \) be convex continuously differentiable functions, \( x \in \mathbb{R}^n \), \( \mathbb{R}^n \) be a convex closed bounded set. The problem is to determine all points \( \hat{x} \in \mathbb{R}^n \) such that

\[
\mathbf{g(\hat{x})} = 0, g \in \mathbb{R}^n. \tag{1}
\]

The roots \( \hat{x} \in \mathbb{R}^n \) of system (1) will be called different if the matrix \( \nabla g(\hat{x}) \) has rank \( n \) and of multiplicity \( k : 1 \leq k < n \), if the rank \( \nabla g(\hat{x}) \) is \( n - k \).

At first we consider different roots. Here we will assume that system (1) has a finite number of roots, i.e., all roots of system (1) are isolated.

2. Some auxiliary reductions.

First of all, the idea of the solution method consists in the regularization of the problem, i.e., in constructing such an auxiliary problem which is equivalent in some sense to the initial one and moreover, its feasible set contains the set \( R = \{ x : x \in \mathbb{R}^n, g_i(x) = 0, i = 1, \ldots, n \} \).

Then an iterative process analogous to the one described in [1] is constructed: find a corresponding to a root of system (1) solution \( x^1 \) of the regularized problem; determine a cutting halfspace which does not contain \( x^1 \), but contains all other real roots; solve again the regularized problem and so on.

Before starting the detailed description of the method let us consider the following auxiliary problem:

\[
\min \{ \varphi(x) : x \in R \}, \tag{2}
\]

where \( R = \{ x \in \mathbb{R}^n : g_i(x) = 0, x \in L, i = 1, m, m \leq n \} \),

\[
L = \{ x : f_j(x) \leq 0, j = 1, \ldots, l \}, \tag{3}
\]

\( \varphi(x), g_i(x), f_j(x) \) are smooth functions.

Corresponding to problem (2)-(3) is the following one:

\[
\min \{ \psi(x) = \varphi(x) + N \sum_{i=1}^{m} g_i(x) : x \in \mathbb{R}^n \}, \tag{4}
\]
where

\[ R^0 = \{ x : g_i(x) \geq 0, f_j(x) \leq 0, i = 1, m, j = 1, I, N > 0 \}. \]  

(5)

It is known [1,5] that under some conditions there exists \( N^* \) such that \( \forall N > N^* \) any minimal point of \( \psi(x) \) over \( R^0 \) satisfies the equalities \( g(x) = 0 \).

Moreover, any solutions \( x^1, x^2 \) of the system \( g(x) = 0 \) such that \( \varphi(x^1) > \varphi(x^2) \) are local minimum points of \( \varphi(x) \) over \( R^0 \), moreover \( \psi(x^1) > \psi(x^2) \). Conversely, minimum points \( x^1 \) and \( x^2 \) of \( \psi(x) \) over \( R^0 \) such that \( \psi(x^1) > \psi(x^2) \) are solutions of the system \( g(x) = 0 \) and \( \varphi(x^1) > \varphi(x^2) \).

Assume now that \( \varphi(x), g_i(x), f_j(x) \) are convex smooth functions. Then problem (4)-(5) of minimization of convex function under convex and concave constraints has local minima for which \( g_i(x) = 0 \) is fulfilled for sufficiently large \( N \).

This problem was studied by various authors, see [1]-[5]. For instance, in [1] the following iterative process convergent under some conditions to a stationary point is suggested:

\[ x^{k+1} = \text{argmin}\{ \varphi(x) + N \sum_{i=1}^m g_i(x) : x \in R^k \}, \]  

(6)

where

\[ R^k = \{ x : g_i(x^k) + \nabla g_i(x^k)^T (x - x^k) \geq 0, i = 1, m, f_j(x) \leq 0, j = 1, I \}. \]  

(7)

Hence, in order to find a local minimum in problem (4)-(5) one has to solve a sequence of convex problems of type (6)-(7) with the same objective function \( \psi(x) \).

If \( f_j(x) \) are linear functions, then the local search is equivalent to minimization of a convex function on a sequence of polytopes \( R^k \) which are not very different from each other.

Consider now the problem

\[ \max\{ \varphi(x) : x \in R \}, \]  

(8)

where \( R \) is determined by (3).

Then, similar to (4)-(5) one can relate to problem (3),(8) the following construction

\[ \max\{ \varphi(x) + N \sum_{i=1}^m g_i(x) : x \in R^0 \}, \]  

(9)

where

\[ R^0 = \{ x : f_j(x) \leq 0, g_i(x) \leq 0, i = 1, m, j = 1, I, M > 0 \}. \]  

(10)

If as in (6)-(7) functions \( \varphi(x), g_i(x), f_j(x) \) are convex and smooth, then (9)-(10) is a problem of maximization of a convex function over a convex set. Moreover, for sufficiently large \( N \) \( g_i(x) = 0(i = 1, m) \) for all local maximum points. For local search in (9)-(10) one can use the same approach as in [1]-[5]:

\[ x^{k+1} = \text{argmax}\{ \nabla \varphi(x^k)^T x + N \sum_{i=1}^n \nabla g_i(x^k)^T x : x \in R^0 \}, \]  

(11)
where \( R^0 \) is determined by (10), i.e. the local search is equivalent to minimization of linear forms over the convex set \( R^0 \).


Let us pass now to the description of numerical methods for solving problem (1). Assume, that in (2) and (3) \( m = n \). Let us order solutions of problem (1) with respect to their norms. Connect (as above) problem (1) with the following one

\[
\min \{ \psi(x) = \Vert x \Vert^2 + N \sum_{i=1}^{n} g_i(x) : x \in R^0 \},
\]

where

\[
R^0 = \{ x : x \in L_i g_i(x) \geq 0, i = 1, n \}.
\]  

(12)

Using, for example, procedure (6)-(7) find a local minimum point \( x^0 \) of \( \psi(x) \) over \( R^0 \). When \( N \) is large enough we have \( g_i(x^0) \equiv 0 \). Construct at the point \( x^0 \) the cone

\[
L^0 = \{ x : \nabla g(x^0)^T x \leq \nabla g(x^0)^T x^0 \}.
\]

It is clear, that \( L^0 \) contains the solution set of system (1). By the assumption the matrix \( \nabla g(x^0) \) is nondegenerate. The equations of the edges of the cone \( L^0 \) has the form

\[
x^j = x^0 - \lambda^j s^{ij}, j = 1, n
\]

(13)

where \( s^{ij} \) are columns of the matrix inverse to \( \nabla g(x^0) \)

From the equations

\[
\sum_i g_i(x^0 - \lambda^j s^{ij}) = \sum_i g_i(x^0)
\]

(14)

find \( \lambda^{ij} > 0 \) and points \( x^{0j} = x^0 - \lambda^{ij} g^{ij} \) of intersections of rays (13) with the boundary of the convex set \( : \sum_i g_i(x) \leq 0 \). If equation (14) do not have positive solutions, then the corresponding \( \lambda^{ij} > 0 \) can be taken arbitrary.

Due to the linear independence of \( s^{ij} \) the points \( \{x^{01}, ..., x^{0m}\} \) uniquely determine a plane \( \beta^T x = t_0 \) and the half-space \( \beta^T x \leq t_0 \) which does not contain \( x^0 \).

Then, find \( x^k \) from solving the following problem

\[
\min \{ \psi(x) = \Vert x \Vert^2 + N \sum_{i=1}^{n} g_i(x) : x \in R^1 \},
\]

where \( R^1 = \{ x : x \in R^0, \beta^T x \leq t_0 \} \) and so on.

Let a sequence \( \{ x^k \} \) such that \( g_i(x^k) = 0, \forall j = 1, k \) be constructed. Determine at \( x^k \) the cone

\[
L^k = \{ x : \nabla g(x^k)^T x \leq \nabla g(x^k)^T x^k \}.
\]

Then, the equations of the edges of the cone \( L^k \) have the form

\[
x^j = x^k - \lambda^j s^{kj},
\]

(15)
where \( s^{kj} \) are columns of the matrix inverse to \( \nabla g(x^k) \) (by the assumption \( |\nabla g(x^k)| \neq 0 \)).

From equations \( \sum_j g_j(x^k - \lambda^k s^{kj}) = 0 \) find \( \lambda^{kj} > 0 \) and the corresponding points \( x^{kj} = x^k - \lambda^{kj} s^{kj} \) of intersection of rays (15) with the surface \( \sum_{k=1}^n g_i(x) = 0 \). Determine a half-space \( j^T x \leq t_k \), which cuts off \( x^k \) and find the next solution from

\[
\min \{ \psi(x) : x \in R^{k+1} \},
\]

where

\[
R^{k+1} \{ x : x \in R^k, j^T x \leq t_k \}.
\]

The following Lemma follows from geometrical considerations.

**Lemma.** The intersection of the cone \( L^k \) with the half-space \( j^T x > t_k \) does not contain solutions of the system of nonlinear equation (1) different from \( x^k \).

Let us consider now numerical methods for solving problem (1), based on the reduction of type (9),(10) and numerical methods of type (11) for local search in problem (9)-(10).


Assume, that solutions \( \{ x^1, ..., x^{k-1} \} \) of the system \( g_i(x) = 0 \) are found. The next solution is searched as the one with the maximum distance from the those previously found. For this purpose define function \( \varphi(x) = \min_{1 \leq j \leq k} || x - x^j || \) and correspond to problem (1) the following one

\[
\max \{ \min_{1 \leq j \leq k} || x - x^j ||^2 + \sum_{i=1}^N g_i(x) : x \in R \}
\]

or

\[
\min \{ x_{n+1} + \sum_{i=1}^n g_i(x) \},
\]

subject to

\[
g_i(x) \leq 0, \quad i = 1, n, \quad x \in L_1, \quad || x - x^j ||^2 \geq x_{n+1}, \quad j = 1, k.
\]

Rewrite the latter problem as follows

\[
\max \{ x_{n+1} + \sum_{i=1}^n g_i(x) \}
\]

subject to \( g_i(x) \leq 0, \forall i = 1, n, || x ||^2 + || x^j ||^2 - 2x^T x^j \geq x_{n+1}, \forall j = 1, k \).

Making the substitution \( x_{n+1} = z + || x ||^2 \) we obtain finally the following problem

\[
\max \{ z + || x ||^2 + \sum_{i=1}^n g_i(x) \}
\]

subject to

\[
2x^T x^j + z \leq || x^j ||^2 \quad \forall j = 1, k
\]
\[ x \in L, \quad g_i(x) \leq 0 \quad \forall i = 1 \ldots k \]  

(18)

Problem (16)-(18) is a problem of maximization of a convex function on a set, determined by linear constraints (17) and constraints (18). The number of the constraints is increased by one after each iteration.

References


On Convergence Rate of Global Optimization Algorithms Based on Statistical Models

Jim Calvin
Dept. of Computer and Information Sciences
New Jersey Institute of Technology
University Heights
Newark, NJ, USA
Email: calvin@cis.njit.edu

Antanas Zilinskas
Institute of Mathematics and Informatics
Akademijos str. 4
Vilnius, LT2600, Lithuania
Email: antanasz@ktl.mii.lt

Statistical models are useful for modeling the global behavior of a complicated multimodal function and the normal uncertainty of the researcher with respect to the features of the function. However, the choice of a particular model may not be obvious, and may be guided by conflicting goals. For example, the Wiener process seems well-grounded as a global model of complicated one-dimensional multimodal functions. The Wiener model has been the most common model used for the implementation of global optimization algorithms, although the local features of non-differentiable Wiener process paths differ fundamentally from the local features of smooth objective functions. Smooth function statistical models have been of interest for some time, but implementation of algorithms based on such models has been hampered by the computational complexity of the auxiliary calculations. The authors recently extended the approach to statistical models of smooth objective functions using an approximation method enabling the construction of an algorithm with computational complexity comparable to that of the algorithm based on the Wiener model. The availability of two competing models suggests their comparison.

In this talk the convergence rate of algorithms based on Wiener and smooth function models is compared for neutral conditions, i.e., for general assumptions on smoothness of an objective function without respect to a statistical model. Under both algorithms, the order of convergence of the error to zero is the same up to a constant. However, the constant depends on features of the objective function and a choice of method parameter ε. This result gives new insight into
the role of the underlying model in determining the asymptotic properties of the algorithm. Until recently the only way to assess the convergence rate and to compare efficiency of statistical model-based methods was implementation and experimental testing. The new techniques obviate the need for experimental investigation.

The specific problem we consider is that of minimizing the continuous objective function \( f(x) \) over the interval \([0, 1]\). We are mainly interested in the case where \( f \) does not satisfy stronger regularity conditions, such as convexity or unimodality. Let the minimal value \( M = \min_{0 \leq x \leq 1} f(x) \) be attained at the point \( x^* \), and assume that \( f(x) > M \) for \( x \neq x^* \). The stochastic process \( \{\xi(x) : 0 \leq x \leq 1\} \) is accepted as a statistical model of the objective function. Fix \( \varepsilon > 0 \). The \( n \)-th observation of the function value is performed by the \( P \)-algorithm at the point

\[
x_{n+1} = \arg \max_{0 \leq x \leq 1} P[\xi(x) < M_n - \varepsilon | \xi(x_i) = y_i, i = 1, \ldots, n],
\]

where \( x_i, y_i = f(x_i) \) are the results of observations at previous minimization steps and \( M_n = \min_{i \leq x \leq n} \xi(x) \).

The magnitude of the error depends in the limit on two quantities associated with the objective function \( f \). We can think of these as a global characteristic \( \Gamma_\beta(\varepsilon) \), depending on the parameters \( \alpha, \beta \), and a local characteristic \( \Lambda \). For \( \varepsilon > 0 \) and \( \beta > 0 \), define the global characteristic by

\[
\Gamma_\beta(\varepsilon) = \int_{x=0}^{\varepsilon} \left( 1 + \frac{f(x) - M}{\varepsilon} \right)^{-\beta} dx.
\]

Note that \( \Gamma_\beta(\varepsilon) \) is decreasing in \( \beta \) and increasing in \( \varepsilon \). The larger values of \( 0 < \Gamma_\beta(\varepsilon) \leq 1 \) correspond to more difficult minimization problems due to the concentration of function values near the global minimum. A constant objective function has maximal index of difficulty \( \Gamma_\beta(\varepsilon) = 1 \).

For the local characteristic, we assume that there exists a positive number \( \alpha \) such that

\[
\lim_{n \to \infty} n^{\alpha} \sup_{|x| \leq 1/n} f(x^* + x) - M = \Lambda(f) > 0.
\]

For example, if \( f(x) = a|x - x^*| \), then \( \alpha = 1 \) and \( \Lambda(f) = a \). If \( f \) is smooth with positive second derivative at \( x^* \), then \( \alpha = 2 \) and \( \Lambda(f) = f''(x^*)/2 \).

Fix \( \varepsilon > 0 \). Let an objective function \( f(x) \) be continuous over the minimization interval and have local characteristic \( \Lambda(f) \) with exponent \( \alpha \) at the unique global minimizer \( x^* \), and global characteristic \( \Gamma_\beta(\varepsilon) \). Then the \( P \)-algorithm, based either on the Wiener or on a smooth function model, converges to the global minimum in the sense that the error \( \Delta_n \downarrow 0 \) and in particular

\[
\lim_{n \to \infty} \sup n^{\beta} \Delta_n \leq 4 \Lambda(f) \Gamma_\beta^2(\varepsilon)
\]

where \( \beta = 1/2 \) for the algorithm based on a smooth process model, and \( \beta = 2 \) for the algorithm based on the Wiener process model. We consider extension of both algorithms to the case where the constant threshold \( \varepsilon \) is replaced by a sequence \( \varepsilon_n \downarrow 0 \).
Parallel Search with Bounded Information

Jim Calvin  
Dept. of Computer and Information Sciences  
New Jersey Institute of Technology  
University Heights  
Newark, NJ, USA  
Email: calvin@cis.njit.edu

Let $f$ be a real-valued function defined on a compact set $A \subset \mathbb{R}^d$. We are interested in approximating the global minimum of $f$ over $A$ based on observation of the function value at sequentially selected points while maintaining a memory of a fixed finite number of function values. In this talk we describe a class of randomized algorithms with the following property: For a large class of objective functions, and for any integer $k$, there exists an algorithm in the class for which the probability that the error after $n$ observations exceeds $n^{-k}$ converges to 0.

For $T > 0$ let $B_T = \{x \in \mathbb{R}^d : |x| \leq T\}$ be the closed ball of radius $T$ in $\mathbb{R}^d$. By rescaling and extending the function if necessary we can take the domain $A = B_1$. We assume that $f$ attains its global minimum $f^* = \min_{t \in A} f(t)$ at a unique point $t^*$ in the interior of $A$. The searcher chooses points $t_1, t_2, \ldots \in A$ and forms an approximation $(t_1, f_1), (t_2, f_2), \ldots$ based on $(t_i, f(t_i)) : i = 1, 2, \ldots, n$. A general adaptive algorithm in our setting will choose the $(n+1)$st point $t_{n+1}$ as a function of all the previous observations and some auxiliary randomization; i.e.,

$$t_{n+1} = h_{n+1}(t_1, f(t_1), t_2, f(t_2), \ldots, t_n, f(t_n), Z_n)$$

for some function $h_{n+1}$ and random variable $Z_n$.

In this talk we are concerned with the case of bounded memory. If a total of $M$ observation pairs are allowed, then to keep a new observation an old one must be discarded (after $M$ are stored), so our algorithm takes the form

$$t_{n+1} = h_{n+1}(t_{i_1}, f(t_{i_1}), t_{i_2}, f(t_{i_2}), \ldots, t_{i_M}, f(t_{i_M}), Z_n).$$

The basic questions include which information should be kept and which discarded, and how should the information be used to select new observation points. Let

$$\Delta_n = \min_{1 \leq i \leq n} f(t_i) - f(t^*)$$

denote the error after $n$ observations. We are mainly interested in the convergence rate of $\Delta_n$ to 0 under various algorithms; that is, we undertake an
asymptotic analysis. The obtainable convergence rates depend on the characteristics of the objective function $f$, as well as on the cardinality of information $M$. In this talk, we consider a broad class of functions $f$, satisfying only a minimal regularity condition.

The basic assumption we make on the objective function is most easily described in the context of independent, uniform sampling. Let $\{U_i : i \geq 1\}$ be a sequence of independent random variables, all uniformly distributed on $B_1$. Let $U_n^*$ be that observation point of the first $n$ with the smallest function value: i.e., $U_n^* = U_j$ for some $1 \leq j \leq n$ and $f(U_n^*) \leq f(U_i), i \leq n$, with ties broken arbitrarily.

Assume that $f$ is Borel measurable and for $T > 0$, let
\[
g_n(T) = \min\{f(U_i) : i \leq n, \|U_i - t^*\| \leq n^{-1/d}T\} - \inf\{f(t) : \|t - t^*\| > n^{-1/d}T\},\]
where $\min \emptyset = +\infty$. Our basic assumption is that
\[
\lim_{T \to \infty} \lim_{n \to \infty} \sup P\{g_n(T) \geq 0\} = 0.
\]

Basically, the assumption can be described as saying that if points are chosen according to a Poisson point process on a ball of radius $T$ centered at $t^*$, then the minimum inside the ball will likely be smaller than the minimum outside the ball if $T$ is large. The assumption holds, for example, if $f$ is monotonically increasing with distance from $t^*$ on a neighborhood of $t^*$, though this condition is not necessary.

We construct a class of optimization algorithms, for varying cardinality of information $M$, that converge at a rate that increases with $M$. Since the algorithms do not make use of local properties such as gradients or even continuity, they provide a reasonable lower bound on the convergence rate achievable for optimization problems in $\mathbb{R}^d$. The algorithms can be implemented on $M$ processors in parallel, with some inter-processor communication.

The main result is that for functions satisfying our basic assumption it is possible to obtain arbitrary polynomial speedup compared with Monte Carlo search. More precisely, for $T > 0$ let $N_n^{\alpha,T}$ denote the point process on $B_T$ induced by the points $n^{\alpha/d}(t_k - t^*)$ that lie within $B_T$ for $k \leq n$; that is, for Borel sets $A \subset B_T$,
\[
N_n^{\alpha,T}(A) = \sum_{k=1}^n I_{\{n^{\alpha/d}(t_k - t^*) \in A\}}.
\]

Under simple Monte Carlo search (choosing points independently and uniformly over $B_1$), $N_n^{1,T}$ converges in distribution to a Poisson process. For any $\alpha > 1$, we can choose a cardinality of information $M(\alpha)$ and construct an algorithm such that $N_n^{\alpha,T}$ converges to the same Poisson process. In other words, it is effectively as if $n^\alpha$ observations were made instead of $n$.

We explore the implications of this result with several concrete examples of objective functions.
Convex Quadratic Programming Approach to the Maximum Matching Problem

Domingos M. Cardoso
Universidade de Aveiro, Departamento de Matemática
Campus Universitário de Santiago
3810 Aveiro (Portugal)
Email: dccardoso@mat.ua.pt

In this presentation we deal with undirected simple graphs, $G$, for which $V(G)$ denotes the set of vertices and $E(G)$ the set of edges. An element of $E(G)$, whose ends are the vertices $i$ and $j$, is denoted by $\{i, j\}$. It is also assumed that $G$ is of order $n \geq 1$ (i.e., $|V(G)| = n \geq 1$). Given the graph $G$ (where, according to our terminology, there is nor loops neither multiple edges), a matching in $G$, is a subset of edges, $M \subseteq E(G)$, no two of which have a common vertex. A matching with maximum cardinality is designated maximum matching. On the other hand if for each vertex $v \in V(G)$ there is one edge of the matching $M$ incident with $v$, then $M$ is called a perfect matching.

The problem of determining a maximum matching or whether there exists a perfect matching, is very common in a large variety of applications and as been extensively studied in graph theory. There are several very readable texts about matching theory, among which we can refer, for instance, the classical monograph ofLovász and Plummer [3] or the survey of Pulleyblank [6].

The matrix $A_G$ denotes the adjacency matrix of the graph $G$, that is, $A_G = (a_{ij})_{n \times n}$ is such that

$$a_{ij} = \begin{cases} 1 & \text{if } \{i, j\} \in E(G) \\ 0 & \text{otherwise,} \end{cases}$$

$\lambda_{\min}(A_G)$ the minimum eigenvalue of $A_G$ and $v(G)$ the optimal value of the convex quadratic programming problem

$$(P_G) \max \{2e^T x - x^T (B_G + I_n) x : x \geq 0\},$$

where $e$ is a all-ones vector of $\mathbb{R}^n$,

$$B_G = \begin{cases} \frac{1}{\lambda_{\min}(A_G)} A_G & \text{if } \lambda_{\min}(A_G) \neq 0 \\ 0 & \text{otherwise,} \end{cases}$$

and $I_n$ is the unit matrix of order $n$.

As it is well known, see [1], if $G$ has at least one edge, then $\lambda_{\min}(A_G) \leq -1$ and,
on the other hand, \( \lambda_{\min}(A_G) = -1 \) if and only if every connected component of \( G \) is a complete subgraph.

A subset of vertices \( S \subseteq V(G) \) is called a **maximum stable set** if there is no other stable set with greater number of vertices. The number of vertices in a maximum stable set of a graph \( G \) is called the **stability number** (or **independence number**) of \( G \) and is denoted (as usually) by \( \alpha(G) \).

As it is proved in [5], \( \nu(G) \) is the best upper bound for \( \alpha(G) \), among the optimal values of the family of quadratic programming problems

\[
\phi_{a,b}^* = \max \{ \phi_{a,b}(x) = ax^T x - bx^T (B_G + I_n)x : x \geq 0 \},
\]

with \( a \) and \( b \) such that \( b \geq 0 \) and \( a - b \geq 1 \). A graph \( G \) such that \( \alpha(G) = \nu(G) \) is called a graph with **quadratic stability number** and this class of graphs is denoted by \( Q \).

Given a vertex \( i \in V(G) \), \( N_G(i) \) is the neighbourhood of the vertex \( i \), that is \( N_G(i) = \{ j \in V(G) : \{i,j\} \in E(G) \} \). According to [4], \( G \in Q \) (i.e., \( \alpha(G) = \nu(G) \)) if and only if for a maximum stable set \( S \) of \( G \),

\[
-\lambda_{\min}(A_G) \leq \min \{|N_G(i) \cap S| : i \notin S\}.
\]

A **line graph** of a graph \( G \) is denoted by \( L(G) \) and it is constructed by taking the edges of \( G \) as vertices of \( L(G) \), and joining two vertices in \( L(G) \) whenever the corresponding edges in \( G \) have a common vertex. The determination of a maximum stable set of a line graph \( L(G) \) is equivalent to the determination of the maximum matching of \( G \). Therefore, since \( \alpha(L(G)) \leq \nu(L(G)) \), the optimal solution of \( (P_{L(G)}) \) is an upper bound on the number of elements of a maximum matching of \( G \).

Polynomial-time algorithms are known for the determination of a maximum matching of a graph \( G \), based on the Edmonds perfect matching algorithm introduced in his landmark paper [2]. As a consequence, we can conclude that the stability number of line graphs can be determined in polynomial-time. However, such algorithms are computational expensive, namely, for the non-bipartite case. In this presentation we introduce the following result.

**Theorem 1** A graph \( G \), without isolated vertices and such that \( L(G) \) is not complete, has a perfect matching if \( L(G) \in Q \).

Based on the above result, we develop a procedure for the determination of a perfect matching for an arbitrary graph, if such matching exists, by solving a sequence of quadratic programming problems of the type \( (P_{L(G)}) \), where \( U \subseteq V(G) \).

It must be noted that the triangles and stars are the only graphs \( G \) for which \( L(G) \) is a complete graph. For these graphs, however, it is very easy to find the maximum matching. Furthermore, if \( L(G) \in Q \) then \( \alpha(L(G)) = \nu(L(G)) \) and \( \nu(L(G)) = \frac{n}{2} \).

Despite the knowledge of several necessary conditions for the existence of Hamilton cycles in a graph \( G \) (cycles which include every vertex of \( G \)), such necessary
conditions are much harder to find than the sufficient ones. However, since every hamiltonian graph has a perfect matching, from the above result a necessary condition for a graph to be hamiltonian is easily derived.

**Corollary 1.** If a graph $G$ of even order (that is $|V(G)|$ is even) is hamiltonian then $\nu(L(G)) = \frac{n}{2}$. ■

For odd order graphs, the above techniques can be adapted, adding a new vertex and linking by edges this vertex to the remaining ones.

**References**


Fast Interval Global Optimization Algorithm for Limited Computational Resources

L.G. Casado and J. Garcia
Dept. Computer Architecture and Electronics
University of Almería
Almería, Spain
Email: leo@miro.ualm.es
inma@monet.ualm.es

A variant of the Moore-Skelboe algorithm [7] with only the cut-off and midpoint test accelerating devices [5] is proposed. The Moore-Skelboe algorithm is suitable for minimizing non-differentiable functions. When accelerating devices, like monotonicity test, concavity test and interval Newton's method can not be applied, the set of n-dimensional interval vectors (boxes) stored in the work list can run out of the available random access memory, slowing down the execution of the algorithm. Mainly, this happens when a high accuracy of the solution is needed and/or the dimension of the problem, n, is high. In the hypothetical case when we chose to remove some boxes from the work list which selection rule would be the most appropriate?

According to the selection rule used in the Moore-Skelboe algorithm, the box X with minimal lower bound (obtained by the inclusion function \( L(X) \)) is selected. Usually, the inclusion function overestimates the real range \( f(X) \) of the objective function in the interval \( X \), and this overestimation grows with the width of the interval \( w(X) \). Therefore, it may happen that a box is selected, just because it is bigger than the others, and not because it contains a minimizer point. A new parameter which achieves more efficient comparisons is needed. The proposed parameter is defined as follows:

\[
pf(\bar{f},X) = \frac{\bar{f} - L(X)}{u(F(X))}
\]  

(1)

where \( \bar{f} \) is the current best upper bound of the global minimum, updated by the midpoint test. Note that the cut-off test is analogous to obtain a negative value of the \( pf \) parameter. In [1, 2, 3] the \( pf \) parameter has been successfully used as a predictor of the computational cost of a particular box, as an indicator whether a subinterval contains a minimizer point and as a parameter to determine the level of multifraction to carry out in a box, respectively.

Convergence properties of interval global optimization algorithms, using the \( pf \) parameter as selection criteria have been studied in [4].
In this work, a variant of the Moore-Skelboe algorithm with only the Cutoff and midpoint test is described and evaluated. Active intervals are stored into two different work lists depending on the value of their pf parameter. The following is a brief description of the algorithm.

1. Run the Moore-Skelboe algorithm while the number of boxes in the work list is less or equal than a threshold \( N \). Denote this work list by \( l_1 \).

2. Change the selection criterion to select those boxes with a maximal \( pf \) value first, instead of those with minimal \( f(X) \) first.

3. While \( l_1 \) is not empty, a new box \( X \) is selected from \( l_1 \) and processed. The subboxes generated from \( X \) are stored in another work list \( l_2 \), in a decreasing order based on the \( pf \) value. Therefore, the algorithm will not work on boxes stored on \( l_2 \). Some of the boxes in \( l_1 \) or \( l_2 \) could be eliminated by the Cutoff test or stored in a final list, if they meet the termination criterion.

4. If \( l_2 \) is empty, STOP.

5. Else, store in \( l_1 \) the first \( N \) boxes of \( l_2 \) and use a garbage collection procedure handling the remained boxes. The boxes with a \( pf \) value equal to the last one moved to \( l_1 \) are stored in \( l_1 \) too.


Here the same value of \( N \) have been used on step 1 and 5, but they can also be different.

**Numerical Results**

A set of preliminary numerical tests were carried out on a Pentium 166 MMX computer, running under Linux operating system. The program was coded in C and interval arithmetic was implemented via the BIAS routines [6].

As an example, in the table below, numerical results of the execution time in seconds (CPU), the number of function evaluations (NFE) and the value of the solution, \( f \), obtained for the problems Goldstein-Price (GP), Shkel 10 (S10), Hartman 6 (H6), Levy 3 (L3) and Six-Hump-Camel-Back (C) have been listed. The value of \( N \) was 20 and the termination criterion \( u(X) \leq 10^{-12} \) in all cases. Notice that a small number of function evaluations are needed to obtain the solution of all test functions compared to the traditional Moore-Skelboe algorithm.

<table>
<thead>
<tr>
<th>Problem</th>
<th>CPU</th>
<th>NFE</th>
<th>( f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP</td>
<td>0.17</td>
<td>5.174</td>
<td>3.00000000000000</td>
</tr>
<tr>
<td>S10</td>
<td>0.75</td>
<td>8.938</td>
<td>-10.53640816692</td>
</tr>
<tr>
<td>H6</td>
<td>1.56</td>
<td>14.929</td>
<td>-3.322388011415</td>
</tr>
<tr>
<td>L3</td>
<td>0.71</td>
<td>5.380</td>
<td>-176.541793136745</td>
</tr>
<tr>
<td>C</td>
<td>0.17</td>
<td>5.353</td>
<td>-1.031628453489</td>
</tr>
</tbody>
</table>

22
The talk will discuss the algorithm in detail and numerical results for an extended set of test functions will be shown.

References


Interval Branch-and-Bound Methods – a Systematic Approach *

Csallner A.E.
Department of Computer Science
Juhász Gyula Teacher Training College
Boldogasszony stg. 6
6725 Szeged (Hungary)
Email: csallner@jgytf.u-szeged.hu

Although interval branch-and-bound methods are not only applied to solve global optimization problems, for the sake of simplicity here we only consider this class of problems. One of the possible definitions of the global optimization problem is the following:

$$\min_{x \in \mathcal{I}} f(x)$$

(1)

where \(\mathcal{I}\) is a compact real interval. Several real-life bound-constrained global optimization problems are covered by (1), e.g., problems where the parameters are given with tolerances or if the optimizers are supposed to be inside a parameter region [2].

Problem (1) can be solved with verified accuracy with the aid of interval branch-and-bound methods (see, e.g., [1, 7]). By setting up a model algorithm, the different parts of it and the connections among these parts can be investigated separately. Moreover, defining each step of this model algorithm precisely, we have a particular method for solving global optimization problems. In this paper we summarize the variants we have investigated in the last few years.

Refining the Steps of the Algorithm

As mentioned above, each iteration step has free scope for variants depending on how it is fixed. The five promising parts are subdivision, storage, interval selection, deletion, and termination. Though none of our investigations has been exhaustive, they can mean the beginning of a systematic theoretical investigation of interval branch-and-bound methods.

The interval subdivision itself only means that the actually investigated subinterval is subdivided. Neither the coordinate direction of subdivision, nor the number and ratio of the resulting intervals are fixed. In the oral presentation

---

*This work has been partly supported by the Grant OTKA F 025743.
the effect of increasing the number of cuts in some special cases is summarized
[3, 6].

Hard to solve problems the storage consumption of interval branch-and-
bound methods can be enormous. It is shown what kind of storage handling
be used and how the efficiency can be improved [4].

One of the most important questions is the way selecting a subinterval from
the storage unit for further subdivision. A less general but steadier method is
chosen and investigated both for the worst case and in the case if the storage
unit can be purged by deleting unnecessary subintervals from the storage unit
[6].

At last the possible termination criteria and the tightness of approximation
using a Lipschitz-continuous objective function is surveyed [5].

References

[1] Alefeld G. and Herzberger J. (1983), Introduction to Interval Computations,

Hungarian Journal of Industrial Chemistry, 21, pp. 303–308.

for Global Optimization and the Joint Effects of Algorithmic Modifications,
IMACS/GAMM SCAN-85 Conference, Wuppertal, p. 33.

Global Optimization, Acta Cybernetica, 13, pp. 413-421.

[5] Csendes T., Markót M.Cs.(1999), Lipschitz Continuity and the Termination of Interval
Methods for Global Optimization, accepted for publication in Computers
and Mathematics with Applications.

Global Optimization, submitted for publication.

Kluwer, Dordrecht.
Convergence properties of interval global optimization algorithms with a new class of interval selection criteria

Tibor Csendes
Institute of Informatics
József Attila University
Szeged, Hungary
Email: csendes@inf.u-szeged.hu

The theoretical convergence properties of interval global optimization algorithms are investigated, that select the next subinterval to be subdivided according to a new class of interval selection criteria based on variants of the RejectIndex: \( p^{f^*}(X) = \frac{f^*(X) - f(X)}{F(X) - F(X)} \), a recently thoroughly studied indicator [1, 2, 3], that can quite reliably show which subinterval is close to a global minimizer point.

The original formulation of the RejectIndex was based on the known value of the global minimum, which is not always available. We introduce here a new expression for a wider class of indicators that still keep the basic properties of \( p^{f^*}(X) \): \( p(\hat{f}, X) = \frac{\hat{f}(X) - F(X)}{F(X) - F(X)} \), where the \( \hat{f} \) value will be a kind of approximation of the global minimum in the subsequent theoretical analysis. In the present investigation we assume that \( \hat{f} \in F(X) \), i.e., this estimation is realistic in the sense that \( \hat{f} \) is within the known widest bounds of the objective function on the search region.

Theoretical results

The following theoretical results could be proved:

**Theorem 1** Assume that the inclusion function of the objective function is isotone, it has the zero convergence property, and the \( p(f_k, Y) \) parameters are calculated with the \( f_k \) parameters converging to \( \hat{f} \approx f^* \), for which there exists a point \( \hat{x} \in X \) with \( f(\hat{x}) = \hat{f} \). Then the branch-and-bound algorithm that selects as next actual interval that interval \( Y \) from the working list which has the maximal \( p(f_k, Z) \) value can converge to a point \( \hat{x} \in X \) for which \( f(\hat{x}) > f^* \), i.e., to a point which is not a global minimizer point of the given problem.

**Theorem 2** Assume that the inclusion function of the objective function has
the zero convergence property, and \( f_k \) converges to \( \bar{f} < f^* \). Then the optimization branch-and-bound algorithm will produce an everywhere dense sequence of subintervals converging to each point of the search region \( X \) regardless to the objective function value.

**Theorem 3** Assume that the inclusion function of the objective function is isotone and it has the zero convergence property. Consider the interval branch-and-bound optimization algorithm that selects as next actual interval that interval \( Y \) from the working list which has the maximal \( p(f_k, Y) \) value. A necessary and sufficient condition for the convergence of this algorithm to a set of global minimizer points is that the sequence \( \{ f_k \} \) converges to the global minimum value \( f^* \) and there exist at most a finite number of \( f_k \) values below \( f^* \).

**Corollary 1** If our algorithm applies the interval selection rule of maximizing the \( p(f^*, X) = p(f^* (X)) \) values for the actual members of the list \( L \) (i.e., if we can use the known exact global minimum value), then the algorithm converges exclusively to global minimizer points.

**Corollary 2** If our algorithm applies the interval selection rule of maximizing the \( p(\bar{f}, X) \) values for the actual members of the list \( L \) where \( \bar{f} \) is the best available upper bound for the global minimum, then the algorithm converges exclusively to global minimizer points.

The interval branch-and-bound algorithm for global optimization is converging to the set of global minimizer points if the interval selection criterion is to choose that interval from the list \( L \) which has the maximal RejectIndex parameter value \( p(f_i, Y) \). The user may set \( \bar{f} \) to the global minimum value \( f^* \), if it is a priori known, or to the iteratively updated lowest upper bound of the global minimum, \( \bar{f} \). The latter parameter is always available, and thus no previous information is needed to ensure convergence. It is still an open question, what are the necessary conditions of the convergence. There is a possibility to utilise the advantages of both the Moore-Skelboe and the new algorithm according to the following theorem.

**Theorem 4** Assume that the inclusion function of the objective function is isotone and it has the zero convergence property. Consider the interval branch-and-bound optimization algorithm that selects as next actual interval that interval \( Y \) from the working list which has the maximal

\[
\min u(p(f_k, X), -\bar{f}(X))
\]

value. A sufficient condition for the convergence of this algorithm to a set of global minimizer points is that the sequence \( \{ f_k \} \) converges to the global minimum value \( f^* \), there exist at most a finite number of \( f_k \) values below \( f^* \) and that the utility function \( u(x, y) \) is strictly monotonously increasing in its both arguments.
On the other hand there is no theoretical proof of the reliability of the RejectIndex parameter. In an earlier paper [2] e.g. the following theorem was proved for the pf*(X) parameter and for the traditional Moore-Skelboe algorithm variant:

**Theorem 5** Assume that for an optimization problem $\min f(x)$ the inclusion function $F(X)$ of $f(x)$ is isotope, and $\alpha$-convergent with given positive constants $\alpha$ and $C$. Assume further that the RejectIndex parameter $pf^*$ is less than 1 for all the subintervals of $X$. Then an arbitrary large number $N(>0)$ of consecutive actual intervals of the basic B&S B Algorithm that selects the subinterval with the smallest lower bound as next actual interval may have the properties that:

1. neither of these processed intervals contains a stationary point, and
2. during this phase of the search the $pf^*$ values are maximal for these intervals.

According to the present theoretical study, intervals with zero width inclusion functions must be handled with special care, even though they are generally quite rare in real life problems.

**Numerical results**

The preliminary numerical tests were run on a 486 PC (100 Mhz.), the algorithms were coded in Fortran, and we used simple natural interval extension with outside rounding. In the tables below we have listed the used time in standard time units (STU), the number of iterations (NTI), the number of function evaluations (NFE), the number of derivative evaluations (NDE) and the necessary number of memory units to save the subintervals (MI).

As two simple examples the numerical results are shown here for the Goldstein-Price and for the Hartman-6 problems (both know as difficult to solve by interval methods) with known global optimum value and with, and without the monotonicity test, respectively.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Selection</th>
<th>STU</th>
<th>NTI</th>
<th>NFE</th>
<th>NDE</th>
<th>MI</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP</td>
<td>M-S</td>
<td>2055.7</td>
<td>17.446</td>
<td>34,883</td>
<td>46,887</td>
<td>3,213</td>
</tr>
<tr>
<td></td>
<td>new</td>
<td>34.8</td>
<td>503</td>
<td>1,007</td>
<td>1,710</td>
<td>502</td>
</tr>
</tbody>
</table>

The respective solutions were: $F(X) = [2.9999517, 3.0000483]$ at

$$X = ([-.50004645E-07,.00000000],[-.1.0000001,-1.0000001])^T$$

in both cases. The same result was obtained with the new technique at the cost of closely two order of magnitude less computation.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Selection</th>
<th>STU</th>
<th>NTI</th>
<th>NFE</th>
<th>NDE</th>
<th>MI</th>
</tr>
</thead>
<tbody>
<tr>
<td>H6</td>
<td>M-S</td>
<td>592.3</td>
<td>5,101</td>
<td>10,203</td>
<td>0</td>
<td>4,500</td>
</tr>
<tr>
<td></td>
<td>new</td>
<td>0.1</td>
<td>118</td>
<td>237</td>
<td>0</td>
<td>117</td>
</tr>
</tbody>
</table>
The respective solutions were: \( F(X) = [-3.3659900, -2.8455968] \) at

\[
X = \begin{bmatrix}
0.21875000, 0.25000000, 0.48750000, 0.15625000, 0.18750000, \\
0.46875000, 0.50000000, 0.21875000, 0.25000000, 0.56250000, \\
0.31250000, 0.34375000, 0.56250000, 0.62500000, 0.62500000
\end{bmatrix}
\]

and \( F(X) = [-3.3156227, -3.3156134] \) at

\[
X = \begin{bmatrix}
0.20312500, 0.20312595, 0.14843655, 0.14843750, \\
0.50000000, 0.50000095, 0.28125000, 0.28125095, \\
0.31152153, 0.31152344, 0.65747070, 0.65747261
\end{bmatrix}
\]

In the case of the Hartman-6 problem, the solutions are actually not comparable, since the traditional technique was unable to locate the solution (due to memory shortage), while the new algorithm could produce an even sharper estimation. Thus we can say here at least that the new method was by several order of magnitude better.

The preliminary numerical studies indicate a surprisingly large improvement in efficiency, especially in larger dimensional and other hard to solve problems. It must be noted, however, that more sophisticated interval techniques, equipped with all the available acceleration devices can ensure better relative efficiency results, especially on easy to solve problems.

We can conclude that the new algorithm does promise better efficiency utilizing the easy to obtain information of the relative place of the global minimum value within the inclusion interval of the objective function for actual intervals, and thus it can be a successful alternative of the Moore-Skelboe and Hansen algorithms. The talk will discuss results of an extensive computational testing together with a detailed theoretical investigation.

References


Global optimization and multiple-funnel landscapes: Lennard-Jones clusters

Jonathan Doye
University Chemical Laboratory, Lensfield Road, Cambridge CB2 1EW, United Kingdom
Email: jpkd1@cam.ac.uk

One of the biggest applications of global optimization in the chemical physics and biophysics communities is to the global minimization of the energy for a (macro)molecular system. For this class of problems, the performance of a global optimization algorithm is intimately linked to the topography of the system’s energy landscape. It is the thesis of this paper that understanding this relationship can provide important physical insights into the optimization process, and thus aid the development of improved algorithms.

One way of characterizing an energy landscape (that originated in the field of protein folding) is in terms of funnels [1]. A funnel is a set of downhill pathways that converges on a single low-energy structure. When a energy landscape possesses a single funnel which ends at the global minimum, global optimization should be relatively easy, despite the potentially huge number of minima on the energy landscape, because on relaxation down the energy landscape the topography guides the system towards the global minimum. However, when the landscape has more than one funnel, potential problems for global optimization begin to arise. In this case, the short-time relaxation dynamics of the system would take the system into one of the funnels with a probability related to its kinetic accessibility. The subsequent interfunnel dynamics can then be very slow [2]. Therefore, if the kinetically most accessible funnel ends at a low-energy, but non-global, minimum, this funnel can act as a trap preventing the system reaching the global minimum.

Lennard-Jones (LJ) clusters, a model system much-used in the testing of global optimization algorithms, nicely illustrate the relationship between the energy landscape and the ease of global optimization. Most of the global minima for small LJ clusters have structures based on the Mackay icosahedra [3, 4]. These are relatively easy to locate because the energy landscape has a single-funnel. (The sizes at which complete Mackay icosahedra can be formed (N = 13, 55) are particularly easy to optimize because of the steepness of the funnel.)

There are also a few examples where a LJ cluster has a double-funnel energy landscape. For these cases the global minimum is not icosahedral but either has a face-centred-cubic structure (LJ36) or is based upon a Marks decahedron.
In addition, the thermodynamic and dynamic properties of these clusters make these clusters particularly difficult to optimize [5, 6]. On cooling from the liquid-state the cluster invariably ends up in the icosahedral funnel. This greater kinetic accessibility has a number of causes. Firstly, the icosahedral funnel is the state of lowest free energy below the melting temperature (only at lower temperatures does the global minimum become most stable). Secondly, it is wider than the funnel associated with the global minimum. Both these causes stem from the greater number of low-energy icosahedral minima. Thirdly, less structural reorganization is required to go from a liquid configuration to an icosahedral structure, because they have some common polytetrahedral character. Furthermore, there is a large free energy barrier between the icosahedral funnel and the funnel of the global minimum making the interfunnel dynamics very slow. For example, for LJ$_{38}$ at the temperature where the face-centred-cubic funnel becomes most stable, the free energy barrier has been found to be $\sim 17kT$ [7] and the canonical rate constant for interfunnel passage has been estimated to be $43e^{-4}$ (using parameters appropriate for argon) [8].

Given the above properties of the LJ clusters with a double-funnel energy landscape, optimization methods, such as simulated annealing, that rely on the real dynamics of the system are going to fail. However, the hypersurface deformation class of global optimization methods [9], in which the aim is to transform the energy landscape to a form that makes optimization easier but leaves the identity of the global minimum unchanged, can potentially overcome this problem because the transformation will change the thermodynamics and dynamics of the system. To take advantage of this potential solution, physical insight is required. Firstly, to understand what changes to the thermodynamics would best alleviate the problems associated with double funnels, and secondly to understand what transformations would achieve such changes.

We shall approach this task in reverse by analysing an algorithm, ‘basin hopping’, that we know to work for LJ$_{38}$ and LJ$_{75-77}$ [4]. In this method the transformed energy, $\tilde{E}$, is defined by:

$$\tilde{E}(\mathbf{X}) = \min \{ E(\mathbf{X}) \},$$

where $\mathbf{X}$ represents the $3N$-dimensional vector of nuclear coordinates and $\min$ signifies that an energy minimization is performed starting from $\mathbf{X}$. Hence, the energy at any point in configuration space is assigned to that of the local minimum, and the energy landscape is transformed into a stepped landscape with plateaus corresponding to the basins of attraction surrounding each minimum. By searching this energy landscape with Metropolis Monte Carlo at a fixed appropriately-chosen temperature we were able to find all the LJ global minima in the size range we have investigated ($N < 110$) [4].

Firstly, we should note that the identity and energy of the global minimum is unchanged by the transformation. Secondly, the transformation has a clear effect on the dynamics: it removes downhill barriers and enables transitions to occur between basins of attractions anywhere along the basin boundaries (on the untransformed energy landscape transitions can only occur when the system
passes along a transition state valley. These changes make relaxation down a funnel easier (e.g., the algorithm can find the global minimum of LJ$_{55}$ in on average fewer than 150 steps when started from a random geometry [6]), but they do not necessarily make interfunnel motion significantly easier, because the cluster must still pass over the high-energy minima that separate the funnels.

If we examine the thermodynamics for the double-funnel clusters, we find that the global minimum is stabilized with respect to the icosahedral states and the higher-energy minima. The transitions between the states occur at higher temperature and are considerably broadened. So much so that for LJ$_{55}$ the transition from the face-centred-cubic to icosahedral funnels overlaps significantly with the 'melting' transition. Therefore, there is now a temperature window where both the global minimum and the higher-energy minima that are at the top of pathways between the two funnels have significant populations, i.e., the free energy barriers between the funnels are small. These changes to the thermodynamics are the key to the success of the basin-hopping algorithm for double-funnel energy landscapes.

We can understand these changes to the thermodynamics if we compare the partition functions for a minimum for the two landscapes. For the untransformed energy landscape, the partition function includes a vibrational term that favours the less-rigid higher-energy minima. By contrast, for the transformed energy landscape the partition function of a minimum is proportional to the hyperarea of the basin of attraction of that minimum. As the basins of attraction become smaller the higher the energy, this term favours the global minimum.

Although the "basin-hopping" transformation makes global optimization easier on multiple-funnel landscapes, the computational effort is still considerably more than for single-funnel landscapes. For example, for both LJ$_{38}$ and LJ$_{75-77}$ the average number of steps to reach the global minimum is considerably more than for clusters of comparable sizes that have icosahedral global minima. Furthermore, although the problems represented by LJ$_{38}$ and LJ$_{75-77}$ are of similar nature, they are in a much more extreme form for the latter. Reflecting this fact, quite a number of methods have now found the global minimum of LJ$_{38}$, but far fewer are able to find the Marks decahedra.

References


Dual Bounding Procedures in Branch-and-Bound Algorithms

Mirjam D'Ur
Dept. of Statistics
Vienna University of Economics
Augasse 2-6
A-1090 Vienna (Austria)
Email: Mirjam.Duer@wu-wien.ac.at

A well-known technique for solving global optimization problems is the Branch-and-Bound method. In this method, the feasible set is successively partitioned and for each partition set upper and lower bounds are computed. It depends on the particular setting of the problem which technique can be used to compute these bounds.

In this talk, we investigate the recently proposed idea to compute lower bounds in Branch-and-Bound algorithms by solving Lagrange-dual problems. We give a simple, but powerful estimate of the duality gap and prove that, under very mild conditions, the duality gap shrinks to zero provided that the sequence of partition sets generated by the Branch-and-Bound procedure converges to a set that is entirely feasible for the original problem. This result entails that the bounding procedure given by solving Lagrange-dual problems leads to convergent Branch-and-Bound algorithms.

A similar result holds for problems that are decomposable into a convex and a nonconvex part. In this case, it is possible to apply decomposition strategies which reduce the computational effort to solve the problem. For example, one can perform the computationally expensive partitioning procedure in the space of nonconvex variables only. We show how the dual bounding procedure presented for general nonconvex problems can be modified for such partly convex problems.

Another subproblem arising in Branch-and-Bound algorithms is to decide whether or not a partition set generated by the algorithm contains a feasible point. We discuss whether our dual bounding procedure is able to detect this. It turns out that infeasibility of a partition set will not necessarily be detected by the procedure. This can be guaranteed only in special cases. In general, an additional feasibility check is needed for every partition set. Normally, this can be accomplished while upper bounds are computed.

Finally, we conclude with remarks on possible practical applications. The dual bounding procedure can be applied whenever the dual problem is easy to
solve. We point out some optimization problems in which the dual problem can be reformulated as a linear problem: This is the case with certain generalized bilinear problems (minimize a linear function with respect to linear and generalized bilinear constraints), with the sum-of-ratios problem (maximize a sum of ratios of affine functions over a polytope) and the problem of minimizing a concave function under reverse convex constraints, i.e., constraints of the form $g(x) \leq 0$, where $g$ is a concave function.
A Global Optimization System for Explicit Problems

A.Frshov, O.Khamisov
Institute of Energy Systems, Irkutsk, Russia

In our report we present the first version of a software developed for solving explicitly (analytically) given global optimization problems. The latter means that objective and constraints in the problem are constructed from the set of elementary functions by multiplication, addition, subtraction, division, composition, taking the maximum or minimum over finite set of functions and the like. In [3] such functions (although without max and min operations) are called factoriable functions. In principle, almost all considered functions can be represented as a difference of two convex functions, i.e. they are d.c. functions [1]. Using whenever it is possible the well developed d.c. optimization methods we tried to implement algorithms based on construction of a convex support majorant functions and concave support minorant functions to separate a point which does not belong to the epigraph or subgraph of the nonconvex objective function. Let \( X \subset \mathbb{R}^n \) be a convex compact set.

**Definition 1** Function \( f(x) : X \to E^1 \), is said to have a concave support minorant and a convex support majorant if there exist functions \( \mu^-(x,y), \mu^+: E^1 \times X \to E^1 \) and \( \mu^-(x,y), \mu^+: E^1 \times X \to E^1 \), such that

1. \( \mu^-(x,y) \) is continuous and concave in \( x \) for fixed \( y \in X \);
   \( \mu^+(x,y) \) is continuous and convex in \( x \) for fixed \( y \in X \);
2. \( \mu^-(x,y) \leq f(x) \leq \mu^+(x,y), \forall (x,y) \in X \times X \);
3. \( \mu^-(y,y) = f(y) = \mu^+(y,y), \forall y \in X \);

The functions \( \mu^-(x,y), \mu^+(x,y) \) are called concave support minorant function and convex support majorant function, respectively. The class of functions that satisfy Definition 1 is quite large. For example, every Lipschitzian function belongs to this class. It is not difficult to see that functions with concave and convex support functions are continuous. The inverse statement is not true.

**Theorem 6** Let \( h(x) \) and \( g(x) \) satisfy Definition 1. Then the functions \( f_i(x), i = 1, 5 \) defined below, also satisfy Definition 1.

\[
\begin{align*}
f_1(x) &= \lambda \ast h(x) + \beta \ast g(x), \lambda, \beta \in E^1; \\
f_2(x) &= h(x) \ast g(x) \\
f_3(x) &= \frac{h(x)}{g(x)}, \text{ provided that } g(x) > 0;
\end{align*}
\]

36
\[ f_1(x) = \max\{h(x), g(x)\}; \]
\[ f_2(x) = \min\{h(x), g(x)\}. \]

The proof of the theorem is constructive, i.e. if we are given by concave and convex support functions of \( h(x) \) and \( g(x) \), then we can compute convex and concave support functions for every \( f_i(x) \) in the theorem.

The problem of global optimization of a function with convex and concave support functions subject to constraints of the same type can be approximated by a sequence of d.c. (sometimes concave) programming problems. The set of algorithms consists of two parts. In the first part the algorithms for constructing concave and convex functions for elementary functions are included. The second part contains algorithms for constructing convex and concave support functions for compositions of elementary functions. As soon as a function in analytical form appears it is analysed which elementary functions the given function contains and which operations are involved. Then algorithms described in [2] are used to construct convex and concave support functions for the composite function.

The aim of the software is to free (as much as we could) the user from coding in solving global optimization problems. The expected size of the problems is between one and ten variables. The reason for that is using enumerative algorithms for solving auxiliary d.c. and concave programming problems.

References


Nonlinear Optimal Control Models in Law Enforcement

Gustav Feichtinger
Institute for Econometrics, OR and Systems Theory
University of Technology
Argentinierstrasse 8, A-1040 Vienna, Austria

In this paper we present an intertemporal extension of Becker’s static economic approach to crime and punishment. For a dynamic supply of offenders we determine the globally optimal dynamic trade-off between damages caused by offenders, law enforcement expenditures and cost of imprisonment. By using Pontryagin’s maximum principle we obtain interesting insights into the dynamical structure of optimal law enforcement policies. It is found that inherently multiple-equilibria are generated which can be saddle-points, unstable points and boundary solutions. As in other nonlinear control models there exist thresholds (denoted as Dechet-Nishimura-Skiba points) which makes the optimal enforcement policy dependent on the initial conditions. It turns out that above this threshold the optimal trade-off between social costs implies an equilibrium with a high level of offences, while below the point the optimal law enforcement should eradicate crime.

There has been considerable debate about what share of drug control resources should be allocated to prevention treatment and enforcement. It seems plausible that the optimal mix of interventions might vary as the size of the problem changes. In the second part of the lecture the choice between the instruments formulated as an optimal control problem. It turns out that under realistic conditions DNS-thresholds occur, and this path-dependence makes economic sense. Furthermore, we discuss the occurrence of stable periodic oscillations in certain dynamic models for drug use.

The purpose of this model is to illustrate a real-world application of optimal control to operations research. The model we discuss is validated with empirical data from the US cocaine epidemics. The aim of our study us to derive management implications and concrete advises for optimal law enforcement and other measures the reduce the consumption of illicit drugs.
Global Optimization for the Graph Coloring Problem

Paola Festa
Dept. Matematica e Informatica
Università di Salerno
via S. Allende
84081 Baronissi (SA)
Italy
Email: paofes@udsab.dia.unisa.it

Panos M. Pardalos
Center for Applied Optimization
Industrial and Systems Engineering
Univ. of Florida
303 Weil Hall
PO Box 116595
Gainesville, FL 32611-6595
Email: pardalos@ufl.edu

Continuous approaches have been already used for solving several discrete optimization problems, including integer programming, the maximum clique problem, the satisfiability problem, the hamiltonian cycle problem, and the Steiner problem. There are several techniques to formulate a discrete problem as an equivalent continuous problem. The discrete feasible domain of the original problem is embedded in a larger continuous space, whose topological and geometric structure play a fundamental role. In fact, by exploring the properties of the obtained larger continuous space, new methods can be developed that lead to efficient algorithms for computing (sub)optimal solutions to the original discrete optimization problem.

In this talk we consider several continuous approaches for the Graph Coloring Problem.

In addition, we consider the problem of constructing test graphs with a known chromatic number.
Global Optimization of Mixed-Integer Nonlinear Problems

Christodoulos A. Floudas
Department of Chemical Engineering
Princeton University
Princeton, N.J. 08544-5263, USA
Email: floudas@titan.princeton.edu

Two novel global optimization algorithms for nonconvex mixed-integer problems are proposed. The first one, referred to as the Special Structure Mixed-Integer $\alpha$BB algorithm (SMIN-$\alpha$BB), is based on the $\alpha$BB algorithm for twice continuously differentiable NLP problems. The SMIN-$\alpha$BB algorithm addresses problems that involve functions with a nonconvex part in the continuous variables and linear and mixed-bilinear terms for the binary variables. The main features of the SMIN-$\alpha$BB algorithm are the generation of a guaranteed lower bound on the nonconvex MINLP through the solution of a convex MINLP, the branching on binary and continuous variables and efficient variable bound tightening strategies. The second global optimization approach, referred to as the General Structure Mixed-Integer $\alpha$BB algorithm (GMIN-$\alpha$BB), is applicable to very general problems for which the only assumption is that the continuous relaxation is twice continuously differentiable. The solution space is explored through branching on the integer variables. Valid lower bounds on nonconvex MINLPs are obtained by applying the $\alpha$BB algorithm to a continuous relaxation of the MINLP. Finally, variable bound updates play a significant role in the performance of the algorithm. Both algorithms have been successfully tested on several nonconvex MINLPs, including a series of small problems, three heat exchanger network synthesis problems, a pump network synthesis problem and four instances of a trim loss minimization problem that arises in the paper cutting industry.
Stochastic Optimisation Methods for Cost-Effective Quality Assessment in Health

Dimitris Fouskas
Department of Mathematical Sciences
University of Bath
Claverton Down
Bath BA2 7AY England
Email: df@maths.bath.ac.uk

David Draper
Department of Mathematical Sciences
University of Bath
Claverton Down
Bath BA2 7AY England
Email: d.draper@maths.bath.ac.uk

Introduction. The US and UK governments have recently begun to examine more closely the quality with which they carry out their national health mandates, focusing in particular on what goes on inside hospitals. One method for indirectly assessing the quality of hospital care is a kind of input-output approach, in which hospital outcomes are compared after adjusting for differences in inputs. With death an important outcome to highlight, the input-output approach takes the form of a contrast between observed and expected mortality rates; given how sick patients are when they arrive at the hospital (e.g., Draper 1995). A broad implementation of input-output hospital quality assessment in the US or UK would involve data-gathering on 100K+ patients/year, making the cost-effective measurement of admission sickness crucial to the success of health policy initiatives of this kind.

Clinical expert judgement typically identifies $O(10^2)$ variables relevant to the construction of a sickness scale for any given disease. The standard way to construct such a scale uses logistic regression with death (within 30 days of admission) as the outcome, sifting through the available sickness indicators using standard frequentist variable-selection methods to find a parsimonious and clinically reasonable subset. From a cost-effectiveness point of view this approach is deficient in that it takes no account of differences in cost of data collection among the available predictors, which can vary by multiplicative factors of up to 100. When both data-collection cost and accuracy of prediction of 30-day
mortality are considered, a large optimisation problem arises in which costly variables that do not predict well enough should be omitted from the scale.

In this paper we take a Bayesian decision-theoretic approach (based on maximisation of expected utility) to solving this optimisation problem. We use simulated annealing and tempering, genetic algorithms, tabu search, and other methods from the optimisation literature to find optimal or near-optimal subsets of predictor variables.

The utility structure. Any reasonable utility function in this problem will have two components, one quantifying data collection costs associated with the construction of a given sickness scale, the other rewarding and penalising the scale’s predictive successes and failures. We use data on approximately $n = 2700$ pneumonia patients from the RAND Corporation DRG Quality of Care Study (Kahn et al. 1990), which was based on a large nationally-representative sample of elderly patients hospitalised in America between 1980 and 1986. We randomly and repeatedly divide the data into modeling and predictive validation subsets, making predictions with the former and quantifying their accuracy with the latter. Using brute-force enumeration based on 500 random modeling/validation splits and taking the RAND 14-variable pneumonia admission sickness scale as an example, we have shown (Figure 1) that the full RAND scale is nowhere near optimal when data collection costs are considered along with predictive accuracy—indeed, any of several scales with 4–7 well-chosen variables would save about $8/patient record (translating into cost savings of well over $1 million/year if input-output screening were to be performed widely in the US or UK).

![Figure 1. Estimated expected utility as a function of number of predictors retained.](image-url)
References


A Radial Basis Function Method for Global Optimization

Hans-Martin Gutmann
Department of Applied Mathematics and Theoretical Physics
University of Cambridge
Silver Street
Cambridge CB3 9EW, United Kingdom
Email: H.M.Gutmann@damtp.cam.ac.uk

We present a method for the global optimization problem

find $x^* \in D$ such that $f(x^*) \leq f(x)$, $x \in D$,

where $D \subseteq \mathbb{R}^d$ is compact and $f : D \to \mathbb{R}$ is continuous. In many important applications the objective function $f$ is nonconvex and has a large number of local minima, but it is essential or desirable to find the global optimum. Our method is mainly intended for problems where only function values are available and each function evaluation is very expensive. For example, one has to build an experiment, to run a computer simulation or to solve a numerical problem. Popular methods are Simulated Annealing and Evolutionary Algorithms. However, they might require a lot of function evaluations and are therefore very slow in the applications in question. Recently, methods have been developed that require much less function evaluations on a set of standard test functions (Huyer, Neumaier (to appear), Jones, Perttunen, Stuckman (1993), Jones, Schonlau, Welch (1998)).

Our method is based on a general method proposed by Jones (1996). It is assumed that a function space $A$ is given such that for any points in $D$ and data values at these points an interpolating function from $A$ can be found. In addition, a measure of the "bumpiness" of a function $s \in A$ should be available. Now if points $x_1, \ldots, x_n \in D$ and their function values have been calculated, the following scheme is employed to determine the next point $x_{n+1}$. An estimate $f^*$ of the optimal value, the so-called target value, is chosen. For any $y \in D$ that is different from the $x_i$, there is a function $s_y \in A$ that interpolates the given data and the target value $f^*$ at $y$. We take the view that the "least bumpy" of these interpolants gives the most likely location of a point with function value $f^*$. This means, the optimization problem

$$\min \sigma(s_y), \quad y \in D \setminus \{x_1, \ldots, x_n\},$$
has to be solved.

In our method the function space \( \mathcal{A} \) is a space of radial basis functions, i.e. functions of the form

\[
s(x) = \sum_{i=1}^{n} \lambda_i \phi(||x - x_i||) + p(x),
\]

with \( n \in \mathbb{N} \), \( x_1, \ldots, x_n \in \mathcal{D} \), real coefficients \( \lambda_1, \ldots, \lambda_n \) and a polynomial \( p \in \Pi_m \), where \( \Pi_m \) is the space of polynomials of degree at most \( m \). In addition, the coefficients \( \lambda_i \) have to satisfy

\[
\sum_{i=1}^{n} \lambda_i q(x_i) = 0, \quad q \in \Pi_m.
\]

Popular choices of \( \phi \) are \( \phi(r) = r^3 \) (cubic splines), \( \phi(r) = r^2 \log r \) (thin plate splines), \( \phi(r) = r \) (linear splines), \( \phi(r) = \sqrt{r^2 + c^2} \) (multiquadrics) and \( \phi(r) = e^{-r^2} \) (Gaussian splines). The motivation for a suitable measure of bumpiness comes from the theory of natural cubic splines in one dimension. They can be written in the form (1) with \( \phi(r) = r^3 \) and \( m = 1 \). It is well-known that an interpolant \( s \) of the form (1) to a function \( f \) at \( x_1, \ldots, x_n \) minimizes the integral \( \int_{\mathbb{R}} f''(x)^2 \, dx \) among all functions \( g : \mathbb{R} \to \mathbb{R} \) that also interpolate \( f \) at these points and for which the integral exists. For a natural cubic spline, the integral can be written as

\[
\int_{\mathbb{R}} s''(x)^2 \, dx = 12 \sum_{i=1}^{n} \lambda_i s(x_i).
\]

The right-hand side now can be used to define a measure of bumpiness in this special case, namely as \( \sigma(s) \coloneqq (\sum_{i=1}^{n} \lambda_i s(x_i))^{1/2} \). Further, it can be generalized to other types of radial basis functions. If additionally the polynomial degree \( m \) is restricted (\( m \geq 1 \) in the cubic and thin plate spline case and \( m \geq 0 \) in the linear and multiquadric case; there are no restrictions in the Gaussian case), \( \sigma \) is a semi-norm on \( \mathcal{A} \).

We show that, under certain assumptions on the choice of the target values \( f^* \), the method converges to a global minimizer, if the objective function \( f \) satisfies the following condition.

**Condition** Let \( x_1, \ldots, x_n, n \in \mathbb{N} \), be given, and assume that there exists a unique interpolant \( s \in \mathcal{A} \) to \( f \). Then

\[
\sigma(s) \leq K,
\]

where \( K \) does not depend on the choice of the interpolation points.

Furthermore, in the cases of cubic or linear splines in odd dimensions and thin plate splines in even dimensions, a stronger result can be proved. Specifically, if
the target values are chosen carefully, one can show convergence of the method
even for a continuous objective function that need not satisfy the condition
above. This is achieved by proving that the sequence of generated points is
dense in $\mathcal{D}$.

It turns out that, given $x_1, \ldots, x_n$, the radial basis function interpolant $s$ to $f$
at these points and a target value $f^*$, the next iteration point $x_{n+1}$ can be found
by minimizing a function that is infinitely differentiable on $\mathcal{D}\setminus\{x_1, \ldots, x_n\}$, that
is strictly positive and that has poles at $x_1, \ldots, x_n$. Unfortunately, it has sev-
eral local minima, so this subproblem is a global optimization problem itself,
but much easier to solve than the original problem. We demonstrate how this
subproblem can be tackled.

A crucial issue for the performance of the method in practice is the choice of
the target values. A balance between local search ($f^*$ is close to the minimum
of $s$) and global search (the difference between the minimum of $s$ and $f^*$ is
very large) must be found. We carried out numerical experiments on a set
of test functions using several different strategies for the choice of $f^*$. Most
of the strategies give favourable results compared to other global optimization
methods, especially the ones mentioned in the first paragraph.

References


mization Without the Lipschitz Constant”, Journal of Optimization Theory

at the Fifth SIAM Conference on Optimization, Victoria, Canada.

tion of Expensive Black-Box Functions”, Journal of Global Optimization, 13,
no. 4, 455-492.
Object Oriented Approach for Direct Methods in Global Optimization

Beidi Hamma
Parallel Algorithms Team, CERFACS, France
Resource Optimization Consulting, ILOG, France

Keywords. Global Optimization, Object Oriented Programming (OOP), Direct Methods in Optimization, Moving Polytope, Simulated Annealing.

The work presented in this paper was first motivated by the needs of Object Oriented Approach for optimization algorithms within the project ODESIM, a High Performance Computing and Networking (HPCN) project funded by the European Commission within ESPRIT programme. The project acronym stands for Optimum DESIGN of Multi-Body systems (MBS). The starting point for the implementation of the ODESIM system was the CAD and MBS simulation packages available within the consortium. They then had to be improved with new modules for HPCN and optimization. For an efficient and reliable implementation, it has been adopted to use standards like UNIX, C++, X-Windows and other internal tools, based on object oriented programming techniques which will guarantee the openness, portability and scalability of the system.

Given that, we designed a general object oriented framework which enables to encapsulate any optimization method.

We considered all optimization problems: Unconstrained, Constrained, availability (or not) of Derivatives, availability (or not) of Second Derivatives. We also considered all type of methods for solving those problems: non-derivative methods, gradients methods, global optimisation methods. This enables to have a very simple and efficient interface which will allow any user to easily define his problem, and to add any new optimization solver to his general framework.

This paper is concerned with some examples of OO approach for Direct Methods: Moving Polytopes and Simulated Annealing. In the first case we show how the OO approach enables to unify the design and implementation of three Moving Polytope algorithms: the original simplex method of Nelder and Mead [1], the direct search method of Dennis and Torczon [2] and the adaptive search of Hamma [3]. In the second one, we show how a general OO approach for Simulated Annealing can be achieved. In both cases, the OO approach enables to plug these packages into any other OO framework and, furthermore, we can easily use a method with its different variants and/or mix the algorithms inside a same package.
The testing results in this work showed how the use of Object-Oriented approach in the design of numerical optimization libraries, gives the following capabilities:

1. better optimization interface in the case where objective functions (and gradients) are evaluated by the output of a simulation that is, as a black-box for the optimizer

2. rapid encapsulation of different optimization routines already written in C, Fortran etc. This encapsulation reduces the user's dependency on the way original routines are implemented and work.

3. rapid evaluation of several optimization routines on a same problem.

4. rapid and efficient combination of several optimization routines in the purpose of global optimization.

5. better parallelization using other object-oriented heterogeneous parallel managers.

6. better code maintenance and re-usability.
Parallel Optimization Algorithms for Multi Body Systems in High Performance Computing and Networking Environment

Beidi Hamma
Parallel Algorithms Team, CERFACS, France
Resource Optimization Consulting, ILOG, France

Luc Giraud
Parallel Algorithms Team, CERFACS, France

Jose M. Jimenez, N. A. Chehayeb
CMP SL, San Sebastian, Spain

Keywords. Parallel (Global) Optimization, Design of Multi-Body Systems (MBS), High Performance Computing and Networking (HPCN), Parallel Computing on Clusters of Heterogeneous Workstations, Parallel Virtual Machine (PVM)

In this paper we present some parallel optimization results achieved in ODESIM project, a High Performance Computing and Networking (HPCN) project funded by the European Commission within ESPRIT programme. The project acronym stands for Optimum DESign of Multi-Body systems. The project main objective has been the development of a set of software tools for finding optimal designs of MBS for kinematic and dynamic simulation. One part of the work in this project was the definition of parallel optimization algorithms suitable for MBS (Multi-Body Systems) and their implementation in a HPCN framework. In a first stage we targeted local minimization but afterwards we realized that in practice MBS systems may have more than one local minimum. Therefore we defined global optimization strategies. In this paper we will present results given by one of these. The paper first presents the MBS problem, then the optimization methods chosen followed by the presentation the HPCN framework that we build to implement the whole process: from the design of a MBS model to its full parallel optimization using a virtual parallelization (based on PVM message passing interface), on a network of clusters of heterogeneous workstations.
Introduction

Current MBS simulation programs have powerful analysis capabilities that allow them to simulate complex systems with rigid and flexible bodies subject to different kinds of external forces, control laws, nonlinear effects, etc. Most of these programs assume that the dimensions of the machine are perfectly known and that the user is only interested in predicting its behaviour over time. However, this is not the case during the design phase, in which the designer starts from a draft design and has to refine its dimensions in successive iterations. With current MBS analysis programs, the designer must perform kinematic and/or dynamic analysis in each design step, assess the performance of the system, and accordingly, take a decision on what dimensions should be modified to improve the performance. This iterative process has two main drawbacks: first, it relies heavily on previous experience with similar designs and therefore can only be done by experts, and second, it is very time consuming since the designer has to repeat the assessment of performance, the decision making and the dimensional modification in every iteration step.

In ODESIM\(^1\), the multibody design problem is addressed from the beginning of the design process, which starts with the CAD definition of parts. In the CAD program, the designer creates a parametric or variational model of each part. From this parametric or variational model, the designer can make fast modifications of the geometry based on only a few dimensions that are taken as design variables. ODESIM will create a multibody model from the parametric definition of parts, that will later be optimized according to some objective function and design constraints.

MBS design and, in particular, dynamic optimization of mechanisms, is a very time consuming task that can only be addressed with High Performance Computing and Networking (HPCN) techniques for moderate to large size problems. ODESIM will make use of parallel computing in environments with shared-memory multiprocessor computers and/or clusters of heterogeneous workstations.

The main features of ODESIM can be summarized in the following points:

1. It is a software module for the kinematic synthesis and dynamic design of MBS using HPCN, linked to a parametric and variational interactive CAD modeller.

2. It is an interactive interface for the definition of design variables, geometry, objective function, etc., that will also permit the user to monitor the progress of the optimization process as well as to stop, modify and restart it at any time.

\(^1\)the ODESIM consortium: CASA and CEIT in Spain, CERFACS and MATRA-DataVision in France, Centro Recerca FTAT in Italy, and SIEMENS in Germany
3. It implements and runs the most time-consuming tasks in a HPCN environment making use of parallel processing in multi-CPU computers and clusters of workstations.
Evolutionary Pattern Search and Hybrid Evolutionary Algorithms  
(Extended Abstract)

William E. Hart  
Dept. Applied Mathematics  
Sandia National Laboratories  
P.O. Box 5800  
Albuquerque, NM 87185-1110, USA  
505-844-2217  
Email: wehart@sandia.gov

Evolutionary pattern search algorithms (EPSAs) are a class of evolutionary algorithms for unconstrained or bound constrained optimization that are inspired by recent analyses of pattern search methods. EPSAs carefully modify the step size of the mutation operator in response to the success of previous optimization steps to ensure a probabilistic stationary point convergence on nonconvex problems. We reconsider this analysis for EPSAs that only use a mutation operator. In this case, we show how the analysis can be simplified and generalized. In particular, the analysis of EPSAs can be extended to include hybrid EPSAs that use a local pattern search algorithm to refine points during the evolutionary search.

4 Introduction

We consider the application of evolutionary algorithms (EAs) to solve minimization problems of the form

\[
\min_{x \in \mathbb{R}^n} \quad f(x) \quad \text{subject to} \quad l \leq x \leq u, \tag{1}
\]

where \( f : \mathbb{R}^n \to \mathbb{R}, l, u \in \mathbb{R}^n, \) and \( l < u. \) These problems have been solved using EAs like evolutionary programming (EP) [1], evolutionary strategies (ESs) [11] and genetic algorithms (GAs) [2]. In this paper we consider evolutionary pattern search algorithms (EPSAs), a class of EAs that can be used to solve problem (1). EPSAs are adaptive EAs like EPs and ESs, which modify the mutation step length during optimization. However, EPSAs have an absolute step length

52
that is used to generate a finite set of offsets, while EPs and ESs generate mutations by adding a continuous random variable that is scaled by a step length parameter. Also, EPSAs adaptively modify the mutation step length using a global step length parameter instead of the per-individual step length parameter commonly used by ESs and EPs.

Like most other EAs, EPSAs are direct search methods, methods that do not use either the derivative or an approximation to the derivative of f to perform optimization. EPSAs are, however, distinguished from other EAs by the fact that they can be cast as stochastic pattern search methods. Pattern search methods are direct search methods that examine a pattern of exploratory moves in search of points with lower functional values [12, 10].

The close relationship between EPSAs and pattern search can be exploited to show that for a continuously differentiable function the sequence of best points found by an EPSA, \{x_k^o\}, has the property that

\[ P(\lim_{k \to \infty} \inf \|g(x_k^o)\| = 0) = 1, \]

where \(g(x)\) is the gradient of \(f\) at \(x\) [7]. For a bound constrained continuously differentiable function a subsequence of \(\{x_k^o\}\) converges almost surely to a constrained stationary point.

Preliminary analyses of EPSAs are given in Hart [4, 5, 7]. Our present analysis extends the analysis in Hart [7] to allow the mutation step length to vary for different members of an EPSA's population. Further, we use this analysis to prove convergence for hybrid EPSAs using pattern search to refine points in each iteration.

5 EPSAs Without Recombination

Consider the pseudo code in Figure 3. This pseudo code describes a class of EPSAs that do not use recombination. Like EPs and ESs, EPSAs couple a step length parameter, \(\Delta\), with each point in the population, \(x\). Additionally, the vector \(\eta\) is coupled with the point \(x\) to keep track of the mutation offsets that have been applied to \(x\).

Mild restrictions are placed upon the selection and compose functions to ensure that (a) the best point in the population is selected with probability of at least \(\pi > 0\) in each iteration and (b) the best point in \(P_t \cup P\) is is always included in \(P_{t+1}\). The call to uint(\(j\)) uniformly generates an integer from 1 to \(j\). The \(\mu\) is the probability of mutating a point in the population by adding a mutation offset. The method update performs the standard pattern search update, which only contracts \(\Delta\) if all of the mutation offsets about a point have worse function values.

Our prior convergence analysis of EPSAs [7] shows how the individuals generated at each generation can be viewed as patterns in a pattern search with respect to the best individual in the population. Our analysis of the EPSAs defined in Figure 3 differs in that the EA is viewed as simultaneously executing \(N\)
(1) Given \( S = \{s_1, \ldots, s_m\} \), where \( s_i \in \mathbb{Z}^n \) forms a positive basis

(2) Given \( P_t = \{p_{1}^t, \ldots, p_{n}^t\} \), where \( p_i^t = (\Delta, x, \eta) \in \{\Delta_0\} \times \mathbb{Q}^n \times \{0\}^m \)

(3) \( x_0^* = \arg \min \{f(p_{1}^0(x)), \ldots, f(p_{n}^0(x))\} \)

(4) Repeat \( t = 0, 1, \ldots \)

(5) \( T = \text{selection}(P_t) \)

(6) For \( i = 1 : N \)

(7) \( \hat{p}_i = \overline{p}_i \)

(8) If (\( \text{unif}() < \mu \)) then

(9) \( j = \text{unif}(m) \)

(10) If (\( \overline{p}_j(x) + \overline{p}_j(\Delta) \cdot s_j \) is feasible)

(11) \( \hat{p}_i(x) = \overline{p}_i(x) + \overline{p}_j(\Delta) \cdot s_j \)

(12) If \( f(\hat{p}_i(x)) \geq f(\overline{p}_i(x)) \) then

(13) \( \hat{p}_i(x) = \overline{p}_i(x) \)

(14) \( \hat{p}_i(\eta) = 1 \)

(15) \( P_{t+1} = \text{compose}(P_t, P) \)

(16) \( x_0^{t+1} = \arg \min \{f(x_1^{t+1}), \ldots, f(x_N^{t+1})\} \)

(17) For \( i = 1 : N \)

(18) \( \text{update}(p_i^{t+1}(\Delta)) \)

(19) Until some stopping criterion is satisfied

---

**Figure 3: Pseudo Code for EPSAs**
different pattern searches. In these EPSAs, each mutation step is equivalent to the type of step defined by a simple pattern search algorithm. Consequently, a point \( p^*_i(x) \) can be described by a sequence of steps that define a pattern search.

Consequently, we can argue that the sequence of best iterates \( x^*_k \) converges. Let \( L_0(y) = \{ x \in \Omega \mid f(x) \leq f(y) \} \) and \( L(y) = L_{\mathbb{R}^n}(y) \). Theorem 1 describes the stationary point convergence theory for mutation-based EPSAs. A similar result can be obtained for the bound-constrained case.

**Theorem 1** Let \( L(x_0) \) be compact and suppose that \( f : \mathbb{R}^n \to \mathbb{R} \) is continuously differentiable on an open neighborhood of \( L(x_0) \). Then for the sequence of iterates \( \{ x^*_k \} \) produced by a mutation-based EPSA,

\[
P \left( \liminf_{k \to \infty} \| g(x^*_k) \| = 0 \right) = 1.
\]

### 6 Hybrid EPSAs

A particularly interesting class of hybrid EAs are those hybrids that apply local search to a subset of the points generated by the evolutionary operators in an EA in each iteration. The motivation for these hybrids is that they decompose the search by allowing the EA to globally sample starting points while using a local search method to quickly refine these points. Hart, Kammeyer, and Belew [3, 9] argue that these types of hybrid EAs are better global optimizers than either EAs or local search separately, and Tôm and Žilinskas [13] note that most successful global optimization methods also apply the same principle of distinguishing the mechanisms for global and local search.

Hybrid EAs using local search have been successfully applied to a range of applications. These techniques have been called memetic algorithms, genetic local search, hybrid genetic algorithms and genetic hillclimbing (see alife.ccl14.ac.uk/memetic/~moscato/memetic_home.html for an extensive bibliography of these hybrid evolutionary algorithms). Despite their success, basic principles have not been formulated to guide the development of effective hybrids. In particular, for continuous domains no convergence analysis has been developed for hybrid EAs using local search.

If gradient information is available, then applying a hybrid EA using local search can ensure stationary point convergence simply by terminating each local search at a stationary point. However, for direct search problems this convergence guarantee is more difficult to provide because the local search cannot guarantee that it terminates at (or near) a stationary point. To address this issue, we consider a slightly modified formulation of the EPSAs in Figure 3 that adds the following lines that perform local search:

\[
\begin{align*}
(14a) & \quad \text{For } i = 1 : N \\
(14b) & \quad \text{if } \text{unif}() < \rho \text{ then} \\
(14c) & \quad p^*_i = \text{localSearch}(p^*_i)
\end{align*}
\]
where $\rho$ is the probability of applying local search to a given point in a population.

If the method **local search** is implemented by a pattern search algorithm, then the step length that is returned in $p_k$ will reflect how much this method refined the initial point generated by the EPSA. Consequently, we can simply extend our analysis of the mutation-based EPSAs in Figure 3 to handle the local search steps. Instead of a point being the result of mutation steps, a point is the result of mutation and local search steps. But since the step length of mutation and local search serve the same role, this entire sequence can be viewed as the result of a single pattern search.

**Theorem 2** Let $L(x_0)$ be compact and suppose that $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is continuously differentiable on an open neighborhood of $L(x_0)$. Then for the sequence of iterates $\{x_k\}$ produced by a hybrid mutation-based EPSA that uses a pattern search for local search,

$$P \left( \lim_{k \rightarrow \infty} \inf \|g(x_k)\| = 0 \right) = 1.$$

7 Discussion

We have recently demonstrated that EPSAs can be successfully applied to challenging global optimization problems [8, 6]. However, we expect that the new EPSAs that we have described will ultimately provide superior empirical performance. Since different parts of the search domain have different scale properties, it is very important to allow the different members of an EAs population to adjust their search scales independently. Additionally, our prior work with hybrid EAs suggests that they are often much more efficient than simple EAs. Thus we expect that these hybrid EPSAs will be more effective than the EPSAs that we have empirically evaluated before.

References


On failure rates for Controlled Random Search

Eligius Hendrix
Group Operations Research and Logistics
Wageningen University
Dreijenlaan, 4
6803 HA Wageningen
Email: eligius.hendrix@oa.wk.wau.nl

P.M. Ortigosa and I. García
DSI
Univ. Almería
Cta Sacramento, SN
04120 Almería (Spain)
Email: inma@monet.ualm.es

Controlled Random Search (CRS) is an algorithm which despite its attractiveness for practical use, has never been very popular by researchers on Global Optimization due to the lack of possibilities to analyze it. In this paper some first results are given on deriving bounds on speed of convergence and the failure rate by comparing it to an ideal algorithm called $N$-points Pure Adaptive Search (NPAS). Extreme cases of functions are used to relate the performance to characteristics of functions to be optimized.

The Controlled Random Search algorithm, proposed by Price, is a simple and direct procedure for global optimization, applicable both to unconstrained and constrained optimization problems [9, 10, 11]. It has contributed to solving (and understanding) many practical problems resulting in many papers with applications on GO, e.g., Garcia et al., 1997, Klepper and Rouse, 1991. It is one of the first methods using a population of candidate points. Therefore, it has been popular in parameter estimation, because the cloud of final points tells something about the sizes of compartments around local optima in the final level set. It gives a kind of cover of the level set providing insight into the uncertainty of estimated parameters (see e.g., Klepper and Hendrix, 1994).

In this paper we are aiming at an intermediate between analysis and application. Test functions are used to analyze deeper, which measurable factors determine the performance of the algorithm. Some extreme cases are analyzed. Moreover, we intend to obtain more insight to deal with questions concerning a
variant of CRS, called PCRS, which includes buffering and was introduced by Ortigosa et al., 1998. The concept of buffering, keeping a part of the population in a buffer, is a generic concept for running population algorithms in parallel. The main question is to determine which factors in problems to be solved influence the speeding up or slowing down of the performance when buffer sizes increase. Moreover, by studying the relevant factors, some insight in test functions which are used for empirical studies in Global Optimization is generated.

A benchmark algorithm is called Pure Adaptive Search (PAS). PAS is not a real implementable algorithm, but a tool for analysis of complexity and in some sense an ideal. The analysis in literature focuses on the question what will happen if we would be able in every iteration to sample a point in the improving region, the level set \( S_k = S(f(x_k)) \) of the current iterate \( x_k \). The most important assumption is, that the next iterate \( x_{k+1} \) is a sample point from a uniform distribution over the interior of the current level set \( S_k \). The most important property, shown among others by Patel et al. (1988) and by Zabinsky and Smith (1992), is that in some sense the number of iterations grows less than exponential in the number of variables of the problem. In Hendrix and Klepper (1998) it has been shown why it is improbable that this ideal will be reached; i.e. it is unlikely that sampling a point from a uniform distribution over the current level set can be performed in a time which grows polynomially in the dimension \( n \) of the problem.

To make the results applicable for the analysis of population based algorithms, the extension of PAS to \( N \)-points PAS (NPAS) was introduced. NPAS implies performing PAS with a population of \( N \) points simultaneously, in which the current level set is defined by the worst point in the population. It was shown by Baritompa and Steel (1993) and Klepper and Hendrix (1994), that the complexity result is the same as that of PAS. NPAS is a hypothetical algorithm, an ideal, with a desirable complexity. CRS is a realistic, implementable algorithm which deviates on the following points from the ideal of NPAS:

1. A newly generated point is not a sample from a uniform distribution.

2. Not every new point is an improving point, i.e., has a better function value than the worst point in the population.

With respect to 2, it is useful now to introduce the notion of Failure Rate (or conversely Success Rate). There is not an exact definition; a measurable indicator can be defined in various ways. One can define the Failure Rate in a theoretical environment as the probability that the next generated point is an improving one, i.e., is in the interior of the current level set. This measure does not only depend on the algorithm, but also on the problem to be optimized, on the dynamics (current level set) and also on the current population (or point). To make it a measurable indicator, usually a definition like frequency with which one samples outside \( S_k \) is used. Either one can calculate at the end of the algorithm the number of points being evaluated without an improvement compared to the total number of function evaluations, or one can update an estimate (a statistic not clearly defined) to get insight in the development during the course.
of the algorithm. In particular in the CRS algorithm, a success rate indicator is measured and used explicitly to steer the algorithm in its choice between local and global search.

An important result with respect to the notion of Failure Rate (FR) is due to a paper by Baritompa et al. (1995) called Towards Pure Adaptive Search. In this paper it was shown that PAS can be relaxed by requiring that there is a fixed probability \( p \) (independent of the dimension) that the next iteration point is uniform in the improving region, called \( p \)-adaptive search. The same complexity bound holds as that of PAS. Again by reasoning reversely, we can conclude that it is unlikely that such a fixed probability exists. For CRS no attempts have been made to derive FR in an analytical way. In Hendrix and Klepper (1998), a derivation can be found for a population algorithm, similar to CRS called Uniform Covering by Probabilistic Rejection (UCPR or Raspberries for short). Essential in such a derivation is to make assumptions on the form of the level sets. This leads us to an important extreme case test function, the spherical problem.

For analysis with PAS, CRS or UCPR and estimation of the failure rate, any optimization problem with spherical level sets could be called a spherical problem. In Hendrix and Klepper (1998) the problem

\[
\min \{ f(x) = |x| \} \text{ on } X = \{ x \in \mathbb{R}^n : |x| \leq 1 \}
\]

(1)

was used, which easily can be transferred to the conical convex program which was used in Patel et al. (1988) to derive the PAS result. Essential in the derivation of results on efficiency is that the improvement (its probability distribution) \( y_{k+1}/y_k \) does not depend on the level \( y_k \) which has been reached. When considering the failure rate, one can note that the spherical problem is an extreme case in the sense that the surface/volume ratio is at a minimum. Therefore, the spherical problem gives a kind of lower bound on the failure rate, as for all other level sets the surface/volume ratio is at least as big. We will use some variants for the analysis. For such extreme cases, pure forms, it is easier to think of relations between the performance and factors which can be influenced such as the population size (cluster size) used by CRS and the buffer size used by PCRS.

References


A Global Optimization Method and its Application to the Sizing of Analog Functional Blocks

Jens Hichert
Institute of Mathematics
Technical University of Ilmenau, Germany
Email: Jens.Hichert@geomagic.pdv.de

Jürgen Kampe
Division of Microelectronic Circuits and Systems
Technical University of Ilmenau, Germany
Email: Juergen.Kampe@inf-technik.tu-ilmenau.de

In general, the high-level synthesis of analog circuits can be understood as a sequence of structural synthesis and sizing steps, starting from a functional behavioral description and leading to a more detailed design representation.

The first approaches to structural synthesis of analog blocks were done by architectural selection [1][2]. Even in connection with hierarchical decomposition [3][4][5], design automation tools could not deal with a large variety of specifications. Now, formalization leads to more flexible synthesis methodologies [7] and overcomes the bounds of rigid circuit classes by shifting the focus to signal processing with the help of more abstract design representations.

Besides simulation based methods [8][9][10], analytical [2] and symbolic [10][11] approaches were established for sizing the structural elements characteristic values.

In the way structural synthesis approaches are becoming more general, the sizing approaches have to be able to deal with a varying amount of specifications, using the allowed parameter space for optimization. On the other hand, there is no initial solution available, if the structural synthesis leads to a new topology of circuit primitives.

In accordance to the high-level design flow, all steps required for the design of the data flow belong to the analog functional synthesis and all steps necessary for the design of the energy flow are part of the electrical synthesis. This scheme of abstraction enables formal methods for putting up the system function as well as the constraints.

In this talk, a global optimization method is discussed for sizing analog functional blocks. We focus on the sizing step according to the analog functional
synthesis. Besides functional terms, the design model regards all kinds of error and fluctuation with respect to the signal parameters carrying information. The links between functional blocks are provided with a error model, enabling freedom for the electrical synthesis steps. The independent functional block parameters, their precisions, and the precisions of inner parameter sources are determined by the sizing step on this level of abstraction.

The global optimization method used was proposed in [12] and in a talk at the workshop in 1995 in Szeged [13]. The method follows the idea of Chichinadze [14], who considered level sets for determining the essential supremum of the objective function. In contrast to Chichinadze's approach, the present method is theoretical based on integrals over the level sets.

The method can be described as a Newton-like method to determine the root of a real valued convex nonnegative function.

As it is known from Newton's method, a linear rate of convergence of the levels can be expected. This linearity gets worse with increasing number of parameters.

The proposed method strongly depends on the way the integrals over level sets can be approximated. In our implementation a combined technique of branching (partitioning) and reducing of the search domain is applied. The partitioning technique comes from deterministic approaches, while the reducing technique is a typical stochastic approach. The combination of both leads to an efficient method that must not converge to the global solution, but has proved successful in solving problems from textbook as well as real-life problems.

References


Solving Sum-of-Ratios Fractional Programs
Using Efficient Points

Mirjam Dflr
Department of Statistics, Vienna University of Economics,
A-1090 Vienna, Austria
Reiner Horst and Nguyen Van Thoai
Department of Mathematics, University of Trier,
D-54286 Trier, Germany

Constrained maximization of a sum of \( p > 1 \) ratios is a difficult nonconvex optimization problem even if all functions involved are linear. The model arises in various economic as well as non-economic applications, whenever one or several rates are to be optimized. Numerators and denominators may represent e.g. profit, cost, capital, risk or time. There is a close relation to the associated multiple-objective optimization problem where a number of ratios are to be maximized simultaneously and the objective function of the Sum-of-Ratios Problem can be interpreted as a utility function expressing a compromise between the different objectives of the multiple-objective problem.

For \( p > 1 \), three approaches have been proposed previously: the method of Alnagy and Levi (1971) (which is incorrect), the approach of Cambini, Martein and Schaible (1980) (which relies on properties that hold only for \( p = 2 \)), and the algorithm of Falk and Palocsay (1992) (which hold theoretically for arbitrary numer \( p \) of ratios, but which is numerically practicable only for \( p = 2 \)).

In this talk we propose a new branch-and-bound algorithm which operates in space \( \mathbb{R}^p \), and converges for very general ratios and constraints. Then we specialize the approach to the affine fractional case. Upper bounds are obtained by solving the Lagrangian-Dual problem (which turns out to be equivalent to a linear program). Lower bounds are constructed via efficient points of the associated multiple-objective problem. Numerical experiments show that the algorithm is practical for at least four ratios.
Global Optimization Problems in the Chemical Process Industry

Josef Kallrath
BASF-AG, ZOI/ZC, C13, D-67056 Ludwigshafen,
e-mail: kallrath@zx.basf-ag.de

In the global economy, proper organization, design as well as planning of production and storage locations, transportation and scheduling are vital to retain the competitive edge of companies. The design, planning and scheduling problems involved are immensely complex. Mathematical optimization is an appropriate approach to solve such problems. The development of new algorithms in mathematical optimization, software and hardware provides tools which allow the solution of larger problems in acceptable times. Typical for the chemical industry are process design and process synthesis problems or many problems in which multi-component flows need to be blended leading to nonlinear or even mixed integer nonlinear problems. A problem, most frequently occuring is the pooling problem describing multi-component network flow situations in which not only the mass but also the composition of the multi-component streams is to be conserved. One objective of this contribution to create some awareness in the global optimization community for optimization problems dominated by nonlinear equality constraints.

The contribution covers optimization problems we solved using mixed integer nonlinear programming (MINLP) methods. The problems presented below, a production planning problem in BASF's petrochemical division, a tanker refinery scheduling problem at a refinery, a site analysis of one of BASF's bigger sites, and a process design problem have been solved within BASF's mathematical consultant group in Ludwigshafen (Germany).

The first problem presented leads to a MINLP model for describing a petrochemical network including several steam crackers and plants located at two different sites. The sites are interconnected by pipelines and ship transport. The sites consist of 2(1) steamcrackers, 3(2) downstream processing plants, recycles and 17(10) storage tanks. The model considers costs for transport, external purchases, raw material, utilities and inventory.

Blending leads to nonlinear structures suitable for recursion. The model includes about 10 recursed streams, which lead to about 65 recursed matrix coefficients for each period. Binary variables are needed to select cracker operation modes, to interpolate between them, or representing semi-continuous shipping
amounts. The yield coefficients in the steamcrackers, depending nonlinearly on temperature and pressure, are determined through interpolation.

The basic multi-period model for the larger site has six periods, includes distributive recursion with about 400 recurse coefficients and about 3500 variables; 30 of them are binary variables. The basic model is part of the mult-site model. The problem is solved using PIAMS (Module PPIMSSX and XPIMS) by AspenTech Corp. (Houston, US).

Great effort had to be made in connection with convergence of the distributive recursion. The algorithm tends to cycle in multi-period-computations, shifting around the property errors between the periods instead of eliminating them, especially when using more than six (shorter) periods. Finally, we succeeded in improving the performance, especially regarding the multi-sites model, speed-up convergence of distributive recursion and in modelling the interpolation of more than three cracker modes.

The aim of the second project, a *Tanker and Refinery Scheduling Problem* was to model and schedule the production and the storage of oil products of the medium size refinery for a period of two to four weeks. The data for supply of crude oil and for the delivery of the products were provided by a preceding production planning optimization.

Since not just a single crude oil type is stored in the tanks but different types of different properties are mixed, the mathematical problem becomes nonlinear. The necessity of deciding the day of production, and of which crude oil tank oil is changed to which production unit in order to get a particular product requires the application of binary variables.

To solve this MINLP problem two methods have been used:

- the software package GAMS (GAMS Inc., Washington) with the DICOPT-algorithm using a nonlinear solver (CONOPT) by ARKI Consulting & Development A/S, Denmark) in combination with a MILP-solver.
- the software package XPRESS-MP by Dash Associates (Bisworth, England) using its new B&B algorithm supporting recursion at each node.

The size of the total problem was as following:

- technically: 16 crude oils, 4 blending components, 17 crude oil and 8 intermediate tanks, 5 production units and 8 final products,
- mathematically: 80000 constraints, 70000 variables, 2000 binary variables, 1300000 nonzeros

The third problem, a network design problem with about 6000 variables, leads to a MINLP problem dominated by pooling problems and about 900 binary variables.
The model describes a network of process units within one or more production units connected by a system of pipes. Some of the process units manufacture substances, others produce substances which can be used in other units, others do both. For all units the demand or the amount of products required is fixed. For some of these substances which are already used within the system an expensive re-processing is necessary in order to get an optimal mixture and quality.

The actual situation is that nearly all raw material comes from external delivery points at high expenses. The idea is to make use of the products manufactured within the model system and to reduce the costs for raw material and for re-processing.

New plumblings may be constructed if the actual pipe system is not sufficient. In addition it might be possible that re-processing of substances or a mixture of substances in a small local re-processing unit is cheaper than getting them from outside. Therefore the construction of new re-processing units has to be decided.

One important aspect concerns the nonlinearities of the model. It becomes nonlinear because of mixing or blending substances where amount of solvent and concentrations are not known in advance (“pooling problem”) and only the resulting mixture has to fulfil some quality constraints. The decision on the construction of plumblings and re-processing units is modelled by binary variables. The constraints (about 6000) in the model describe the following features:

- mass balance equations for substances and solvents
- inequalities to fulfil quality constraints
- inequalities which can force the construction of plumblings or re-processing units
- objective function including all costs (raw material, re-processing, construction)

Solution Approach: The model is structured in several sub-models formulated in GAMS of which each is based upon the former, i.e., we use a homotopy method. Therefore there are purely nonlinear submodels including a very rough linear approximation which only provides initial values. Both are solved by the NLP-solver CONOPT. The MINLP problem is presolved by relaxing the binary variables which allows them to have any real value between 0 and 1. The main solution procedure afterwards is based upon the “Outer Approximation” (Viswanathan & Grossman, 1990) which is included in the solver program package DISOPT (Viswanathan & Grossman, Carnegie Mellon University). Solution times are of the order of one hour on a PC.

The fourth problem is concerned with a process design problem in which some process parameters and the optimal topology of a cascade of chemical reactors are computed w.r.t. optimizing total production, selectivity, energy, and costs.
Nonlinearities are related to the exponential terms for the reaction kinetics and mass fractions used to interpolate density and viscosity. Discrete features are required to model minimum flow rates between reactors, the number of reactors and their connections. The variables are the flow rates, and the number and size of reactors. The optimization model has been embedded into an attractive and easy to use user-interfaces. It helps the client in his daily production planning duties to adjust his plant immediately to current needs, i.e., changes in costs, capacities fluctuations or to attributes of orders. The tool supports the process design phase and helps to lay out cascades and connections of a system of reactors. The new designs save raw material, minimize waste material and increase the capacity of the reactor system. In the layout phase the tool support design and other changed constraints.

**Keywords:** production planning, scheduling, refinery scheduling, pooling problem, recursion, MINLP, Branch&Bound, Outer Approximation, network design problems, process design problem
A Multistart-Based Global Minimization Method by Removing Sample Points Converging on Already Found Local Minima

Hideo Kanemitsu, Hideaki Konno, Yuriko Komiya, and Masaru Shimbo

†Hokkaido University of Education, Hakodate 040-8567, Japan
1) hkanemt@hak.hokkyodai.ac.jp
2) konno@hak.hokkyodai.ac.jp
†Hokkaido University, Sapporo 060-8628, Japan
3) komiya@hipecs.hokudai.ac.jp
4) shimbo@main.eng.hokudai.ac.jp

We propose a new multistart-based method for finding a global minimum of multimodal functions. This method constructs a hyper-rectangle region that approximates the unimodal region at a local minimum using eigenvectors of a Hessian matrix. Since the unimodal region is defined as a maximum region that includes only one local minima, all sample points in the unimodal region converges on the same local minimum. From this fact, all sample points that are included in or monotonically decrease to one of hyper-rectangles are expected to converge on one of already known local minima. By removing such a point before applying a local minimizer, the method can effectively finds local minima.

8 Introduction and Preliminaries

We deal with a minimization problem (P):

minimize $f(x) \equiv f(x_1, x_2, \ldots, x_n)$, $x \in D^\alpha \equiv \prod_{j=1}^{m_0} \{a_j, b_j\}$. (P)

Suppose the function $f(x) \in C^2$ has a finite number of isolated local minima $x_k^* \in D^\alpha$ $(k = 1, 2, \ldots, M)$. A set $X^*$ of isolated local minima and a set $F^*$ of
their minimal values are written as

\[ X^* = \{ x_1^*, x_2^*, \ldots, x_M^* \}, \quad F^* = \{ f(x_1^*), f(x_2^*), \ldots, f(x_M^*) \}. \tag{1} \]

To solve this problem, our proposed method uses a hyper-rectangle that approximately covers a unimodal region at a local minimum. This method finds as many local minima as possible without duplication by removing points in hyper-rectangles, denoted by |A|.

**Notation 1** Let a small level set \( L^e(\alpha) \) be

\[ L^e(\alpha) = \{ x \mid f(x) < \alpha \}. \tag{2} \]

A connected component of \( L^e(\alpha) \) including \( x \) is denoted by \( L^e(\alpha; x) \), and the number of connected component of a set \( S \) is denoted by \(|S|_c\).

**Notation 2** A unimodal-region \( R_0(x^*) \) of local minimum \( x^* \), that is defined as a maximum open connected level set including only one local minimum \( x^* \), is denoted by

\[
\left\{ \begin{array}{l}
R_0(x^*) = \{ x \mid x \in L^e(\alpha_u(x^*); x^*) \}, \\
\alpha_u(x^*) = \max_{\alpha} \{ |L^e(\alpha_u) \cap L^e(\alpha_s; x^*)|; \forall \alpha_s \in [f(x^*), \alpha] \},
\end{array} \right\} \tag{3}
\]

**Notation 3** A hyper-rectangle of local minimum \( x^* \) is denoted by

\[ H(x^*) = \prod_{j=1, \ldots, n} [x^*_j - \delta^*_j v_j, x^*_j + \delta^*_j v_j], \tag{4} \]

where \( v_j, (j = 1, \ldots, n) \) are the normalized eigenvectors of the Hessian matrix \( \nabla^2 f(x^*) \in \mathbb{R}^{n \times n} \), and \( \delta^*_j > 0(i = 1, 2; j = 1, 2, \ldots, n) \) is the step length along each eigenvector \( v_j \) from the origin \( x^* \).

A set \( H \) that consists of \( M \) hyper-rectangles and a set \( V \) of eigenvectors \( v_j \) are written as

\[ H = \{ H(x_1^*), H(x_2^*), \ldots, H(x_M^*) \}, \quad V = \{ v_1, v_2, \ldots, v_n \}. \tag{5} \]

**9 Main algorithm**

One purpose of the algorithm is to remove, in advance, points that are included in or expected to converge on the region of a hyper-rectangle an already known local minimum. If a hyper-rectangle can be constructed such that it approximates the unimodal region of a local minimum, then almost all points in the hyper-rectangles can converge on the same local minimum.

The main algorithm finds the set of local minima \( X^* \) and the set of minimal function values \( F^* \) for a function \( f \) over a searching region \( D^p \), for a given limited number of sampling \( N \) and for a given step length \( h \). The steps of the algorithm are as follows.
\[ B \quad (F^*, X^*) \leftarrow \text{gmin}(D^3, f, N_h) ; \]

M1. [Initialize]
\[ X^* \leftarrow \emptyset ; \quad F^* \leftarrow \emptyset ; \quad X_c \leftarrow \emptyset ; \quad N_s \leftarrow 0 ; \quad \text{# X_c is the set of candidates} \]

M2. [Take a sample point, and check whether the sample point is contained in one of the hyper-rectangles]
\[ x_s \leftarrow \text{random}(D^3) ; \quad N_s \leftarrow N_s + 1 ; \quad \text{if } N_s > N_s \text{ then terminate} ; \]
\[ \text{if } x_s \in H(x_j^*) \text{, } (x_j^*) \in X^*, j = 1, 2, \ldots , |X^*| \text{ then go to M2} ; \]

M3. [Concentrate the sample point around a local minimum and check whether the point is included in one of the hyper-rectangles]
\[ f(x_s) \leftarrow \min \{ f(x_s + \lambda (\nabla f(x_s)) \} ; \]
\[ x_s \leftarrow \arg \min \{ f(x_s + \lambda (\nabla f(x_s)) \} ; \]
\[ \text{if } x_s \in H(x_j^*) \text{, } (x_j^*) \in X^*, j = 1, 2, \ldots , |X^*| \text{ then go to M2} ; \]
\[ X_c \leftarrow \{ x_s \} ; \]

M4. [Check whether each candidate can monotonically decrease to one of the hyper-rectangles]
\[ \text{for all } x_j \in X_c \text{ do} \]
\[ \text{for all } x_j^* \in X^* \text{ such that } -\nabla f(x_j)^T (x_j^* - x_j) \geq 0 \text{ do} \]
\[ e_{jk} \leftarrow \frac{(x_j^* - x_j)}{\| x_j^* - x_j \|} ; \quad \text{imin} \leftarrow \text{imin} \left( \frac{\| x_j^* - x_j \|}{h} \right) ; \quad f(0) \leftarrow f(x_j) ; \]
\[ \text{for } i \leftarrow 1 \text{ to max do} \]
\[ f(i) \leftarrow f(x_j + i \cdot e_{jk}) ; \quad \text{if } f(i - 1) \leq f(i) \text{ then break} ; \]
\[ \text{if } x_j + i \cdot e_{jk} \in H(x_j^*) \text{ then } X_c \leftarrow X_c - \{ x_j \} ; \]
\[ \text{go to nextp} ; \quad \text{fi} ; \]
\[ \text{od} ; \quad \text{od} ; \quad \text{nextp} ; \]
\[ \text{if } X_c = \emptyset \text{ go to G2} ; \]

M5. [Find a local minimum \( x^* \)]
\[ x_s \leftarrow \arg \min \{ f(x_s) \mid x_s \in X_c \} ; \quad f_s \leftarrow f(x_s) ; \]
\[ (F^*, x^*, V) \leftarrow \text{tomin}(f_s, x_s) ; \quad X_c \leftarrow X_c - \{ x_s \} ; \quad \text{# tomin is local minimizer} \]

M6. [Make a new hyper-rectangle]
\[ (H, X_c) \leftarrow \text{make hyper}(H, X_c, f^*, x^*, V, h) ; \quad X^* \leftarrow X^* + \{ x^* \} ; \]
\[ F^* \leftarrow F^* + \{ f^* \} ; \]
\[ \text{if } X_c = \emptyset \text{ go to M2} \text{; else go to M5} ; \]

---

10 Algorithm for making a hyper-rectangle

Step M6 of the main algorithm makes a new hyper-rectangle of the local minimum so that the unimodal region of the local minimum is approximately covered with the hyper-rectangle \( H(x^*) \) as follows:

\[ H(x^*) = \prod_{j=1}^{m} [x^* - \delta_j] v_j, x^* + \delta_j v_j ] \approx R_u(x^*) , \]

72
The hyper-rectangle \( H(\mathbf{x}^\ast) \) can be constructed by calculating the step length \( \delta^j \). To calculate \( \delta^j \) along vector \( \mathbf{v}_j \) from the origin \( \mathbf{x}^\ast \), we define a function \( \phi(\delta) \equiv f(\mathbf{x}^\ast + \delta \mathbf{v}_j) \) and estimate the following values.

1. Firstly, the algorithm evaluates the function value \( f^{(k)} \) at the point \( \mathbf{x} = \mathbf{x}^\ast + (-1)^i \mathbf{k} h \mathbf{v}_j \) with the same interval step \( h > 0 \) as follows:

\[
\begin{align*}
    \frac{f^{(k)}}{k!} &= f(\mathbf{x}^\ast + (-1)^i \mathbf{k} h \mathbf{v}_j), \quad (k = 1, 2, \ldots, K^j) \\
    K^j &= \max\{ k \mid x^\ast + (-1)^i \mathbf{k} h \mathbf{v}_j \in D \} + 1.
\end{align*}
\]

(6)

2. Secondly, second derivatives \( d^{(k)}, \ k = 1, 2, \ldots \) of \( \phi(\alpha) \) by interpolation of the second-order function \( \phi(\delta) = \sum_{i=0}^{2} p_i \delta^i = p_2 \delta^2 + p_1 \delta + p_0 \), are estimated. Let \( \delta^k_i \equiv (-1)^i \mathbf{k} h \) and \( f^{(k)} \equiv \phi(\delta^{(k)}_i) \), then for \( k \geq 2 \) the following equations hold.

\[
\phi(\delta^{(k-m)}_i) = \sum_{i=0}^{2} p_i \delta^{(k-m)}_i f^{(k-m)}, \quad (0 \leq m \leq 2)
\]

(7)

By solving the equations and assuming \( \phi^{(k)} = 2p_2 \), we obtain

\[
\delta^{(k)} = \frac{f^{(k-2)} + f^{(k)} - 2f^{(k-1)}}{h^2}, \quad k \geq 2.
\]

(8)

Moreover, from \( \phi^{(1)}(\mathbf{h}) = 0 \) at the minimum or maximum point, \( \mathbf{h} \) is estimated by

\[
\mathbf{h} = (k-1)\mathbf{h} - \frac{f^{(k)} - f^{(k-2)}}{2h\delta^{(k)}}.
\]

(9)

Similarly, from \( \phi^{(2)}(\alpha^{(0)}) = 0 \), we can estimate the second derivative \( \delta^{(1)} \) as follows:

\[
\delta^{(1)} = \frac{2f^{(1)} - f^{(0)}}{h^2}.
\]

(10)

3. Finally, we estimate the point of inflection \( \mathbf{h} \) by interpolation of the third-order function \( \phi(\delta) = \sum_{i=0}^{3} p_i \delta^i = p_3 \delta^3 + p_2 \delta^2 + p_1 \delta + p_0 \). Let \( \delta^k_i \equiv (-1)^i \mathbf{k} h \) and \( f^{(k)} \equiv \phi(\delta^{(k)}_i) \), then for \( k \geq 3 \) the following equations hold.

\[
\phi(\delta^{(k-m)}_i) = \sum_{i=0}^{3} p_i \delta^{(k-m)}_i f^{(k-m)}, \quad (0 \leq m \leq 3)
\]

(11)

By solving the equations and noting that \( \phi^{(3)}(\mathbf{h}) = 0 \) holds at the point of inflection, we can estimate \( \mathbf{h} \) for \( k \geq 3 \) as follows:

\[
\mathbf{h} = (k-2)\mathbf{h} + \frac{\mathbf{h}(2f^{(k-2)} - f^{(k-3)} - f^{(k-1)})}{f^{(k)} + 3f^{(k-2)} - 3f^{(k-1)} - f^{(k-3)}}.
\]

(12)

Similarly, for \( k = 2 \), from equation (11) where \( k = 2, k = 1, k \) and \( \phi^{(2)}(\alpha^{(0)}) = 0 \), we have

\[
\mathbf{h} = (k-1)\mathbf{h} - \frac{2h\delta^{(k)}}{3(3f^{(k-2)} + f^{(k)} - f^{(k-1)})}.
\]

(13)

Using the above estimations, the step length \( \delta^j \) is determined as follows.
\[ B \quad (\delta_j, \overline{x}_c, \overline{v}_c) \leftarrow sptl(h, k_j, v_j, x^{(k)}, \overline{x}_c, \overline{v}_c); \]

S1. \[ f^{(0)} \leftarrow f(x^*); \quad f^{(1)} \leftarrow f(x^* + (-1)^i kh v_j); \quad \delta^{(1)} = 2(f^{(1)} - f^{(0)})/h^2; \]
if \( \delta^{(1)} \leq 0 \) then \( \delta_j \leftarrow h; \) go to end; \[ \text{fi}; \]

S2. \[ iflg \leftarrow 0; \quad mflg \leftarrow 0; \quad \delta_j \leftarrow h k_j; \]
for \( k \leftarrow 2 \) to \( k_j \) do
\[ x^{(k)} \leftarrow x^* + (-1)^i kh v_j; \quad f^{(k)} \leftarrow f(x^{(k)}); \quad \delta^{(k)} \leftarrow (f^{(k-2)} + f^{(k)} - 2f^{(k-1)})/h^2; \]
S3. \[ if \delta^{(k-1)} \delta^{(k)} < 0 \] then
Estimate the step length \( h \) of a point of inflection by eq.(12) or eq.(13),
\[ iflg \leftarrow 1; \quad \text{fi}; \]
S4. \[ if \delta^{(k)} < 0 \]
Estimate the step length \( h \) of with maximal value by eq.(9),
\[ if \ iflg > \text{ifflg} \ then \ break; \ else \ iflg \leftarrow iflg + 1; \quad \text{fi}; \ \{ \ast \ iflg \ is \ threshold \ of \ ifflg \ \# \} \]
S5. \[ if f^{(l-1)} < f^{(l-2)} \ and \ f^{(l+1)} > f^{(l)} \]
Add the point \( x^{(k)} \) and the vector \( v_j \) to the set of temporary candidates \( X_c \) and vectors \( V_c \), because the descent point after hill climbing is expected to converge on the other local minimum,
\[ X_c \leftarrow X_c + \{x^{(k)}\}; \quad V_c \leftarrow V_c + \{v_j\}; \quad \delta_j \leftarrow h; \quad \tilde{f} \leftarrow f(x^* + (-1)^i hv_j); \]
\[ mflg \leftarrow 1; \quad \text{break}; \quad \text{fi}; \]
\[ od; \ \{ \ast \ End \ of \ k-loop \ \# \} \]
\[ if \ mflg = 0 \ then \ \delta_j \leftarrow h; \ else \ \delta_j \leftarrow h; \ quad \text{end}; \]

Using the above step length \( \delta_j \), the following algorithm constructs the new hyper-rectangle \( H(x^*) \), and updates the sets of hyper-rectangles \( H \) and candidates \( X_c \).

\[ B \quad (H, X_c) \leftarrow \text{make-rect}(H, X_c, f^*, x^*, V, h); \]
B1. [ Initialize ]
\[ \overline{x}_c \leftarrow \emptyset; \quad \overline{v}_c \leftarrow \emptyset; \]
B2. [ Calculate the step length \( \delta_j \) and construct the new hyper-rectangle ]
for \( j \leftarrow 1 \) to \( n \) do
\[ v_j \leftarrow v_j; \quad \overline{k}_j \leftarrow \max \{ k \mid x^* + (-1)^i hv_j \in D^n \} + 1; \]
for \( i \leftarrow 1 \) to \( 2 \) do
\[ (\delta_j, X_c, \overline{v}_c) \leftarrow \text{stpl}(h, k_j, v_j, x^{(k)}, \overline{x}_c, \overline{v}_c); \]
\[ od; \ od; \]
\[ H(x^*) \leftarrow \prod_{j=1,...,n} [x^* - \delta_j v_j; x^* + \delta_j^2 hv_j]; \]

74
B3. [ Check whether the candidates are included in the new hyper-rectangle $H(x^*)$ ]
   for all $x_j \in X_c$ do if $x_j \in H(x^*)$ then $X_c \leftarrow X_c - \{x_j\}$; od;

B4. [ Check whether the temporary candidates are included in the previous hyper-rectangles ]
   for all $x_j \in X_c$ do
     for all $x_j' \in H$ do
       if $x_j' \in H(x_j)$ then $X_c \leftarrow X_c - \{x_j\}$; $V_c \leftarrow V_c - \{x_j\}$; od; od;

B5. [ Update the sets of hyper-rectangles and candidates ]
   $H \leftarrow H + \{H(x^*)\}$;
   for all $x_j \in X_c$ and $x_j' \in V_c$ do
     $x_j \leftarrow \text{arg min}_{x_j} \{f(x_j + \lambda x_j')\}$ asmc $X_c \leftarrow X_c + \{x_j\}$;
   od;

11 Numerical experiments

We tested the method for standard test functions and the results of our experiments are shown in Table 1. These results show that our method can be used to effectively find global minimum with high reliability in only a few function evaluations.

Table 1: Results for functions of $n$-variables with $M$-local minima ($N_0$, $M$, and $T[sec]$ are average of ten trials in each function).

<table>
<thead>
<tr>
<th>Function name</th>
<th>$n$</th>
<th>$M$</th>
<th>interval</th>
<th>$N_0$</th>
<th>$N_f$</th>
<th>$T$</th>
<th>$M/N_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Branin</td>
<td>2</td>
<td>2</td>
<td>$[0,10] \times [0,15]$</td>
<td>0.78</td>
<td>252</td>
<td>0.32</td>
<td>3.8</td>
</tr>
<tr>
<td>Branin-60</td>
<td>2</td>
<td>2</td>
<td>$[0,10] \times [0,15]$</td>
<td>0.83</td>
<td>250</td>
<td>0.29</td>
<td>1.9</td>
</tr>
<tr>
<td>Cof namely &amp; Price</td>
<td>2</td>
<td>3</td>
<td>$[-1,2]_n$</td>
<td>0.20</td>
<td>245</td>
<td>0.048</td>
<td>4.8</td>
</tr>
<tr>
<td>Shekel-2</td>
<td>4</td>
<td>2</td>
<td>$[0,10]^n$</td>
<td>0.60</td>
<td>132</td>
<td>0.116</td>
<td>1.4</td>
</tr>
<tr>
<td>Shekel-7</td>
<td>4</td>
<td>2</td>
<td>$[0,10]^n$</td>
<td>0.60</td>
<td>132</td>
<td>0.116</td>
<td>1.4</td>
</tr>
<tr>
<td>Shekel-19</td>
<td>4</td>
<td>10</td>
<td>$[0,10]^n$</td>
<td>0.50</td>
<td>132</td>
<td>0.25</td>
<td>5.6</td>
</tr>
<tr>
<td>Shekel-29</td>
<td>6</td>
<td>3</td>
<td>$[0,10]^n$</td>
<td>0.60</td>
<td>1246</td>
<td>0.228</td>
<td>10.5</td>
</tr>
</tbody>
</table>

$N_0$ is number of function evaluations
$T$ is execution time [sec]
$M$ is number of local minima found
$N_f$ is the number of duplicated found minima
$N_0$ is the number of performed local minimizer.

12 Concluding Remarks

This method has characteristic step that construct a hyper-rectangle for approximately covering a unimodal region. By the use of the hyper-rectangle, the method can remove sampling points in advance that are included in or linearly monotonically decrease to one of already known local minima. By numerical
experiments, we showed that the method can find global minimum and almost all local minima in a few function evaluations.
Objective function decomposition in global optimization

Q. Khamisov
Institute of Energy Systems, Irkutsk, Russia

Introduction

In this paper we consider global optimization problems in which objective functions are explicitly given and can be represented as compositions of some other functions. Many practical problems can be formulated in a such form, see for example [9], [8] [10], [6], [7]. Approaches similar to one described below were suggested in [1] and [5]. In [3] an equivalent approach was used for utility objective function, i.e. for the case when objective composite function has some monotonicity properties.

1. Objective decomposition and induced constraint

Consider the following mathematical programming problem.

\[ \min g(x), \quad (1.1) \]
\[ x \in R, \quad (1.2) \]

where \( g(x) \) is a continuous composite function

\[ g(x) = F(f_1(x), \ldots, f_p(x)), \quad (1.3) \]

\( F : E^n \to E^1, f_i : R \to E^1 \) are continuous functions, \( R \) is a compact subset of the \( n \)-dimensional Euclidean space \( E^n \).

Introducing new variables \( y_i \)

\[ y_i = f_i(x), \quad i = 1, p \]

we formulate the following equivalent to (1.1)-(1.2) problem

\[ \min F(y), \quad (1.4) \]
\[ y_i = f_i(x), \quad x \in R, i = 1, p. \quad (1.5) \]

In the latter problem minimization is performed in \( y \) variables. Assume that \( p < n \) or function \( F(y) \) is less complicated than \( f(x) \). In such case we can obtain a reduction in difficulty of the initial problem providing that equations
(1.5) are practically tractable, that means, for example, a possibility to solve equations (1.5) in \( x \) for a given \( y \) subject to the inclusion \( x \in R \) by an efficient computer code. Solving problem (1.1)-(1.2) corresponds to finding \textit{optimal} and \textit{feasible} point simultaneously. In problem (1.4)-(1.5) optimality (i.e. minimization in (1.4)) and feasibility (i.e. determining \( x \) for a given \( y \) in (1.5)) stages are separated: they are performed in different spaces. What is exactly done in the reduction of problem (1.1)-(1.2) to problem (1.4)-(1.5) and what is understood under objective function decomposition in this paper is deleting some complexity from the objective function to the constraints, i.e. moving a part of difficulty from the optimality stage to the feasibility stage. The motivation of such decomposition is to distribute difficulty of the initial problem between objective and constraints more or less uniformly.

In practical minimization of \( F(y) \) in (1.4) it is quite often necessary to localize a global minimum in some compact subset of \( E^n \) (see [4]). Define numbers \( \underline{L}_i, \overline{F}_i \)

\[
\underline{L}_i = \min_{x \in R} f_i(x), \quad \overline{F}_i = \max_{x \in R} f_i(x), \quad i = 1, \ldots, p, \quad (1.6)
\]

and consider the following problem

\[
\min F(y), \quad (1.7)
\]

\[
y_k \leq y_i \leq \overline{y}_i, \quad i = 1, \ldots, p, \quad (1.8)
\]

\[
y_i = f_i(x), \quad x \in R, \quad i = 1, \ldots, p, \quad (1.9)
\]

where

\[
y_k \leq \underline{L}_i, \quad \overline{F}_i \geq \overline{y}_i, \quad i = 1, \ldots, p.
\]

Let us define the set \( Y \subseteq E^n \) as an image of the set \( R \subseteq E^n \) under nonlinear continuous mapping (or transformation) \( f : R \rightarrow Y; f(x) = (f_1(x), \ldots, f_p(x)) \)

\[
Y = \{ y : y_k = f(x) \text{ for some } x \in R \}. \quad (1.10)
\]

Then problem (1.7)-(1.9) can be rewritten as follows

\[
\min F(y), \quad (1.11)
\]

\[
y_k \leq y_i \leq \overline{y}_i, \quad i = 1, \ldots, p, \quad (1.12)
\]

\[
y \in Y. \quad (1.13)
\]

Since (1.13) is a reformulation of the feasibility stage constraint (1.9) the inclusion \( y \in Y \) will be called an \textit{induced constraint}.

2. Using concordant variable decomposition

Assume that the composite objective function \( g(x) \) has concordant variable decomposition, i.e.

\[
g(x) = F(f_1(x^1), \ldots, f_p(x^p)), \quad (2.1)
\]
where \( x^i \in R^p \cup E^n, i = \overline{1, p}, R_1 \times \cdots \times R_p = R \) and \( n_1 + \cdots + n_p = n \). Conversely, we will say that the function \( g(x) \) has nonconcordant variable decomposition if \( g(x) \) is still representable in the form \((2.1)\) and \( x^i \in E^n, n_i < n, i = \overline{1, p} \), but \( n_1 + \cdots + n_p > n \).

In this case we rewrite problem \((1.7)-(1.8)\) in the following way

\[
\min F(y), \quad (2.2)
\]

\[
y_k \leq y_k \leq \overline{y}_k, i = \overline{1, p}, \quad (2.3)
\]

\[
y_k = f_i(x^i), x^i \in R^p, i = \overline{1, p}, \quad (2.4)
\]

Note, that in \((2.3)\) we use exact lower and upper bounds on \( y_k \) since otherwise, i.e. if constraints \((1.8)\) are used with \( y_k < \underline{y}_k \) or \( \overline{y}_k > \overline{y}_k \) for some \( i \) instead of \((2.3)\) in the latter problem the inclusion \( x^i \in R^p \) can be violated.

Due to the variable decomposition property \((2.4)\) we can use the following three-stage approach.

I. Variable bounding stage. Determine numbers

\[
\underline{f}_i = \min_{x^i \in R^p} f_i(x^i), \overline{f}_i = \max_{x^i \in R^p} f_i(x^i), i = \overline{1, p}.
\]

II. Optimal solution stage. Solve the problem \((2.2)-(2.3)\) and determine an optimal solution \( y^* \).

III. Feasibility stage. Solve feasibility problem \((2.4)\) for \( y = y^* \) and obtain an optimal solution \( x^p \) for the initial problem. If \( F(y) = \sum_{i=1}^{n} y_i \) then we have the well-known separable problem.

3. A reduction to d.c. program

We start this section by the definition of d.c. function from [4].

**Definition 1** Let \( C \subset E^n \) be convex. A function \( h : C \rightarrow E^1 \) is called d.c. on \( C \) if there are two convex functions \( p : C \rightarrow E^1, q : C \rightarrow E^1 \) such that

\[ h(x) = p(x) - q(x), \forall x \in C. \]

A function that is d.c. on \( E^n \) will be called d.c.

In this section we assume that \( F(y) \) is the d.c. function.

Let us introduce into consideration function \( \Phi(y) \)

\[
\Phi(y) = \min_{x \in R^p} \sum_{i=1}^{p} \left[ y_i - f_i(x)^2 \right]. \quad (3.1)
\]

**Proposition 3.1** Let \( f_i(x), i = \overline{1, p} \) be continuous functions. Then \( \Phi(y) \) is the d.c. function.
Proof Since

\[ \Phi(y) = \min_{x \in R} \left\{ \sum_{i=1}^{p} [y_i - f_i(x)]^2 \right\} = \min_{x \in R} \left\{ \sum_{i=1}^{p} \left[ f_i^2(x) - 2f_i(x)y_i + y_i^2 \right] \right\} = \]

\[ \min_{x \in R} \left\{ \sum_{i=1}^{p} f_i^2(x) - 2f_i(x)y_i \right\} + \sum_{i=1}^{p} y_i^2 = \Phi_1(y) + \Phi_2(y) \]

where

\[ \Phi_1(y) = \min_{x \in R} \left\{ \sum_{i=1}^{p} f_i^2(x) - 2f_i(x)y_i \right\}, \]

\[ \Phi_2(y) = \sum_{i=1}^{p} y_i^2, \]

\( \Phi_2(y) \) is obviously convex and \( \Phi_1(y) \) is concave as a pointwise minimum of a family of linear functions.

Hence, we can formulate an equivalent to (1.7)-(1.9) d.c. problem.

\[ \min F(y), \] \hspace{1cm} (3.2) \]

\[ \Phi(y) \leq 0, \] \hspace{1cm} (3.3) \]

\[ y_i \leq y_i \leq \mathcal{J}_i, i = 1, p. \] \hspace{1cm} (3.4) \]

Hence, for solving problem (3.2)-(3.4) one can use different d.c. minimization methods described in [4]. The particular type of the function \( \Phi(y) \) in (3.1) allows one to use the regression theory and methods (see, for example [2]). In the report we consider different kinds of functions \( f_i(x) \) and \( F(y) \). Numerical examples are presented.

References


80


Global Optimization on Funneling Landscapes

Robert H. Leary
San Diego Supercomputer Center
University of California, San Diego
P. O. Box 85608
San Diego, CA 92186-9784
Email: leary@sdsc.edu

Many optimization problems arising in computational chemistry and biology require the global minimization of a non-convex potential energy function representing the interactions of, for example, the atoms in a molecular system. The potential energy function may have an enormous number of local minima, each of which corresponds to a metastable conformation which is mechanically stable with respect to small perturbations. It is of particular interest to determine the global minimum, or native state conformation, as this is believed to be of fundamental importance in determining the function of a biomolecule such as a protein.

In this paper we will formalize the concept of a landscape associated with the global minimization of a potential energy function. For our purposes the landscape consists of the (finite) collection of local minima, an adjacency rule which designates certain pairs of minima as being neighbors, and a set of pathways connecting adjacent local minima. Thus an underlying directed landscape graph can be defined with nodes corresponding to local minima and arcs to pathways between adjacent local minima, with the arc direction from the higher to lower local minimum.

We will be primarily concerned with problems in which the complexity of the energy landscape, as measured by the number of local minima, is so high as to render ineffectual techniques that randomly sample local minima in the search space. The well-known Levinthal paradox is essentially this observation applied to the protein folding problem; the biological time scale of protein folding is much too short to be reconciled with the high landscape complexity under a simple sampling model using physically reasonable sampling frequencies. However, the "paradox" is easily resolved if the folding process is biased toward following descending trajectories in the landscape graph, provided such trajectories tend to converge to the global minimum. This leads naturally to the concept of a "funneling" landscape in which all or nearly all local minima are connected to the global minimum via an energetically descending chain of
adjacent local minima in the landscape graph. Indeed the hypothesis that protein folding is guided by a funnelling landscape is fundamental to our current understanding of that process.

On a strictly funnelling landscape, a simple "quenching" search strategy based on following any descending path through the landscape graph will terminate at the global minimum. In particular, such quenching searches include random walks from arbitrary starting nodes on the landscape graph which respect the directionality of the arcs. More generally, on landscapes dominated by a relatively few strong funnels, a quenching random walk with a sufficiently large number of restarts will locate the bottom of each of the most important funnels and hence the global minimum, if it indeed lies at the bottom of such a funnel. In this latter multiple funnel case, a Metropolis search at a suitable temperature over the landscape graph is also a viable strategy. Indeed, the quenching random walk is simply the low temperature limit of the Metropolis search.

In this paper we will elucidate these concepts for a well-known global optimization problem, namely that of minimizing the potential energy of clusters of particles interacting via a Lennard-Jones pair potential. Using an adjacency relation based on the presence of a first order saddle point connecting two local minima, essentially complete landscape graphs are computed for small clusters. Such computations for clusters with up to fourteen particles result in landscape graphs with up to several thousand nodes and several tens of thousands of arcs. These graphs demonstrate that the energy landscapes in this size range exhibit strong funnelling properties and are dominated by single primary funnel with at most a few secondary funnels. For these small clusters the global minimum lies at the bottom of the primary funnel.

These results can be extended to larger clusters with up to 100 particles using a computationally simplified adjacency relation based on geometrical proximity of the local minima in the particle coordinate space. Results of application of the quenching random walk algorithm with multiple restarts demonstrate the continued validity of the multifunnel energy landscape characterization for these larger clusters, although the number of funnels appears to grow slowly with cluster size, and the global minimum may lie at the bottom of a secondary funnel. The results also demonstrate the power of that algorithm to exploit the multifunnel landscape structure. In fact, we believe the algorithm is the most effective to date for the Lennard-Jones cluster problem and is the only known unbiased algorithm to have successfully obtained the unusual and apparently globally optimal decahedral configurations for the 76- and 77-particle cases, as well as all other known global optima in this size range. Comparisons are made with the related Metropolis landscape search algorithm, which is also quite effective on such landscapes, although with considerably different behavior.
A Global Minimization Algorithm for Lipschitz Functions

Marco Gaviano
Mathematics Department
University of Cagliari
via Ospedale, 72
09124 Cagliari
Email: gaviano@unica.it

Daniela Lera
Mathematics Department
University of Cagliari
via Ospedale, 72
09124 Cagliari
Email: lera@unica.it

The global optimization problem we are concerned with is

\[ \text{find } x^* \text{ such that } f(x^*) \leq f(x) \quad \forall x \in S \]  \hspace{1cm} (1)

with \( S = \{ x \in \mathbb{R}^n | a \leq x \leq b \} \) \hspace{1cm} a, b \in \mathbb{R}^n.

The function \( f(x) \) is assumed to be continuous and lipschitzian with respect to the maximum norm, i.e. there exists \( l > 0 \) such that

\[ |f(x) - f(y)| \leq l |x - y|_\infty \quad \forall x, y \in S. \]  \hspace{1cm} (2)

To solve (1) we introduce an algorithm which combines a local minimum search with a procedure that decreases the measure of the region where the global minimum is located.

Its global strategy is similar to that of an algorithm proposed by Luc P. Devroye [Progressive global random search of continuous functions, Mathematical Programming (1978)], but the specific techniques that are involved are different.

At each iteration \( i \) of the algorithm we construct a sequence of intervals \( s_j^{(i)} \), \( s_j^{(i)} \subset S \), with the property that the \( s_j^{(i)} \) do not overlap, and \( S_i = \bigcup s_j^{(i)} \) does not contain a global minimum. Specifically, at each iteration \( i \) we choose a point \( x_i \) in the set \( S - S_i \) (starting choice \( S_0 = \emptyset \)). Then, according to the value of \( f(\cdot) \) at \( x_i \) we choose whether or not to carry out a new local minimization: let \( f(y_i) \) be the value of the last local minimum found (starting choice \( f(y_0), y_0 \) an arbitrary point in \( S \)). Then if \( f(x_i) \) is greater than \( f(y_i) \), \( y_{i+1} \) is taken equal to \( y_i \), else a local search is carried out from \( x_i \) in order to find a new local minimum \( y_{i+1} \). The values \( f(y_{i+1}) \) and \( f(x_i) \) in conjunction with condition (2) are used to construct a new set \( S_{i+1} \). More specifically, if no local search is carried out, that is \( y_{i+1} = y_i \), a set \( Q \) containing \( x_i \), with the property that it does
not contain a global minimum, is added to \( S_i \) in order to get \( S_{i+1} \). In the case of \( f(x_i) \leq f(y_i) \), first the intervals of which \( S_i \) are made up are enlarged and arranged as to take into account the new value \( f(y_{i+1}) \). Then \( S_{i+1} = \bigcup s_j^{(i+1)} \) is constructed so that the intervals \( s_j^{(i+1)} \) do not overlap.

A convergence property of the algorithm is studied. It is shown that, as \( i \) goes to infinity, \( S_i \) covers \( S \) with probability one and the probability that \( y_i \in M^* \) tends to one as \( i \to \infty \), with \( M^* \) the set of all global minima of \( f \).

An implementation of the algorithm is given.

**Key Words.** Random search, global optimization, Lipschitz function.
A continuation approach via d.c. (difference of convex functions) algorithm for general distance geometry problems

Le Thi Hoai An
Mathematical Modelling and Applied Optimization Group
Laboratory of Mathematics - CNRS UPR 8050, National Institute for Applied Sciences-Rouen
BP 8, F 76 131 Mont Saint Aignan Cedex, France
Email: lethi@insa-rouen.fr

Distance geometry problems, which play a key role in the molecular optimization, have earned active researches in recent years. These problems for the determination of protein structures are specified by a subset $S$ of all atoms pairs and by the Euclidean distances $\delta_{ij}$ between atoms $i$ and atoms $j$ for $(i,j) \in S$. They initially consist in finding a set of $x^1, \ldots, x^n$ in $\mathbb{R}^3$ such that

$$||x^i - x^j|| = \delta_{ij}, (i,j) \in S. \quad (1)$$

It is called the exact distance geometry problem and can be formulated, in several ways, as global optimization problems.

In practice, lower and upper bounds on the distances are specified instead of their exact values. We then are faced with the so called general distance geometry problem

$$l_{ij} \leq ||x^i - x^j|| \leq u_{ij}, (i,j) \in S. \quad (2)$$

An important case of the general distance geometry problem is to obtain an $\varepsilon$-optimal solution of (1), namely a configuration $x^1, \ldots, x^n$ in $\mathbb{R}^3$ satisfying

$$||x^i - x^j|| - \delta_{ij} \leq \varepsilon, (i,j) \in S, \quad (3)$$

for some $\varepsilon > 0$. An $\varepsilon$-optimal solution is useful when the exact solution to the distance geometry problem (1) does not exist because of small errors in the data. Such a situation can happen, for example, when the triangle inequality

$$\delta_{ij} \leq \delta_{ik} + \delta_{kj}$$

is violated for atoms $\{i,j,k\}$ because of possible inconsistencies of the experimental data.

86
In this paper we are interested in the large-scale molecular conformation from the general distance geometry problem (2) with the new d.c. formulation:

$$0 = \inf \left\{ \sum_{(i,j) \in S} p_{ij} (|x^i - x^j|^2 - t_{ij}^2) : x^1, \ldots, x^n \in \mathbb{R}^3, \quad t_{ij} \leq t_{ij} \leq u_{ij}, \quad (i,j) \in S \right\},$$

(4)

where $p_{ij} = p_{ji} > 0$ for $i \neq j, (i,j) \in S$.

Our method is based on a continuation approach for global optimization in which the original objective function $f$ is transformed into a smoother function $\langle f \rangle_\lambda$ with fewer local minimizers (via the Gaussian transform). An optimization algorithm is then applied to the transformed function $\langle f \rangle_\lambda$, tracing their minimizers back to the original function. The basic idea of the continuation approach is to trace a curve $\{x(\lambda) : \lambda \geq 0\}$, where $x(\lambda)$ is a minimizer of $\langle f \rangle_\lambda$. For this we choose a sequence $\{\lambda_k\}$ of smoothing parameters that converges to zero and compute a minimizer $x_k$ of each $\langle f \rangle_{\lambda_k}$ by using an efficient algorithm in d.c. optimization approach called DCA. The DCA (introduced by Pham Dinh Tao 1988) is a primal-dual subgradient method for solving a general d.c. program that consists in the minimization of difference of convex functions. It is at the present time one of a few algorithms in the local approach which has been successfully applied to many large-scale d.c. optimization problems and proved to be more robust and efficient than related standard methods. Using local optimality conditions and duality in d.c. programming, it cannot guarantee the globality of computed solutions for general d.c. programs. However, we observe that with a suitable starting point it converges quite often to a global one. This property motivates us to investigate a continuation approach via DCA in the solution of (4). DCA seems to be robust and efficient in the large scale setting as proved by numerical simulations which furthermore indicated that DCA-continuation approach always converges to global solutions.
A Deterministic Global Optimization Approach for Solving the Problem of Packing Equal Circles in a Square

Marco Locatelli  
Dept. Sistemi e Informatica  
Università di Firenze  
via di S. Marta, 3  
50139 Firenze (Italy)  
Email: locatell@dsi.unifi.it

Ulrich Raber  
Department of Mathematics  
University of Trier  
54286 Trier  
Email: raber@uni-trier.de

The packing problem is a widely explored problem in the field of optimization. One tries to find the maximum radius $r$ of $n$ equal and non-overlapping circles belonging to the unit square. This problem can be formulated as

$$\begin{align*}
\max & \quad r \\
S(x_i, r) & \subseteq U \quad i = 1, \ldots, n \\
S(x_i, r) & \cap S(x_j, r) = \emptyset \quad 1 \leq i < j \leq n
\end{align*}$$

where, for each $i \in \{1, \ldots, n\}$, $S(x_i, r) = \{x \in \mathbb{R}^2 : ||x - x_i||_2 < r\}$ denotes the open sphere with center $x_i \in \mathbb{R}^2$ and radius $r$, and $U := [0, 1]^2$ denotes the unit square. The circle packing problem is equivalent to the problem of scattering $n$ points into the unit square such that the minimum pairwise distance becomes as large as possible. This point scattering problem can be formulated as a nonconvex all-quadratic problem in the following way

$$\begin{align*}
\max & \quad t \\
t - ||x_i - x_j||_2^2 & \leq 0 \quad 1 \leq i < j \leq n \\
x_i & \in U \quad i = 1, \ldots, n .
\end{align*}$$

Most of the optimal solutions of Problem (1) for more than 20 points are not known till now. We will present a rectangular branch-and-bound algorithm,
which is able to determine \( \varepsilon \)-optimal solutions of this problem. Even though Problem (1) with \( n > 20 \) is - from a global optimization point of view - a very large problem, the suggested algorithm is able to solve this problem with acceptable effort, at least for \( n \leq 27 \).

We will start with the presentation of some theoretical results showing the existence of optimal solutions of Problem (1) with a special behavior on the boundary of the unit square \( U \). These results state the intuitive fact that as many as possible members \( x_i \ (i = 1, \ldots, n) \) of an optimal solution \( x = (x_1, \ldots, x_n)^T \in U^n \) lie on or near to the boundary of \( U \).

After this we present the framework of our rectangular branch-and-bound algorithm. We will sketch the construction of an LP-relaxation for Problem (1), which exploits the special structure of this problem. In particular, we will see that this LP-relaxation is better than those obtained by general approaches for this type of global optimization programs. Moreover, the special structure of Problem (1) can be used in order to develop several subdivision set manipulation strategies:

- **Unique numbering strategy**: this tool tries to avoid that our algorithm looks for solutions, which differ only by the numbering of their members.

- **Symmetry avoiding strategy**: this tool tries to avoid that our approach considers solutions of Problem (1), which are symmetric to each other, i.e., which differ only by a rotation or a reflection.

- **Dimension reduction strategies**: using the mentioned theoretical results, it can be possible to diminish the dimension of some hyperrectangles, which are relevant during the execution of our algorithm.

- **Volume reduction strategies**: under some circumstances it is possible to cut away parts of the considered hyperrectangles, since these areas cannot contain solutions with a better distance behavior than the best known so far.

We will sketch the basic ideas of these subdivision set manipulation strategies. It should be noted that these strategies were the key for the acceleration of our rectangular branch-and-bound approach. The good LP-relaxations were not decisive.

We will finish the talk with some computational results. We were able to solve approximately Problem (1) with up to 27 points within 2 hours on a SUN ULTRA 60 workstation. Furthermore, we could determine approximate solutions of this problem with even more points \( (n > 30) \). By the way we have detected a solution for the case \( n = 32 \), which is slightly better than the best known so far.

The main advantage of the suggested approach for solving Problem (1) is the possible guarantee of the \( \varepsilon \)-optimality of determined solutions. The good numerical performance of our method essentially depends on the quality of the initial solutions, i.e., of the solutions which are already known. However, in the literature there are several papers (see, e.g., [4, 5]) with good solutions for
Problem (1), which can be used as such initial points. In the examined cases these known solutions were nearly always approximately optimal.

Another interesting aspect of our method is that this algorithm demonstrate the possibility of solving large problems also globally, if the structure of a special problem instance can be exploited. Note that we were able to solve a global optimization problem with a dimension of 55 and 351 quadratic concave constraints within two hours on a single processor.

References


Cone covering algorithms for concave minimization

Christophe Meyer
Ecole Polytechnique de Montréal
Département de Mathématiques et de Génie Industriel
C. P. 6079, succ. Centre-ville
Montréal (Québec) Canada H3C 3A7
Email: christop@cert.umontreal.ca

In 1964, in a seminal paper, Tuy proposed a simple algorithm for concave minimization over a polytope. This algorithm was shown to cycle by Zwart a few years later. Recently however it has been shown that despite this possibility of cycling, Tuy’s 1964 algorithm always finds the optimal solution of the problem. We recall this algorithm and present a small modification of it which simplifies the cycle detection.

4 Introduction

The concave minimization problem

\[(CP) \quad \min\{f(x) \mid x \in P\}\]

consists in finding a global minimizer \(x^*\) of the concave function \(f\) over the polytope \(P = \{x \in \mathbb{R}^n : Ax \leq b\}\) where \(A\) and \(b\) are a matrix and a vector of appropriate size.

In [6], Tuy proposed several ideas and algorithms for solving this problem, that still influence today’s methods (for surveys, see e.g., Horst and Tuy [3], Benson [2]). One of these algorithms, which consists in covering the polytope by cones, was shown to cycle by Zwart [7]. To avoid cycling, a small modification [8][9] was brought to Tuy’s algorithm, which results in a cone partitioning algorithm (rather than cone covering). However, rather surprisingly, it was shown in [5] that the original algorithm of Tuy, despite it may cycle, always finds an optimal solution of problem \((CP)\). We first recall Tuy’s 1964 algorithm and then explain how to modify it in order to simplify the cycle detection.
5 Tuy’s 1964 algorithm

Let $O$ be an extreme point of the polytope, which we assume for simplicity to be non-degenerated. The algorithm starts with the cone $K^0$ of origin $O$ and whose $n$ edges are determined by the $n$ adjacent extreme points to $O$.

At a given iteration we have a set of cones that cover the polytope $P$. Let $K = \text{cone}(O, x^1, \ldots, x^n)$ be one of these cones, where $x^j, j = 1, \ldots, n$ are $n$ extreme points of $P$. Along each halfline $(Ox^j)$, we compute the so-called $\gamma$-extension, i.e., the farthest point $y^j$ such that $f(y^j) \geq \gamma$ where $\gamma$ is the value of the current best known solution. Note that since $O \in P$, we can assume $\gamma \leq f(O)$. Consider the following linear program

$$(P) \quad \max \quad \sum_{j=1}^{n} \lambda_j$$

s.t. \quad \begin{cases} Aw \leq b \\ w = \sum_{j=1}^{n} \lambda_j y^j. \end{cases}$$

If its optimal value $\rho^*$ is less than or equal to 1, this means that the polytope $P$ lies in the halfspace defined by the hyperplane passing through the points $y^j, j = 1, \ldots, n$ and containing $O$, and hence that the intersection $K \cap P$ with the cone $K$ is included in the simplex $S = \text{conv}(O, y^1, y^2, \ldots, y^n)$. We then have

$$\min_{x \in K \cap P} f(x) \geq \min_{x \in S} f(x) = \min\{f(O), f(y^1), \ldots, f(y^n)\} \geq \gamma$$

which shows that $K \cap P$ cannot contain a point with value $< \gamma$. Therefore if $\rho^* \leq 1$, we eliminate the cone.

If $\rho^* > 1$, we will subdivide the cone $K$. Let $(\lambda^*, w^*)$ be an optimal solution of the linear program. For each $j$ such that $\lambda_j^* > 0$, we replace $x_j$ by $w_j$. It is well-known that the union of all these cones $K^j$ forms a cover of the cone $K$, therefore we replace $K$ by the cones $K^j$. The point $w^*$ is used to try to improve the value of the current best known solution.

The algorithm is then the following. Select a cone of the cover and construct the linear program. If the optimal value is less than or equal to 1, delete the cone and select another one. Otherwise replace the cone by a certain number of subcones $K^j$, as just explained, and try to update the value of the current record. Iterate until all cones have been deleted.

Let $\bar{x}$ be any point of $P$. By perturbing it slightly if necessary, we can assume that $\bar{x}$ cannot be written as a convex combination of less than $n + 1$ extreme points of $P$. Let $K = \text{cone}(O, x^1, \ldots, x^n)$ be a cone containing $\bar{x}$. Since $K$ is non-degenerated, there exists a unique $\lambda (\geq 0)$ such that $\bar{x} = \sum_{j=1}^{n} \lambda_j y^j$. We define $\rho_K(\bar{x}) = \sum_{j=1}^{n} \lambda_j$. Note that if $\rho_K(\bar{x}) \leq 1$, $f(\bar{x}) \geq \gamma$. 

92
The correctness of Tuy’s 1964 algorithm is a consequence of the following result:

**Proposition 1** If \( \rho_K(\hat{x}) > 1 \) and if \( K' \) is the cone obtained from \( K \) by subdivision that contains \( \hat{x} \), then

\[
\rho_{K'}(\hat{x}) < \rho_K(\hat{x})
\]

Since the number of possible cones is finite, we will eventually have \( \rho_K(\hat{x}) \leq 1 \), which shows that \( \hat{x} \) cannot improve on the best known solution.

The main inconvenient of this algorithm is that in order to detect cycle we have to keep the list of all cones generated since the beginning of the algorithm to check if a newly generated cone has already been processed. We now propose a small modification of Tuy’s algorithm that has a descent property which allows to forget all cones with a value greater than the current one.

### 6 A simple modification of Tuy’s algorithm

Note that in the proof of the convergence of Tuy’s algorithm, we used only positive \( \lambda \). Hence, following Balz and Zwart, we modify the linear program by adding the constraint \( \lambda \geq 0 \) (we also eliminate the variable \( u \)):

\[
\text{(P1) } \quad \max \sum_{j=1}^{n} \lambda_j
\]

\[
\text{s.t. } \quad \begin{cases}
\sum_{j=1}^{n} \lambda_j \mathbf{a}_j^i \leq b \\
\lambda \geq 0
\end{cases}
\]

Its dual is

\[
\text{(D1) } \quad \min \mu^* b
\]

\[
\text{s.t. } \quad \begin{cases}
\mu^* \mathbf{a}_j^i \geq 1, & j = 1, \ldots, n \\
\mu \geq 0
\end{cases}
\]

If the optimal value \( \rho^* \) is less than or equal to 1, we can still eliminate the cone. If \( \rho^* > 1 \), let \( \lambda^* \) and \( \mu^* \) be respectively an optimal solution of the primal and of the dual. It was shown (see e.g., [4]) that the hyperplane \( H = \{ x \in \mathbb{R}^n : \mu^* A x = \rho^* \} \) supports the polytope \( P \) at point \( u^* = \sum_{j=1}^{n} \lambda_j^* \mathbf{a}_j^i \), hence defines a face \( F \) of \( P \).

The subdivision point \( u' \) is obtained by solving a second linear program:

\[
\text{(P2) } \quad \max \sum_{j=1}^{n} \lambda_j
\]

\[
\text{s.t. } \quad \begin{cases}
A u \leq b \\
w = \sum_{j=1}^{n} \lambda_j \mathbf{a}_j^i \\
\mu^* A w = \rho^* b
\end{cases}
\]
(Note that this linear program is the same than the one used by Tuy to compute the subdivision point, except that we add the requirement that the point belongs to the face of $P$ determined by the solution of the first linear program.)

We have the following result:

**Proposition 2** Let $K'$ be a subcone of $K$ by subdivision with respect to $u'$. Then either $f(u') < \gamma$, or $\rho^*(K') \leq \rho^*(K)$.

By selecting appropriately the cone to subdivide (i.e., by choosing the one with greatest value $\rho^*$), the algorithm generates a sequence of cones in non-increasing order of $\rho$. Cycling is only possible between cones of same value. To detect cycling, it suffices now to maintain a list of the cones with value equal to the current value of $\rho$ (rather than maintaining a list of all cones).

An adaptation of the proof used to prove the validity of Tuy's algorithm shows that this modified algorithm still finds the optimal solution of problem (CP).

**References**


Locally Exact Minorants And Optimality Cuts
For Globally Solving Quadratically Constrained
Quadratic Optimization Problems

Ivo Nowak
Institut für Mathematik - Angewandte Mathematik
Humboldt-Universität zu Berlin
Unter den Linden 6
D-10099 Berlin
Germany
Email: ivo@mathematik.hu-berlin.de

We introduce a new class of convex minorants for globally solving quadratically constrained quadratic optimization problems (QQP) of the form

\[
\text{global minimize} \quad f_0(x) \\
\text{subject to} \quad f_k(x) \leq 0, \quad 1 \leq k \leq m.
\]

where \( f_k(x) := x^T A_k x + b_k^T x + c_k \) are quadratic forms. A well-known difficulty of solving QQP by a branch-and-bound algorithm is that often regions containing a global minimizer have to be subdivided very often in order to achieve that lower bounds are almost exact. In large dimensions this can prevent the method from terminating in reasonable time. We propose two methods for avoiding this difficulty:

1. **Locally Exact Lower Bounds**
   Given a local minimizer \( x^* \) which fulfills a second order optimality condition (SOC) we compute a locally exact lower bound with respect to \( x^* \), i.e., the lower bound is exact applied to all sets \( \Omega \cap S \) where \( \Omega \) is the feasible set of QQP, \( x^* \in S \) and \( \text{diam } S \) does not exceed a positive value. An important property of locally exact bounds is that they lead to finitely many iterations of a branch-and-bound procedure.

2. **Optimality Cuts**
   Using the new convex minorants it is possible to derive cutting planes which allow to cut off regions which contain local minimizers fulfilling SOC.

**Outline of the Method**
Given a local minimizer \( x^* \) of QQP we define

\[
\mathcal{B} := \{ k \in \{1, \ldots, m\} : \lambda_k > 0 \}
\]
where \( \lambda_k \) is a Lagrange multiplier corresponding to an active constraint \( f_k(x) \leq 0 \). Let \( S \) be a partition set of the feasible set of QQP containing \( x^* \). Using the gradients \( \nabla f_k(x^*) \), \( k \in B \), we construct a linear cone \( K(x^*) \) such that \( K(x^*) \supset S \) and \( x^* \) is a KKT-point of the quadratic optimization problem \( \text{QP}[x^*] \):

\[
\min_{x \in K(x^*) \cap H} f_0(x)
\]

where \( H \) is a half-space defined by a cutting plane. A lower bound of the optimal value of \( \text{QP}[x^*] \) is computed by

\[
\Psi^* = \max_{\alpha \geq 0, \alpha^2} \min_{x \in H} f_0(x)
\]

where

\[
f_0(x) = f_0(x) + \sum_{k \in I_1} \alpha_k \varphi_k(x) + \sum_{k \in I_2} \alpha_k^2 \varphi_k(x).
\]

The functions \( \varphi_k(x) \) are quadratic functions which are non-positive over \( K(x^*) \cap H \). The index set \( I_1 \) corresponds to inequality constraints and \( I_2 \) to equality constraints respectively. Obviously \( \Psi^* \) is lower bound of the global minimum of \( f_0(x) \) over \( \Omega \cap S \). \( \Psi^* \) is called Lagrangian relaxation, dual estimate or Shor’s relaxation. We can prove the following global optimality criterion (GOC):

Let \( x^* \) be a global minimizer of \( \text{QP}[x^*] \). Then \( \Psi^* = f_0(x^*) \) if and only if there exist \( \tilde{\alpha} = (\tilde{\alpha}^1, \tilde{\alpha}^2) \) with \( \tilde{\alpha}^1 \geq 0 \) such that \( \nabla^2 f_0 \geq 0 \), \( \nabla f_0(x^*) = 0 \) and \( f_0(x^*) = f_0(x^*) \).

It can be shown that, if \( \nabla^2 f_0(x) \) is positive semidefinite on the tangent space of \( K(x^*) \) at \( x^* \), a half-space \( H \) can be determined such that the right hand-side of GOC holds. This proves that \( \Psi^* \) is a locally exact bound for QQP. The cutting plane which defines \( H \) can be computed explicitly by solving an eigenvalue optimization problem.

**Implementation**

We coded a branch-and-bound algorithm for solving QQP in C++ which uses locally exact bounds and optimality cuts. The max-min problem (1) is formulated as an eigenvalue optimization problem which is solved by the bundle method NOA3 (Kiwiel). NOA3 is also used for computing optimality cuts. Subgradients are computed using the Lanczos algorithm ARPACK++ (Sorensen, Lehoucq, Yang and Maschhoff). Since only matrix-vector products are required it possible to exploit problem structure, such as sparsity of Hessians. For the local minimization we used MINOS5.5 (Murtagh and Saunders) and some C++ class definitions of LEDA3.6 (Mehlhorn, Näher, Seel, Uhrig) are used for representing the branch-and-bound tree.
Evaluating uEGO, an evolutionary global optimization algorithm

P. M. Ortigosa
Dept. Computer Architecture and Electronic
University of Almería
La Cañada de San Urbano
04120 Almería
Email: ortigosa@ualm.es

I. García
Dept. Computer Architecture and Electronic
University of Almería
La Cañada de San Urbano
04120 Almería
Email: igarcia@ualm.es

Márk Jelasity
Research Group on Artificial Intelligence
MTA-JATE, Szeged, Hungary
Email: jelasil@inf.u-szeged.hu

In this paper, uEGO, a new general technique for accelerating and/or parallelizing existing search methods is analyzed. uEGO is a generalization and simplification of GAS, a genetic algorithm (GA) with subpopulation support, so the niching (i.e. clustering) technique of GAS can be applied along with any kind of optimizers. Additionally, uEGO can be effectively parallelized. Empirical results on real functions including analysis of the effects of the user-given parameters and a comparison to a hill climber, a multistart hill climber and GAS algorithms are presented.

Introduction

uEGO stands for Universal Evolutionary Global Optimizer. Though this method is not 'evolutionary' in the usual sense, we have kept the name for historical reasons. The predecessor of uEGO was GAS, a steady-state genetic algorithm with subpopulation support. GAS offers a solution to the so-called niche radius problem which is a common problem of many simple niching techniques such as fitness sharing ([2]), simple iteration or the sequential niching [1].

The common part of uEGO with GAS is the species creation mechanism and the 'cooling' method. However, the species creation and 'cooling' mechanism has been logically separated from the actual optimization algorithm, so it is possible to implement any kind of optimizers that work 'inside a species'. This allows the adaptation of the method to a large number of possible search domains.
using existing domain specific optimizers while enjoying the advantages of the old GAS-style subpopulation approach. In this paper, an algorithm called sasG, proposed by Solis and Wets (85), has been used as the optimizer algorithm. Our results confirm that UEGO can indeed outperform GAS at least on the problems and parameter settings we have considered.

**Description of UEGO**

In this section the basic concepts, the algorithm, and the setting of the parameters are outlined. In UEGO, a domain specific optimizer (i.e. SASS) has to be implemented.

A key notion in UEGO is that of a species. A species can be thought of as a window on the whole search space. This window is defined by its center and a radius. The center is a solution, and the radius is a positive number. Of course, this definition assumes a distance defined over the search space. The role of this window is to 'localize' the optimizer which is always called by a species and can 'see' only its window, so every new sample is taken from there. This means that the largest step made by the optimizer in a given species is no larger than the radius of the given species. If the value of a new solution is better than that of the old center, the new solution becomes the center and the window is moved though it keeps the same radius value.

The radius of a species is not arbitrary; it is taken from a list of decreasing radii, the radius list. The first element of this list is always the diameter of the search space. If the radius of a species is the $i$th element of the list, then we say that the level of the species is $i$. The parameter levels indicates the maximal number of levels in the algorithm. Every level $i$ (i.e. for levels from $1, levels$) is characterized by $r_i$, new; and $n_i$; where $r_i$ is the radius value, and new; and $n_i$ are the number of function evaluations at the species creation and at the optimizer procedures, respectively.

During the optimization process, a list of species is kept by UEGO. The algorithm is in fact a method for managing this species list (i.e. creating, deleting and optimizing species). The maximal length of the species list is given by max_spec_num.

The following is a short algorithmic description of UEGO.

UEGO

init_species_list()
optimize_species(n[1])
for $i = 2$ to levels
    create_species(new[i]/length(species_list))
    fuse_species(r[i])
    shorten_species_list(max_spec_num)
    optimize_species(n[i]/max_spec_num)
    fuse_species(r[i])
rof
OGEU

98
**Init.species_list.** A new species list consisting of one species with a random center at level 1 is created.

**Create_species(evals).** For every species in the list, random pairs of solutions in the 'window' of the species are created, and for every such pair the objective function is evaluated at the middle of the section connecting the pair. If the objective function value of the middle is worse than the value of the pair, then the members of the pair are inserted in the species list. Every new inserted species is assigned the actual level value \( a \). The motivation behind this method is simple: to create species that are on different 'hills' so ensuring that there is a valley between the new species. The parameter of this procedure (evals) is an upper bound of the number of function evaluations. Note that this algorithm needs a definition of section in the search space.

**Fuse.species(radius).** If the centers of any pair of species from the species list are closer to each other than the given radius, both species are fused. The center of the new species will be the one with the best function value while the level will be the minimum of the levels of the original species.

**Shorten_species_list(max_spec_num).** It deletes species to reduce the list length to the given value. Higher level species are deleted first.

**Optimize_species(evals).** Execute the optimizer (SASS) for every species with a given number of evaluations (evals).

It is clear that if for some level \( i \) the species list is shorter than the allowed maximal length, \( \text{max.spec.num} \), the overall number of function evaluations will be smaller than \( n_i \). In our implementation we make use of the difference of the actual number of function evaluations and \( n_i \), which is used to create species. This technique has no effect when there are many species but if the number of species is small, a lot of extra effort is devoted to find new ones.

Finally, let us make a remark about a possible parallel implementation. The most time-consuming parts of the basic algorithm is the creation and optimization of the species. Note that these two steps can be done independently for every species, so each species can be assigned to a different processor.

**Experiments with Real Functions**

In this section experimental results on real functions will be presented. For real functions the optimizer used by UEGO was the hill climber suggested in [6] (SASS). No fine tuning of the parameters of the optimizer was done.

Our experiments consist of three stages: the first stage of training is intended to determine the values of the free parameters of UEGO which produce good solutions; the second stage of testing has been designed for evaluating and comparing the performance of UEGO in front of several similar algorithms (the ancestor of UEGO, GAS [4] and the simple and multistart version of a hill climber described in [6]). In the third stage of experiments we try to confirm the conclusion obtained at the first stage. In these experiments a set of 49 standard test functions was used, most of them are described in [5].

The first stage of experiments has been carried out on a set of four different test functions (F1, F2, F3, F4) that differ only in controllable features. This will
allow us to analyze the effect of only one separated feature of the test problem: the number of local optima and the dimension $n$ of the function.

Our experimental results have shown that both the clustering technique and the level-based "cooling" techniques, exhibit some advantage over its predecessor. The advantages of the uEGO clustering technique and the level-based "cooling" technique, which was proved in [3] for combinatorial problem, has been also confirmed on the real domain. uEGO outperforms the MHC, MHC and GAS algorithms (at least for the examined test problems).

To summarize the results, we can conclude that uEGO is a reliable global optimization that not only finds the global maximum/maximum, but also several local optima. On the other hand, uEGO can be parallelized as effectively as MHC.

References


Global optimization approaches for distance geometry problems

Panos M. Pardalos
Center for Applied Optimization
Industrial and Systems Engineering Department
303 Weil Hall, University of Florida
PO Box 116595, Gainesville, FL 32611-6595
Email: pardalos@cao.isie.ufl.edu

The distance geometry problem is that of determining the coordinates of a set of points in space from a given set of pairwise distance measurements. Distance geometry constraints are used to determine protein structures and can be formulated as global optimization problems. We discuss several computational methods for solving the related global optimization problems.
An Integrated Application Development Environment for Continuous Global Optimization

Janos D. Pinter
Pinter Consulting Services and Dalhousie University
Halifax, NS, Canada
jdpinter@is.dal.ca

The program system LGO serves to analyse and solve global optimization (GO) problems under 'minimal' (Lipschitz-continuity or continuity) structural assumptions. This makes it applicable to a broad range of GO models, specifically including stand-alone ('black box') systems, or models defined by tentative - possibly changing - analytical information.

LGO seamlessly integrates a suite of global and local search methods that can be activated in both automatic and interactive operational modes. The program system also has model visualization and report generation features, to assist the application development process.

Current LGO implementations include a fully Windows-compatible version, as well as a version equipped with a simpler graphical interface, and a command line style version. The latter two can also be directly migrated to workstations and mainframes.

The Windows version is embedded under a menu-driven user interface: this enables model formulation, solution, and visual/textual result analysis. LGO is supported by concise on-line help files, and it is accompanied by a User's Guide. This version can also be connected to a range of other development platforms, and also to complete external application program systems.

In recent years, LGO has been applied to a variety of GO problems of real-life size and complexity. During the presentation, we shall illustrate the complete functionality of the LGO system, by solving several such application examples.
Global Optimization of Lennard-Jones Clusters

Fabio Schoen  
Dept. Sistemi e Informatica  
Università di Firenze  
via di S. Marta, 3  
50139 Firenze  
Email: schoen@ingfii.ing.unifi.it

Marco Locatelli  
DSI  
Univ. Firenze  
via di S. Marta, 3  
50139 Firenze (Italy)  
Email: locatell@dsi.unifi.it

The optimization of the potential energy function of Lennard-Jones clusters of atoms has attracted much theoretical as well as computational research in recent years. One reason for this is the practical importance of discovering low-energy configurations of clusters of atoms, in view of applications and extensions in molecular conformation research; another reason of the success of Lennard Jones minimization in the global optimization literature is the fact that this is an extremely easy-to-state problem, yet it poses enormous difficulties for any unbiased global optimization algorithm.

Several attempts have appeared in the literature for the minimization of the Lennard-Jones potential:

\[ E = E(X_1, \ldots, X_N) = \sum_{i=1}^{N} \sum_{j<i} \left( \frac{1}{||X_i - X_j||^2} - \frac{2}{||X_i - X_j||^6} \right) \]

where \( X_i \in \mathbb{R}^3 \) represent the coordinates of the center of the \( i \)-th atom. Among the most successful approaches, we can cite

- lattice methods, in which a discrete local optimization is performed, with atoms constrained to belong to a pre-specified lattice of points, followed by a continuous local search;
- forward and reverse methods, in which either an atom is added to a low energy configuration with \( N - 1 \) atoms, or the least tightly bonded atom is removed from a low energy configuration with \( N + 1 \) atoms and then a local search is started;
• simulated annealing methods, in which starting from a random configuration, new molecular conformations are generated through random perturbation; each new configuration may be accepted not only if it corresponds to a lower energy one but also if it is worse than the current one. This last event, however, happens only with a specified probability whose definition is part of the algorithm itself.

To date many putative global optima in the range $N \leq 147$ are known, most of which can be easily discovered through lattice methods. There are however important exceptions which correspond to hard challenges for any method devoted to Lennard-Jones potential energy minimization. In particular the cases $N = 38$ and $N = 75, 76, 77$ are considerably harder than most of the other in this range. To date the only unbiased method which discovered all putative optimum configuration is simulated annealing applied to the transformed energy landscape obtained by starting a local search from every sampled point. This method, among other results, was capable to discover the “ideal” truncated octahedron for $N = 38$ roughly once every 330 local searches and the Marks decahedron for $N = 75$ roughly once every $10^6$ local searches.

In this paper we present a class of new approaches, based upon variations of classical Multistart global optimization methods, which appear to be much more efficient than previously published numerical results, at least in the test range $N \leq 80$. The numerical experiments were performed on Pentium PC’s and did not extend after the $N = 80$ case only because of the time necessary to obtain sufficiently reliable statistics.

The methods we implemented were capable of discovering all the known putative global optima and, in particular, produced the $N = 38$ optimum once every 4 local searches and the extremely hard $N = 75$ case roughly once every 2500 local searches, thus improving the previously best result by almost two orders of magnitude.

The proposed approach sequentially performs local optimizations according to the following scheme:

1. a random initial atom configuration is generated in a box and slightly modified by means of a greedy technique;
2. the potential energy function is modified
3. a local optimization is started for the modified potential energy
4. a local optimization is started from the local optimum obtained during the previous optimization, with the original potential function as the objective

The key issue in this scheme is the definition of the modified potential function. We introduced several modifications and, in particular, we run tests based upon a potential modified in the following way:

$$\sum_{i=1}^{N} \sum_{j<i} \left( \frac{1}{||X_i - X_j||^{2p}} - \frac{2}{||X_i - X_j||^p} \right)$$
where $p$ is an integer ($p = 6$ corresponds to the original Lennard-Jones potential);

2. a penalty term is added which enforces lower interatomic distance:

$$\mu_1 \sum_{i \neq j} ||X_i - X_j||$$

where $\mu_1$ is a weight factor;

3. a penalty term is added for the diameter of the resulting cluster. In particular a term is added of the form

$$\mu_2 \sum_{i \neq j} (\max\{0, ||X_i - X_j||^2 - D^2\})^2$$

where $\mu_2$ is a weight factor and $D$ is a threshold parameter.

The first modification, if $p < 6$ is chosen, aims at modifying the shape of the potential energy by reducing the barrier effect caused by a couple of atoms at low distance; the second modification has the effect of closing the gap between pairs of atoms; the third component, which was crucial in solving the difficult case $N = 75$, gives a strong penalty for molecules with a diameter higher than the threshold $D$. Overall, the modified potential has the effect of producing quite compact forms, which are then relaxed through the following optimization, which is based on the original, unmodified, Lennard-Jones potential.
"Divide the Best" global optimization algorithms

Yaroslav D. Sergeyev
Institute of Systems Analysis and Information Technology,
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
Email: yaro@si.deis.unical.it

Many algorithms have been proposed for solving the general global optimization problem

$$\min f(x), x \in M,$$

of finding the set $X^*$ of the global minimizers and the corresponding value $f(x^*), x^* \in X^*$, under various assumptions on the objective function $f(x)$ and the search region $M$ (a wide literature is presented in Archetti and Schoen (1984), Dixon and Szegő (1978), Floudas and Pardalos (1996), Hansen, Jaumard and Lu (1992), Horst and Pardalos (1995), Horst and Tuy (1996), Pardalos and Rosen (1990), Pintér (1996), Ratschek and Rokne (1988), Rimooy Kan and Timmer (1989), Strongin (1978), Törn and Zilinskas (1989), etc.).

It has been noted that some of the algorithms (of both stochastic and deterministic type) have a very close structure. Therefore several papers trying to construct a general framework for describing computational schemes and proving convergence conditions of the algorithms in a unified manner have been published. In Horst and Tuy (1987), Horst (1988), Horst and Tuy (1996) a branch-and-bound approach is effectively used to gain this object. Grishagin (1979), (1983) and Pintér (1983), (1986), (1992) developed two other close approaches independently.

In this paper a new class of multidimensional global optimization algorithms (called "divide the best" algorithms) is proposed. The class unifies and generalizes the classes of the characteristic methods and the adaptive partition algorithms introduced by Grishagin and Pintér respectively. The new scheme includes also some methods which do not fit either the characteristic or the adaptive partition families. A detailed convergence study is presented. A special attention is paid to cases where sufficient conditions of everywhere dense, local and global convergence are fulfilled only over subregions of the search domain.
References


A general framework for the stochastic/deterministic hybridization with global convergence results

José Eduardo Souza de Cursi
Dept. Mécénique
INSA - Rouen
Place Emile Blondel BP 8
76131 Mont Saint Aignan CEDEX
Email: souza@insa-rouen.fr

This work is mainly concerned by global optimization methods adapted to a class of inverse problems arising from Mechanical Engineering. The approach considered has been initially conceived for situations as

1. the estimation of material parameters of composite anisotropic parts;
2. the control of the curing of elastomers;
3. the estimation and the control of the position of a solidification/melting front;
4. the strategic planification of the physical distribution of goods (modelling, simulation and fleet definition)

These situations present the following characteristics

1. the objective function is nonconvex but regular;
2. its evaluation involves the numerical resolution of bidimensionnal or tridimensionnal Partial Differential Equations (eventually non-linear and involving time). So, a sequence of Finite Element Analysis must be performed in order to get a single value of the objective function;
3. derivatives of the objective function can be obtained by solving analogous Partial Differential Equations. Thus, here yet, a sequence of Finite Element Analysis must be performed in order to get a gradient of the objective function. However, the cost of the evaluation of the derivatives is comparable to the cost of the evaluation of the objective function;
4. restrictions on the design parameters can be explicitly formulated and correspond inequalities involving regular functions of the parameters.
In order to simplify the presentation, let us consider a generic unconstrained situation. We denote the objective function by $F$ and the parameters by $x$. We denote by $d$ the dimension of $x$. The model problem reads as

$$x \in \mathbb{R}^d \quad \text{and} \quad F(x) = m = \min F \quad .$$

The set of solutions is denoted $S^*$. Let us introduce a local search method for this problem: an initial guess $x_0$ is given and a sequence $(x_n)_{n \geq 0}$ is generated by an iterative procedure. For instance, we can consider a generic descent method which reads as

$$x_{n+1} = Q(x_n) \quad , \quad n \geq 0 \quad .$$

Since the objective function $F$ is nonconvex, the sequence may converge to a local minimum and do not verify $F(x_n) \rightarrow m$. For unconstrained minimization, a simple way to prevent this consists in adding a random perturbation to the basic method defined by $Q$: we consider a sequence of random vectors $(P_n)_{n \geq 1}$ and

$$x_{n+1} = Q(x_n) + P_n \quad . \quad (1)$$

It is shown in the literature that, for suitable $P_n$, the sequence above satisfies $F(x_n) \rightarrow m$ independently of the initial point $x_0$ (see Pogu & Souza de Cursi 1994). We denote by $Id$ the Identity Matrix of order $d$, $Z$ a random vector of the same dimension $d$, having the standard Gaussian distribution $N(0, Id)$. An example of suitable perturbation is

$$P_n = \rho_n Z \quad ,$$

where

$$\rho_n = a \sqrt{c \log(n + n_0)} \quad , \quad a, c, n_0 = \text{strictly positive constants.}$$

In practice, we have to generate a finite sample. For instance, we can consider the $NR + 1$ values $P^1_n = 0$ and $P^n_{i}; i = 0, \cdots, NR$ as random values from $P_n$. Then

$$x_{n+1} = \arg \min_{M_n} F \quad ; \quad M_n = Q(x_n) + P^i_n ; i = 0, \cdots, NR \quad .$$

This particular choice of $P_n$ leads to the following theorem:

**Theorem:** Let $U_n = \min \{ F(x^n_j), \ 1 \leq k \leq n, \ 1 \leq j \leq np \}, \ \forall n \in N : U_{n+1} \leq k \leq U_n$. Thus, $(U_n), n \geq 0$, is a sequence of decreasing bounded monotonous random variables and

$$U_n \rightarrow m \quad \text{as} \ n \rightarrow \infty.$$

The method and this result extend straightforwardly to more general perturbations $P$. 

110
This algorithm generates a single solution. In order to get multiple solutions, an evolutionary version may be considered. In fact, an evolutionary version of this method can be easily derived by considering that an evolutionary algorithm (EA) is entirely defined by a set of three rules: a reproduction rule \( R \), a mutation rule \( M \), a selection rule \( S \). So, a generic EA reads as follows:

1. We give an initial population \( S_0 \).
2. At step \( n \) the actual population is \( S_n \).
   
   [a] The "children" of \( S_n \) are \( R_n = R(S_n) \). This step is usually referred as reproduction (crossing/over).

   [b] The "mutation" of \( S_n \) are \( M_n = M(S_n) \). This step is usually referred as mutation.

   [c] The new population is \( S_{n+1} = S(A_n) \), where \( A_n = S_n \cup R_n \cup M_n \). This step is usually referred as selection.

This remark leads to an unified approach where local search appears as an evolutionary algorithm generated by a particular choice of the rules. For instance, the basic descent method considered above corresponds to

- Choose an initial population \( S_0 = \{x_0\} \in E \);
- Generate a sequence of populations \( \{S_n\}, n \geq 1, S_n = \{x_n\} \subset E \), with \( F(S_n) = \emptyset, M_n = \{Q(x_n)\}, S(A) = A \).

Into an analogous way, the algorithm involving random perturbations can be interpreted as a modification of the rules of mutation and selection. This interpretation furnishes a simple way to combine local search, stochastic methods and evolutionary algorithms. For instance, the combination between the basic local method \( Q \) and an elitist evolutionary algorithm involving reproduction by affine combinations of elements reads as

1 - Let be given the initial population \( S_0 \) formed by \( np > 0 \) elements.
2 - \( S_{n+1} \) is obtained from \( S_n \) by performing:
   
   [a] the generation of \( nc \) children from \( S_n \). A simple way to do this consists in using a random combination of the form \( \alpha x^j + \beta x^k + \gamma \)
   of elements \( \{x^j, x^k\} \subset S_n \).

   [b] the generation of the mutations of the elements from \( S_n \) by (1).

   [c] the selection of the best \( np \) elements from \( A_n \).

The global convergence of such an EA is stated in the following result:

**Theorem** We consider a population \( S \) ordered according to the values of \( F \), i.e., \( F(x_1^1) \leq \cdots \leq F(x_{np}^1) \) then \( U_n \rightarrow m \) as \( n \rightarrow \infty \) a.s.

All the algorithms considered extend straightforward to more general perturbations \( P \) and can be adapted to constrained optimization. For instance, they can be adapted to Uzawa's method, projected gradient, penalty methods.
Moreover, the selection operator can also be interpreted in terms of dynamics. For instance, we can modify the elitist selection of the algorithms presented in order to take into account other dynamics such as, for instance, the Metropolis or Tsallis ones.

Thus, the approach exposed leads to an unifying framework which includes Simulated Annealing procedures, local search methods and evolutionary algorithms. In addition, the framework constructed leads to, on the one hand, a simple way to derive hybrid algorithms and, on the other hand, formal results of convergence.

Methods for the numerical resolution of algebraical systems can also be introduced in this framework. For instance, the usual Robbins-Monro procedure of recursive estimation can be adapted in order to solve deterministic systems of equations or minimize an objective function (in this case, the basic method Q becomes the Robbins-Monro procedure).

The numerical experiments performed have shown that the methods are effective to calculate. Results from numerical experiments will be presented.

Future work may concern extensions to non-differentiable optimization, discrete problems. The use of recent results on the asymptotical behavior of record statistics may also to be considered in order to propose stopping criteria.

References


D.C. Programming: Theory and Algorithms

Alexander Strekalovsky
Institute of System Dynamics and Control Theory SB of RAS
Lermontov Str., 134
Irkutsk-33, 664033, Russia
Email: strekal@icc.ru

In this paper we consider the following problems of d.c. (difference of two convex functions) programming:

\[ F(x) \triangleq g(x) - f(x) \downarrow \min, \quad x \in D; \quad (1) \]

\[ h(x) \downarrow \min, \quad F(x) \triangleq g(x) - f(x) \geq 0, \quad x \in S; \quad (2) \]

where \( g(\cdot) \) and \( f(\cdot) \) are convex functions over \( R^n \), \( D \) and \( S \) are some sets from \( R^n \) and \( h(\cdot) \) is a continuous function.

It can be readily seen, that the convex maximization and reverse convex problems:

\[ f(x) \uparrow \max, \quad x \in D; \quad (3) \]

\[ h(x) \downarrow \min, \quad g(x) \geq 0, \quad x \in S; \quad (4) \]

are the particular cases of Problems (1) and (2) respectively. It suffices to set \( g(x) \equiv 0 \) in (1) and \( f(x) \equiv 0 \) in (2). To begin with, we propose so-called Global Optimality Conditions (GOC) for Problems (1) and (2).

Suppose that \( f(\cdot) \) is differentiable and there exists \( v \in R^n \), s.t.

\[ F(v) > F(z) \triangleq \zeta, \quad v \in D. \quad (H) \]

It is easy to see that the assumption \((H)\) is natural for Problem (1), if a point \( z \in D \) is suspected to be a global solution to Problem (1).

The following theorem gives a characterization of a global solution to (1).

**Theorem 1.** [1]

A feasible point \( z \) is a global solution to Problem (1) \((z \in Sol(1))\), if

\[ \forall \beta; \beta - f(y) = \zeta \triangleq \sum g(z) - f(z), \]

\[ y \in D, \quad g(y) \leq \beta \leq \sup (g, D), \]

\[ g(x) \geq \beta \geq \langle \nabla f(y), x - y \rangle \quad \forall x \in D. \quad (E) \]
If in addition (H) takes place then (E) becomes sufficient for $z \in D$ being a global solution to (2).

Consider the following assumption for Problem (2)

$$\operatorname{Sol}(2) \cap \{x | F(x) > 0\} = \emptyset. \quad (G)$$

It means that there is no solution of Problem (2) $x_0 \in S$ s.t. $F(x_0) > 0$.

First, we present the necessary condition for Problem (2).

**Theorem 2.** Let $z \in \operatorname{Sol}(2)$, $F(z) = 0$, and the assumption (G) takes place. Then

$$\forall (y, \beta) : \beta = g(y), \forall y^* \in \partial g(y), \forall x \in S, h(x) \leq h(z), f(x) - \beta \geq \langle y^*, x - y \rangle. \quad (E1)$$

There is a gap between necessary and sufficient Global Optimality Conditions for Problem (2).

Suppose that $S$ is convex set and consider the following assumptions

$$\exists v \in S : F(v) < 0; \quad (5)$$

$$\forall y \in S : g(y) = f(y) \text{ (i.e., } F(y) = 0) \quad \exists y^* = y^*(y) \in \partial g(y), \exists p = p(y, y^*) \in S, \quad f(p) - f(y) < \langle y^*, p - y \rangle. \quad (H1)$$

Now we can formulate the sufficient condition for problem (2).

**Theorem 3.** Suppose the assumptions (5), (H1) hold. Besides $h()$ is upper semicontinuous and $F(z) = 0$. Then the condition

$$\forall (y, \beta) : g(y) = \beta, \ y \in S, \ f(y) \leq \beta \leq \sup(f, S), \ \exists y^* \in \partial g(y) : \forall x \in S, h(x) \leq h(z), \ f(x) - \beta \geq \langle y^*, x - y \rangle. \quad (E1)$$

is sufficient for the point $z$ to be Global Solution to (2). ($z \in \operatorname{Sol}(2).$)

It is easy to see, that the GOC for (3) [2] follows from (E) with $g(x) \equiv 0$. In the same manner GOC for reverse convex problem (4) [3] follows from GOC for (2) when $f(x) \equiv 0$. So the relation between GOC completely corresponds to the relation between the statements of Problems.

The constructed Theory of GOC is connected with Classical Extremum Theory and may be interpreted by means of linearization idea. Moreover, these theory possess, as classical optimality conditions (OC), so-called Algorithmic Property (AP). It means that if OC is broken down at a point, there can be constructed a feasible point which is better than the point of interest.

So, there exists a possibility to develop Global Search Algorithms based upon GOC [4]. We exploited the possibility and proposed such Solving Methods for Problems (1) and (2).
Now let us put the idea described above in detail. Let be given a feasible point \( x^\lozenge \in D \) and a number sequence \( \{\varepsilon_k\} \), \( \varepsilon_k \downarrow 0 \) \( (k \to \infty) \). Set \( k := 0 \) and describe a global search algorithm step by step.

Step 1. Starting at the point \( x^k \in D \) obtain a stationary (critical) point \( z^k \in D \) by means of a local search method for (2).

Step 2. Choose a number \( \beta \in [m, M] \), where \( m \triangleq \inf(g, D), M \triangleq \sup(g, D) \). To begin with, one may set \( \beta_0 := g(z^k) \).

Step 3. Construct an approximation
\[
R_k = \{ y^1, ..., y^N | f(y^j) = \beta - \zeta_k \}, \quad \zeta_k := F(z^k).
\]

Step 4. For \( i = 1, ..., N \) solve Linearized Problem
\[ g(x) - \langle \nabla f(y^j), x \rangle, \quad x \in D. \]
Let \( u^i \) be an \( \varepsilon_k \)-solution to \((PL_d)\).

Step 5. For \( i = 1, ..., N \) find point \( u^i \), \( f(u^i) = \beta - \zeta_k \),
\[
\langle \nabla f(u^i), u^i - u^j \rangle + \varepsilon_k \geq \sup_v \{ \langle \nabla f(v), u^i - v \rangle \mid f(v) = \beta - \zeta_k \}.
\]

Step 6. Set
\[
\eta_k(\beta) = g(u^i) - \beta - \langle \nabla f(u^i), u^i - u^j \rangle
= \min_{1 \leq i \leq N} \{ g(u^i) - \beta - \langle \nabla f(u^i), u^i - u^j \rangle \}.
\]

Step 7. If \( \eta_k(\beta) < 0 \), set \( x^{k+1} = u^i \) and go to step 1.

Step 8. If \( \eta_k(\beta) \geq 0 \), then set \( \beta := \beta + \Delta \beta \) by means of any one-dimensional minimization method and go to step 3.

Step 9. If \( \eta_k(\beta) \geq 0 \forall \beta \in [m, M] \), \( \varepsilon_k > \delta \), where \( \delta \) is a given tolerance, then set \( x^{k+1} = z^k, k := k + 1 \), and go to step 1.

Step 10. If \( \eta_k(\beta) \geq 0 \forall \beta \in [m, M] \), and \( \varepsilon_k \leq \delta \), then stop.

The Algorithms are tested on such the problems as quadratic optimization on boxes, balls, polyhedrons and also well-known combinatoral problems as knapsack problem, quadratic assignment problem, maximum clique problem.

References


116
Global Optimality Conditions Approach to the Maximum Clique Problem

Antonina Kuznetsova
Institute of System Dynamics and Control Theory SB of RAS
Lermontov Str., 134
Irkutsk-33, 664033, Russia
Email: kuznet@icc.ru

Alexander Strekalovsky
Institute of System Dynamics and Control Theory SB of RAS
Lermontov Str., 134
Irkutsk-33, 664033, (Russia)
Email: strekal@icc.ru

We consider the maximum clique problem (MCP), the well-known combinatorial optimization problem which is estimated by some scientists as intractable [3]. For solving MCP we apply an algorithm based upon Global Optimality Conditions (GOC).

Let $G = (V, E)$ be undirected graph where $V = \{1, \ldots, n\}$ denotes the set of vertices (nodes), and $E$ denotes the set of edges. The general assumption is that $G$ has no parallel edges and no self-loops. Denote by $(i, j)$ an edge joining vertex $i$ and vertex $j$.

A subset $C$ of vertices is called clique if every pair of vertices in $C$ is joined by an edge.

The maximum clique problem is the problem of finding a clique $C$ of maximum size (cardinality). Following Motzkin and Straus [2] the MCP can be stated as indefinite quadratic programming problem:

$$F_G(x) = \sum_{(i,j) \in E} x_i x_j = \frac{1}{2} \langle x, A_G x \rangle \uparrow \text{max},$$

$$x \in S = \{ x = (x_1, \ldots, x_n)^T : \sum_{i=1}^n x_i = 1, x_i \geq 0 \text{ for } i = 1 \ldots n \},$$

(1)

where $A_G$ is the adjacency matrix of $G$.

It is well-known, that if $\alpha = \max(F_G, S)$, then $G$ has a maximum clique $C$ of cardinality $k = \frac{1}{2} \alpha$. This maximum can be attained by setting $x_i = \frac{1}{k}$ if $i \in C$ and $x_i = 0$ if $i \notin C$.

Nevertheless, the global solution of Problem (1) is not directly connected to a maximum clique. Therefore I. Bomze [1] gave a regularization of Problem (1)
replacing the objective function in (1) by quadratic function with the matrix \( A \triangleq A_G + \frac{1}{2} I_n \). Then a global solution \( x^* \) to the regularized problem:

\[
F_A(x) = \frac{1}{2} (x, Ax) \uparrow \max, \ x \in S; \quad (P)
\]

allows us to define immediately corresponding maximum clique as follows:

\[
C = \{i \in V : x^{*}_i > 0\}.
\]

The problem (P) can be considered as a instance of the following d. c. minimization problem.

\[
F(x) = g(x) - f(x) \downarrow \min, \ x \in D; \quad (2)
\]

where \( g, f \) are convex functions and \( D \) is a convex set in \( R^n \).

We begin the solving the problem (P) by advancing a characterization of a global solution to (2) by means of Global Optimality Conditions (GOC).

**theorem 1** [4]

If \( z \in D \) is a global solution to Problem (2). Then

\[
\forall (y, \beta) : \beta - f(y) = \zeta \triangleq g(z) - f(z), \quad \begin{cases} 
\forall y \in D, \ g(y) \leq \beta \leq \sup(g, D), \\
g(x) - \beta \geq \langle \nabla f(y), x - y \rangle \ \forall x \in D.
\end{cases} \quad (E)
\]

If in addition the following condition takes place

\[
\exists v \in D : F(v) > F(z), \quad (H)
\]

then (E) becomes sufficient for \( z \in D \) being a global solution to (2).

GOC (E) possesses so-called Algorithmic Property (AP), i.e. if (E) is violated there then exists a machinery allowing to construct a better feasible point. Actually, if there is a triple \((y, \beta, \hat{x})\) verifying

\[
\beta - f(y) = \zeta, \ y \in D, \ g(y) \leq \beta \leq \sup(g, D),
\]

but the inequality in (E) is browken down, i.e \( \exists \hat{x} \in D : \)

\[
g(\hat{x}) - \beta < \langle \nabla f(\hat{y}), \hat{x} - \hat{y} \rangle;
\]

we immediately obtain due to convexity of \( f(\cdot) \)

\[
F(\hat{x}) = g(\hat{x}) - f(\hat{x}) < \beta - f(\hat{y}) = F(z),
\]

that means, \( \hat{x} \) is better than \( z \).

Exploiting this AP we get the Global Search Algorithm for (2). Let us describe a global search algorithm step by step.

Let be given an initial feasible point \( x^0 \in D \) and a numerical sequence \( \{\varepsilon_k\} : \varepsilon_k > 0, \varepsilon_k \downarrow 0 (k \rightarrow \infty) \).
Step 1. Starting at the point $x^k \in D$ obtain a stationary (critical) point $z^k \in D$
by means of a local search method for (2).

Step 2. Choose a number $\beta \in [m, M]$, where $m \triangleq \inf(g, D), M \triangleq \sup(g, D)$.
To begin with, one may set $\beta_0 := g(z^k)$.

Step 3. Construct an approximation

$$R_k = \{y^1, \ldots, y^N \mid f(y^i) = \beta - \zeta_k\}, \quad \zeta_k := F(z^k).$$

Step 4. For $i = 1, \ldots, N$ solve Linearized Problem

$$g(x) - \langle \nabla f(y^i), x \rangle, \; x \in D.$$

Let $u^i$ be an $\varepsilon_k$-solution to $(PL_i)$.

Step 5. For $i = 1, \ldots, N$ find point $w^i$, $f(w^i) = \beta - \zeta_k,$

$$\langle \nabla f(w^i), u^i - w^i \rangle + \varepsilon_k \geq \sup_v \{ \langle \nabla f(v), u^i - v \rangle \mid f(v) = \beta - \zeta_k \}.$$

Step 6. Set

$$\eta_k(\beta) = g(w^i) - \beta - \langle \nabla f(w^i), u^i - w^i \rangle$$

$$= \min_{1 \leq i \leq N} \{ g(u^i) - \beta - \langle \nabla f(u^i), u^i - w^i \rangle \}.$$

Step 7. If $\eta_k(\beta) < 0$, set $x^{k+1} := u^i$ and go to step 1.

Step 8. If $\eta_k(\beta) \geq 0$, then set $\beta := \beta + \Delta \beta$ by means of any one-dimensional
minimization method and loop to step 3.

Step 9. If $\eta_k(\beta) \geq 0 \forall \beta \in [m, M]$, but $\varepsilon_k > \delta$, where $\delta$ is a given
tolerance, then set $x^{k+1} := z^k, k := k + 1$, and go to step 1.

Step 10. If $\eta_k(\beta) \geq 0 \forall \beta \in [m, M]$, and $\varepsilon_k \leq \delta$, then stop.

Then using the structure of Problem (P) we improved the Algorithm which is
called by $(RM)$-algorithm below.

In order to assess the effectiveness of the proposed approach for solving MCP
the $(RM)$-algorithm was tested on DIMACS benchmark graphs.

The code was written in the Borland Pascal and run on a PC Pentium-166.
The experiments were conducted by starting the process from the barycenter
$x^0 = \left(\frac{1}{n}, \ldots, \frac{1}{n}\right)$ of the canonical simplex $S$.

The computational results showed that the approach based on GOC turns out
to be rather successful, competitive and promising.
References


Lower Bounds for Equal Circles Packing in a Square Problem Using the TAMSASS-PECS Stochastic Algorithm

P. G. Szabó and T. Csendes
Department of Applied Informatics
József Attila University
Árpád tér 2.
H-6720 Szeged (Hungary)
Email: {pszabo,csendes}@inf.u-szeged.hu

L. G. Casado and I. García
Department of Computer Architecture and Electronics
University of Almería
La Cañada de San Urbano S/N
04120 Almería (Spain)
Email: leo@miro.ualm.es
inma@monet.ualm.es

The problem to resolve is, how to place $n$ equal and non-overlapping circles into a square (henceforth unit square), in such a way that the $r_i$ radius of the circles will be maximal. An equivalent problem is to locate $n$ points in the square, such that, the $m_n$ minimal distance between the points is maximal. This problem can be formulated as a nonlinear continuous global optimization problem:

$$
\begin{align*}
\mu(\mathbf{\bar{x}}) &= \min_{1 \leq i < j \leq n} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \\
\max \mu(\mathbf{\bar{x}}) &= 2^n, \quad n \in \mathbb{N}\setminus\{0, 1\}
\end{align*}
$$

where $x_i, y_i$ are the coordinates of the $i$-th point. The optimal solution of (1) is known up to $n=27$ using a computer-aided elimination method [1, 2] and for $n = 36$, which was proven with mathematical tools [3]. To prove that a packing is optimal, is not an easy problem. Therefore, when a new better packing is found, at least the new value of $m_n$ is a lower bound of the optimal solution. The better lower bounds and the corresponding packings, not proven to be optimal,
are named good packing. In the packing literature, good packings are known up to \( n = 52 \) and \( n = 54, 55, 56, 60, 61, 62, 72, 78 \) [4, 5]. The earlier approaches to solve this problem were using billiard-simulation, by minimizing an energy function, using a standard quasi-Newton algorithm or a nonlinear programming solver.

In this work, good packings, up to \( n = 100 \), have been found, using repeated patterns and the TAMSASS-PECS (Threshold Accepting Modified Single Stochastic Search for Packing Equal Circles in a Square) algorithm. With this algorithm 45 new good packings (40 unpublished and 5 better packings than those found in the literature (e.g., those in Figure 1)) have been found.

The TAMSASS-PECS algorithm is based on the Threshold Accepting method [6] and MSASS (our simplified version of SASS [7]), trying to maximize the minimum distance between the centers of the circles. Formally, the Threshold Accepting algorithm is very similar to simulating annealing algorithms. In Threshold Accepting we accept every move that leads to a new solution not much worse than the current solution. To be more specific, if \( s \) is the current solution, the proposed next solution \( s' \in \text{Neighbourhood}(s) \) is accepted as the next candidate solution, if

\[
\Delta = f(s') - f(s) < T_h
\]  

(2)

where \( T_h > 0 \) is the so-called threshold level. During the optimization process the threshold level is gradually decreased like the temperature in simulated annealing. In Threshold Accepting it is guaranteed that a move which would make the current solution much worse is never accepted, unlike in simulated annealing. The TAMSASS-PECS algorithm starts with a pseudorandom initial solution (location of the \( n \) points \( s_i, 1 \leq i \leq n \)) by tiling the square. Nevertheless, the initial solution is not crucial if the threshold level is not very small, because large random movements are allowed. The initial value for the threshold level is \( T_h = 0.02 \) and the standard deviation \( \sigma \) is equal to the diameter of
the tiles.

TAMSASS-PECS tries to improve the current solution by an iterative procedure. At every iterative step, the MSASS (See Algorithm 1) subroutine is executed for all the $n$ points, using the same values for both the standard deviation ($\sigma$) and the threshold level ($T_h$). The criterion to stop this iterative procedure is based on the value of the standard deviation, which is decreased by a factor of 0.99 at every iteration. The threshold level is also decreased by the same factor.

**Algorithm 1: Modified Single Agent Stochastic Search**

```
proc MSASS (s, $\sigma_0$, $T_h$, i)
  var sent := 0, fent := 0, Fent := 3, $c_t$ := 0.5, k := 1
  while ($fent < 4Fent \land sent = 0$)
    $\sigma_k$ := \begin{cases} 
      \alpha \cdot \sigma_{k-1} & \text{if } fent > Fent \\
      \sigma_{k-1} & \text{otherwise}
    \end{cases}
    $\xi_k$ ~ $N(0, \sigma_k)$  \hspace{1cm} \text{Gaussian perturbation}
    $s'_{k+1}(i) := s_k(i) + \xi_k$
    if ($f(s') \geq f(s_k(i)) \cdot T_h$)
      $s_{k+1}(i) := s'(i)$
      sent := sent + 1
    else
      $s'(i) := s_k(i) - \xi_k$
      if ($f(s') \geq f(s_k(i)) \cdot T_h$)
        $s_{k+1}(i) := s'(i)$
        sent := sent + 1
      else
        $s_{k+1}(i) := s_k(i)$
        fent := fent + 1
        k := k + 1
  return $s$
```

At every iterative step, MSASS is executed for all the points $n$ in an increasing order, which is determined by the value of the minimum distance from any point to the rest of points $(d_{ij})$. The value of the minimum distance are updated after every execution of MSASS.

With the TAMSASS-PECS algorithm circles packings in another shapes can be obtained, by changing the bounds of the possible movements.

Another method to obtain lower bounds is to use repeated patterns for the initial positions of the center of the circles. In this work we used 5 different patterns to obtain 48 lower bounds for the packings in the $n = 1..100$ cases [8]. Often, the obtained packing is the optimal one. In Figure 2 we can see our lower bounds [9], asymptotic

$$m_n \approx \sqrt{\frac{2}{\sqrt{3n}}}$$

124
and upper bounds

\[ m_n \leq \frac{1 + \sqrt{1 + (n - 1) \frac{\pi}{\sqrt{3}}}}{n - 1} \]

of \( m_n \).

References


Global Optimization Techniques for Optimizing over Efficient Sets

Nguyen V. Thoai
Dept. of Mathematics
University of Trier
54286 Trier, Germany
Email: thoai@uni-trier.de

An important approach in multiple criteria programming is the optimization of some function over the efficient or weakly-efficient set. This is in general a difficult nonconvex optimization problem. In this article we consider multiple objective programming problem of the form

\[
\max \ c^i x \ (i = 1, \ldots, p), \ \text{s.t.} \ x \in X, \quad \text{(MOP)}
\]

where \( X \) is a closed convex subset of \( \mathbb{R}^n \) and \( c^i \in \mathbb{R}^n \setminus \{0\} \) for \( i = 1, \ldots, p \). The vectors \( c^i \) are called criterion vectors of problem (MOP).

Let \( C \) be the \( p \times n \) matrix with rows \( c^1, \ldots, c^p \). A point \( x \in X \) is called an efficient (or nondominated or Pareto-optimal) solution of Problem (MOP), if there is no point \( y \in X \) such that \( Cy \geq Cx \) and \( Cy \neq Cx \). A point \( x \in X \) is called a weakly-efficient solution of Problem (MOP), if there is no point \( y \in X \) such that \( Cy > Cx \). Denote by \( E_X \) and \( W_X \) the sets of all efficient solutions and all weakly-efficient solutions of Problem (MOP), respectively, and let \( f \) be a real-valued function on \( \mathbb{R}^n \). We consider the optimization problems

\[
\max \ \{ f(x) : x \in E_X \}, \quad \text{(P1)}
\]
\[
\max \ \{ f(x) : x \in W_X \}. \quad \text{(P2)}
\]

Following assumptions are needed:

(A1) \( f : \mathbb{R}^n \to \mathbb{R} \) is a concave function or a composite function \( f(x) = \varphi(c^1 x, \ldots, c^p x) \) with \( \varphi : \mathbb{R}^p \to \mathbb{R} \) being a convex function.

(A2) The (convex polyhedral) criterion cone of (MOP) is generated by the vectors \( c^1, \ldots, c^k \), \( (k \leq p) \), i.e., each vector \( c^i \) \( (i = k+1, \ldots, p) \) can be described as

\[
c^i = \sum_{j=1}^{k} \lambda_j c^j, \ \lambda_j \geq 0 \ (j = 1, \ldots, k). \quad \text{(1)}
\]
The vectors $c^1, \ldots, c^k$ in Assumption (A2) are usually called extreme criterion vectors.

Let

$$Z = \{ z \in \mathbb{R}^k : z_i = c^i x \ (i = 1, \ldots, k), \ x \in X \},$$

and, for each $z \in \mathbb{R}^k$, define two subsets $L(z)$ and $L^+(z)$ of $\mathbb{R}^k$ by

$$L(z) = \{ v \in \mathbb{R}^k : z - v \leq 0, \ z \in Z \}.$$  \hfill (3)

$$L^+(z) = \{ v \in \mathbb{R}^k : z - v < 0, \ z \in Z \}.$$ \hfill (4)

The following result shows the relationship between efficient (weakly-efficient) solutions of Problem (MOP) and the sets defined in (3)-(4).

**Proposition 1.** If $x^*$ is an efficient (weakly-efficient) solution of Problem (MOP) and $z^*_i = c^i x^* \ (i = 1, \ldots, k)$, then $|L(z^*)| = 1$, i.e., $L(z^*) = \{ z^* \}$, $|L^+(z^*)| = 0$, i.e., $L^+(z^*) = \emptyset$. Conversely, if $z^* \in Z$ satisfies $|L(z^*)| = 1$, then each point $x^* \in X$ satisfying $z^*_i = c^i x^* \ (i = 1, \ldots, k)$ is an efficient (weakly-efficient) solution of Problem (MOP).

In view of Proposition 1, in what follows, we shall denote by $C$ the $k \times n$ matrix with rows $c^1, \ldots, c^k$.

In order to construct the master problem, which is equivalent to (P1)-(P2), we define the following sets.

$$E_Z = \{ z \in Z : |L(z)| = 1 \}$$

$$W_Z = \{ z \in Z : |L^+(z)| = 0 \}$$

$$G = \{ z \in \mathbb{R}^k : z \leq v, \ v \in Z \}.$$

**Proposition 2.** (i) If $Z \neq \emptyset$, then $\text{int} G \neq \emptyset$.

(ii) If $E_Z \neq \emptyset$, then $E_Z \subseteq W_Z = \partial G \cap \partial Z$.

(As usual, for a set $S$, we denote by $\text{int} S$ and $\partial S$ the interior and the boundary of $S$, respectively).

Next, let $T_1 \subset \mathbb{R}^{k+1}$ and $T_2 \subset \mathbb{R}^{k+1}$ defined by

$$T_1 = \{ (z, t) \in \mathbb{R}^{k+1} : z \in \partial Z \cap \partial G, \ t \leq f(z) \},$$

$$T_2 = \{ z \in \mathbb{R}^k : z \in \partial Z \cap \partial G \}.$$

For the case where the function $f$ is concave consider the following problem.

$$\max \{ t : (z, t) \in T_1 \}. \hfill (Q_1)$$

If $f$ is a composite function $f(x) = \varphi(c^1 x, \ldots, c^k x)$ with $\varphi : \mathbb{R}^k \to \mathbb{R}$ being a convex function, then consider the problem

$$\max \{ \varphi(z) : z \in T_2 \}. \hfill (Q_2)$$

The following result shows the equivalence between the problems $(Q_1)$ or $(Q_2)$, respectively, and Problems (P1), (P2).
Proposition 3. (i) Assume that the function $f$ is concave. If $x^*$ is an optimal solution of $(P2)$, then $(z^*, t^*)$ with $z^* = Cx^*$, $t^* = f(x^*)$ is an optimal solution of $(Q_1)$. Conversely, if $(z^*, t^*)$ is an optimal solution of $(Q_1)$, then every point $x^* \in X$ satisfying $z^* = Cx^*$ is an optimal solution of $(P2)$ with the optimal value $f(x^*) = t^*$. If, in addition, $z^* \in E_Z$, then $x^*$ is optimal to $(P1)$ as well.

(ii) Assume that $f$ is a composite function $f(x) = \varphi(c^1 x, \cdots, c^k x)$ with $\varphi : \mathbb{R}^k \to \mathbb{R}$ being a convex function. If $x^*$ is an optimal solution of $(P2)$, then $z^*$ with $z^* = Cx^*$ is an optimal solution of $(Q_2)$. Conversely, if $z^*$ is an optimal solution of $(Q_2)$, then every point $x^* \in X$ satisfying $z^* = Cx^*$ is an optimal solution of $(P2)$. If, in addition, $z^* \in E_Z$, then $x^*$ is optimal to $(P1)$ as well.

Based on a conical branch and bound algorithm in global optimization, we establish convergent algorithms for solving Problems $(Q_1)$ and $(Q_2)$. In particular, for the case $k = 2$ we obtain immediately very simple and efficient algorithms.
Global Optimisation
Applied to Molecular Architecture

Graham Wood
Institute of Information Sciences and Technology
Massey University
Palmerston North
Private Bag 11 222
New Zealand
Email: g.r.wood@massey.ac.nz

Wayne Pullan
Information Technology Division, DSTO
PO Box 1500
Salisbury SA, 5108
Australia
Email: wayne.pullan@dsto.defence.gov.au

This session will address the problem of identifying molecular structures which correspond to the globally minimum potential energy. Molecular structures arise as a result of non-bonded and bonded atomic interactions and experimental evidence shows that, in the great majority of cases, the potential energy global minimum corresponds to the most stable configuration of the molecular structure. This configuration is of particular importance as it dictates most of the physical properties of the molecular structure.

The potential energy of a molecular structure may be calculated, as a function of the atomic positions, using appropriate molecular models. These give rise to potential energy functions that are typically non-convex with many local minima, so that finding the global minima is extremely difficult. For many years this problem has been investigated by chemists and physicists; in more recent years, research specialists in optimisation and computer science have become involved. The minimisation of non-convex potential energy functions arising from molecular conformation or protein folding problems has become a significant interdisciplinary problem.

This session describes a molecular structure global optimisation method using both deterministic local and stochastic global optimisation techniques within a genetic algorithm environment. By incorporating different genetic operators, the one basic method was able to globally optimise a number of different types
of molecular structure. Of particular interest is that, in determining these global minima, the method always started from randomly generated initial configurations and at no stage used any heuristic information to accelerate the search.

From an experimental point of view, the method was successful and found, at the time of the investigation:

- All currently accepted global minima for scaled Lennard-Jones atomic clusters of 2 to 80 atoms.
- Two new global minima for 77 and 78 atom scaled Lennard-Jones atomic clusters.
- All currently accepted and some improved global minima for mixed argon-xenon atomic clusters of 7, 13 and 19 atoms. In addition, minima were determined for all remaining clusters in the 2, ..., 20 atom range.
- All currently accepted global minima for clusters of benzene molecules of 2 to 6 molecules and new minima for clusters of 8 to 12 molecules.
- All currently accepted global minima for a two-dimensional model molecular structure where the number of atoms ranged from 3 to 42.
- Currently accepted global minima for a number of small molecules.

In this session, the problem will be formulated, the mechanism of the algorithm will be described and the application of the algorithm to mixed argon-xenon atomic clusters and benzene clusters will be illustrated.
Theoretical and Computational Insight into Parallel Global Optimization Using Random Search *

Zelda B. Zabinsky
Wen Luo
Industrial Engineering Program
University of Washington
Seattle, Washington (USA) 98195-2650
Email: zelda@u.washington.edu
Email: wendyl@u.washington.edu

Mark E. Tuttle
Mechanical Engineering
University of Washington
Seattle, Washington (USA) 98195-2600
Email: tuttle@u.washington.edu

Objectives

Dramatic increases in computer processing speeds are expected to be realized using parallel processors. In fact, supercomputer facilities already offer subscribers the ability to use as many as 128 parallel processors, and desktop workstations featuring two to four parallel processors are now commercially available. It is expected that parallel computing will have a tremendous impact on optimization [4, 6]. Computing speed can be increased drastically by parallelization on multiprocessor machines. It is possible to solve problems, within real time constraints, that serial machines could not handle, even though the computational cost is extremely high.

Now we are facing the situation of how to exploit the capabilities of modern parallel processors, and there is a need to develop parallel global optimization algorithms. Several questions arise on how to make full use of the processing capability. Is it more effective to parallelize the optimization algorithm and execute the function evaluation on a single processor, or should we parallelize the function evaluation and maintain a sequential optimization algorithm? In

---

*This research has been supported, in part, by a grant from the National Science Foundation (NSF), DMI-96224/33, and parallel computing resources from the National Partnership for Advanced Computational Infrastructure (NPACI).
parallel computing, when is the proper time to halt computation and communicate among processors to synchronize the application? Is the effectiveness and efficiency of parallelization dependent on a specific algorithm and/or function evaluation?

Our objective is to develop different parallelization approaches for the global optimization algorithm *Improving Hit-and-Run* (IHR), and to conduct computational experiments on supercomputers. We also seek a theoretical understanding that could explain the performance of parallel algorithms in a parallel computing environment. This will give us guidance to measure, analyze, design and improve global optimization algorithms which are efficient and effective on massive parallel computers.

**Parallel Random Search Algorithms**

Random search algorithms offer considerable promise as efficient optimization methods for a large class of problems. *Improving Hit-and-Run* [7, 12] generates a random direction and then a uniform random point in that direction. Only improving points are accepted in the sequence.

We developed three schemes of parallel IHR that are implemented on IBM SP supercomputers, which were provided by the National Partnership for Advanced Computation Infrastructure. All three schemes are data-parallel programs implemented on a single-instruction multiple-data parallel system [5]. The three parallelized versions of IHR, assuming \( p \) processors are available, are as follows:

- **IHR Replication**: replicate \( p \) sequential IHR runs independently over the original feasible domain.
- **Domain Partition**: partition the original feasible domain into \( p \) subdomains, and run one sequential IHR over each sub-domain.
- **Direction Replication**: generate \( p \) random directions from the current point, and sample one point on each direction, then update current point with the best improving point.

**Computational Experiments with Parallel IHR**

Two test functions, the sinusoidal function [10] and the Hartman function [9], both with known global minimum, are used in experiments with 8 and 16 processors. We also applied parallel IHR with 16 processors on a real optimization problem: minimizing the potential energy function for buckled composite plates [8], whose global minimum is unknown. For the test functions, we compared the success rate of IHR schemes, and the number of iterations needed by those successful runs for each scheme. For the composite plate buckling problem, we compared the best energies found in a fixed number of iterations, for 23 and 46 dimensional potential energy functions.

The experimental results display the effectiveness and efficiency of parallel IHR by parallelizing the optimization algorithm and executing the function...
evaluation on a single processor. In comparison with sequential IHR, parallelization of IHR greatly improves the success rate on the test functions. IHR replication has the highest success rate on the sinusoidal and Hartman functions. Domain partition is the second to IHR replication in terms of success rate on the test functions, but is the best scheme on the 23-dimensional potential energy function. The 46-dimensional potential energy function is difficult for all schemes. The large increase in success rate and decrease in number of iterations provide strong evidence that it is worthwhile and beneficial to develop parallel optimization algorithms.

Theoretical Analysis on Parallel PRS

Pure Random Search (PRS) is a simple random search algorithm which generates a sequence of independent, uniformly distributed points in the feasible region and returns the best point as an approximation to the optimum [1, 2, 3]. A great merit of pure random search is that its simplicity lends itself to theoretical analysis [11]. We analyze parallel replication and domain partition of PRS to draw analogies to the IHR replication and domain partition schemes.

<table>
<thead>
<tr>
<th>PRS Scheme</th>
<th>$P$</th>
<th>$E$</th>
<th>$F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential PRS</td>
<td>$P_S = \frac{1}{n!}$</td>
<td>$E_S = \frac{1}{n!}$</td>
<td>$F_S = \frac{1}{n!}$</td>
</tr>
<tr>
<td>PRS Replication</td>
<td>$P_A \approx \frac{1}{p\cdot n!}$</td>
<td>$E_A \approx \frac{1}{p\cdot n!}$</td>
<td>$F_A \approx \frac{1}{p\cdot n!}$</td>
</tr>
<tr>
<td>Domain Partition</td>
<td>$P_B = \frac{1}{p\cdot n!}$</td>
<td>$E_B = \frac{1}{p\cdot n!}$</td>
<td>$F_B = \frac{1}{p\cdot n!}$</td>
</tr>
</tbody>
</table>

$n$: Number of variables in the objective function.
$p$: Number of processors used in parallelization.
$P$: Probability of sampling within $\varepsilon$-region in one iteration.
$E$: Expected number of iterations.
$F$: Expected number of function evaluations.

Table 1: Theoretical Performance Comparison for PRS

The performance analysis on PRS is summarized in table 1. Based on uniform sampling, we see that by running $p$ pure random searches concurrently, either by PRS replication or domain partition, the probability of success in one iteration is nearly or exactly $p$ times the probability of success for sequential PRS. It is exciting that the success rate has a linear speedup, which is ideal in parallelism.

The expected number of iterations gains linear speedup as well, although the total amount of function evaluations is the same for parallel and sequential PRS. This implies that, even though we cannot expect to reduce the total amount of computation to solve the global optimization problem, we can easily distribute the computation among $p$ processes and achieve linear speedup in terms of the expected number of iterations.

We are currently investigating the implications of the theoretical analysis on PRS with the computational experiments on IHR. For computationally expensive objective functions, it may be very difficult to achieve linear speedup by
parallelizing the function evaluation. This research suggests that linear speedup may be more easily attained by parallelizing the global optimization schemes.

References

Statistical Inference in Global Random Search
Based on Observations at Random and
Semi-Random Points

Anatoly Zhigljavsky
School of Mathematics
Cardiff University
Senghenydd Rd
Cardiff, CF2 4YH, UK
Email: ZhigljavskyAA@cardiff.ac.uk

Our overall aim is increase of efficiency of statistical inferences in a general class of random search algorithms that include so-called branch and probability bound methods and some versions of the genetic algorithms. This is achieved by means of a certain reduction of randomness of test points which makes statistical inference about the distribution of the objective function values more efficient than for simple random arrangements of test points.

Let $\mathcal{X}$ be a compact subset of $\mathbb{R}^d$ and $f$ be the objective function given on $\mathcal{X}$. The maximum of $f$, $M = \max_{x \in \mathcal{X}} f(x)$ and the corresponding maximizer $x^*$ are to be found (approximated).

Let $X_N = \{x_1, \ldots, x_N\}$ be a sample of random points in $\mathcal{X}$. It is well-known that any properly organized stratified sampling procedure is superior to the independent sampling with respect to the variance of Monte Carlo estimates of integrals of functions in $L_2(\mathcal{X})$. Based on the results obtained in [1-3], we establish similar results for certain performance characteristics important in global optimization. We demonstrate that the stratified sample with the maximum stratification is optimum, in a suitable sense. For Lipchitz functions, we also derive some results concerning the efficiency of the random 'tabu search', or the 'progressive random search', in the terminology of L.Devroye [4].

We consider two related performance characteristics: (i) the cumulative distribution function (c.d.f.) of the record value $f^*_N = \max_{x \in X_N} f(x)$ achieved at the sample points, and (ii) $\beta$-th moment of the difference $M - f^*_N$, for every $\beta > 0$. As a consequence, we prove that efficiency of statistical inferences about $M$ is higher when one uses a stratified sample rather than an independent one.

Many global random search algorithms consist of several iterations, at each one for various subsets $Z$ of $\mathcal{X}$, there are samples of points (not necessarily independent) distributed according to certain probability measure $P_Z$ in $Z$. Values of $f$ at these points in $Z$ constitute a sample from the distribution with
the c.d.f.,

$$F_Z(t) = P_Z \{ z \in Z : f(z) < t \}$$

and the upper endpoint $M_Z = \text{esssup}_{z \in Z} f(z)$. In order to make a decision about whether to locate new points in $Z$ we can draw statistical inference concerning the parameter $M_Z$ and the behavior of the c.d.f. $F_Z(t)$ in the vicinity of this point, that is, for $t$ such that $F_Z(t) \approx 1$. This behavior is closely related to the behavior of $f$ in the vicinity of the maximizer $x_Z^*$ of $f$ in $Z$.

The statistical inference drawn for the case $Z = X$ can serve for construction of stopping rules based on a precision control for global random search algorithms. The algorithms based on statistical inference about $M_Z$ for various $Z \subset X$ are called the branch and probability bound methods, see [1].

Since for all sets $Z$ the statistical procedures can be constructed in a similar manner we only consider the case when $M = M_X = \sup f$,

$$F(t) = F_X(t) = \Pr \{ x \in X : f(x) < t \} = \int_{f(x) < t} P(dx)$$

and a sample $X_N = \{ x_1, \ldots, x_N \}$ is given from a distribution $P$ on $X$.

Statistical inference considered in this work are semiparametric in the sense that they are based on a particular semiparametric representation for the c.d.f. $F(t)$ in the vicinity of $M$. This representation is due to the well-known result in the asymptotic theory of the extreme order statistics [5] according to which

$$\lim_{N \to \infty} F^N(M + \theta_N z) = \Psi_{\alpha}(z)$$

for all $z < 0$ where $\theta_N$ is the $(1 - \frac{1}{N})$-quantile of the c.d.f. $F(t)$ and $\Psi_{\alpha}(z)$ is the c.d.f. $\Psi_{\alpha}(z) = \exp((-z)\alpha)$, $z < 0$, under the assumption that the function $V(v) = 1 - F(M + \frac{1}{v})$, $v > 0$, regularly varies at infinity with some exponent $(-\alpha)$. Note that this assumption holds for rather general classes of c.d.f. $F(t)$ with a finite upper endpoint $M$. In particular, it holds if the representation

$$F(t) = 1 - c(M - t)^{\alpha} + O((M - t)^{\alpha}), \quad t \uparrow M,$$

is valid for some positive constants $c$ and $\alpha$. As shown in [1] and [6], in typical problems of global random search this condition holds with the value $\alpha = d/2$.

Let $X$ be divided onto $m$ disjoint subsets $X_j$ ($j = 1, \ldots, m$) of equal $P$-measure and let there be given $l$ $P$-distributed points $x_{ji}, x_{ji}$ in each subset $X_j$. Then the sample

$$X_{m,l} = \{ x_{ji}, j = 1, \ldots, m, i = 1, \ldots, l \}$$

of size $N = ml$ is called stratified. In case $m = 1$ the sample is independent. Considering the asymptotics we assume that the number of stratifications $m$ tends to infinity but the number of points $l$ in each subset remains fixed.

Denote the sample of the objective function values by $Y = (y_1, \ldots, y_N)$ and by $\eta_1 \leq \ldots \leq \eta_N$ the corresponding order statistics. (In the present case $Y = (f(x_{ji}), j = 1, \ldots, m, i = 1, \ldots, l)$). Let us fix $k+1$, the number of order statistics to
be used for statistical inferences about $M$ and let $\eta_{N,k} = (\eta_N, \ldots, \eta_{N-k})'$ be the corresponding vector of these statistics. For an arbitrary vector $a = (a_0, \ldots, a_k)'$ we can define $a'\eta_{N,k}$ a linear estimator of $M$. For this estimator to be asymptotically unbiased, we need $\sum_{i=0}^{k} a_i = 1$. For an explicit formulas of the optimal asymptotically unbiased linear estimates see [1,2]. Note also that according to [1] the optimal linear estimators of $M$ are as efficient as the maximal likelihood estimates.

The results of [1-3] imply that under some regularity conditions concerning $f$, the limiting distribution of the sequence of random variables $(\eta_{N-M})/(M-\theta_N)$ has the c.d.f. $\Psi_{\alpha,l}(u) = (1 - \frac{\alpha}{2\pi})/(-i/\alpha < u < 0)$ when $N = ml \to \infty$, $l = \text{const}$. This yields that for every $k, 0 < k < l$ and for every asymptotically unbiased estimator of $M$ we have $E(M - a'\eta_{N,k}) \sim (M - \theta_N) a'b P_l(1/\alpha) / \Gamma(i + 1) \cdot (i+1) \cdot (i+2) / \Gamma(i+2)$.

Since $P_l(u) < 1$ for $l > 1$ and $u > 0$ then the stratified sample provides more accurate estimates of $M$ than the independent sample. Minimisation of $P_l(1/\alpha)$ with respect to $l$ shows that $l = 1$ is optimum. From the other side, in the limiting case $l = \infty$ we get the efficiency of the estimators exactly as in the case of the independent sample $X_N = X_{1,N}$.

References


Global Optimization by Means of Select and Clone

Antanas Zilinskas
Institute of Mathematics and Informatics
Vytautas Magnus University
Akademijos str. 4, Vilnius LT-2600, Lithuania
Email: antanasz@ktl.mii.lt

Julius Zilinskas
Discrete Signal Processing Laboratory
Kaunas University of Technology
Kaunas, Lithuania

The first statistical model used for the global optimization was Wiener process. Several one-dimensional algorithms were developed using Wiener or Wiener related models. The extensive testing has shown that the implemented algorithms favorably compete with the algorithms of other approaches. In the present paper a statistical model for global optimization is constructed generalizing some properties of Wiener process to the multidimensional case. The conditional average of the constructed model is piecewise linear function, and conditional variance is piecewise quadratic function.

A new approach, which is similar to the Branch and Bound approach, is proposed to the construction of the algorithms based on the statistical models. Known implementations of algorithms based on statistical models are reduced to the optimization of an auxiliary function which defines rationality of search, e.g., probability to exceed an aspiration level or expected increment in the next iteration. Such an implementation requires to solve a difficult auxiliary problem. Similar difficulty arises in Lipschitz optimization while implementing algorithms of Pjavnitski-Shubert type: an auxiliary problem of minimization of multimodal lower bound function should be solved. In Lipschitz optimization Branch and Bound algorithms have been implemented to avoid an auxiliary multimodal problem. Normally a feasible region initially is covered by boxes. The cover is refined by selecting the box with minimal bound estimate, and by subdividing the selected box into smaller boxes. Tremendous reduction of auxiliary computations necessary to implement an algorithm may be achieved in this way. The idea related to the branch and bound approach is proposed in the present paper. The search is started by the initial covering of the feasible region by the standard subsets (correct simplexes). The advantages of simplex
based covering with respect to a proposed statistical model are discussed. The search means refinement of a current cover at each iteration. The selection of the candidate for cloning is justified by the statistical model. The cloning of a selected simplex means its subdivision into smaller simplexes. The rationality of the search is aimed by means of optimal selection. Cloning facilitates a rather simple implementation. Although the proposed approach is similar to B&B, nevertheless there are essential differences: no bounds are calculated and subdivision is rather specific.

Two dimensional version of the proposed algorithm is implemented, and the testing results are presented. The comparison of efficiency of the two-dimensional versions of select and clone algorithm and of Lipschitz B&B algorithm implies similar conclusions as the comparison of efficiency of the one-dimensional Wiener model based P-algorithm and the Pjavski-Shubert algorithm.
The phase unwrapping problem as a global optimization problem

Francesco Zirilli
Dipartimento di Matematica G.Castelnuovo
Università di Roma La Sapienza
00185 Roma Italy
Email: f.zirilli@caspur.it

Phase unwrapping is the key problem in building the elevation map of a scene from interferometric synthetic aperture radar (SAR) system data. Phase unwrapping consists in the reconstruction of the phase difference of the radiation received by two SAR systems as a function of the azimuth and slant range coordinates. The data available to reconstruct the phase difference are a measure of the phase difference modulo 2\pi. This problem has been approached as an optimization problem by several authors, see for example [1], [2]. Due to the great variety of possible scenes (for example urban scenes or agricultural scenes) to reconstruct and to the huge number of independent variables involved in a scene of reasonable size the resulting optimization problem is an interesting one that may be formulated in many different ways. We discuss formulations such as a global optimization problem in continuous optimization or in combinatorial optimization. We present phase reconstructions obtained with the methods proposed using synthetic data and real data obtained from the ERS mission of the European Space Agency.

References


## Index

<table>
<thead>
<tr>
<th>Name</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blanquero</td>
<td>1</td>
</tr>
<tr>
<td>Boer</td>
<td>3</td>
</tr>
<tr>
<td>Borze</td>
<td>7</td>
</tr>
<tr>
<td>Bulatov</td>
<td>9</td>
</tr>
<tr>
<td>Calvin</td>
<td>14, 16</td>
</tr>
<tr>
<td>Cardoso</td>
<td>18</td>
</tr>
<tr>
<td>Carrizosa</td>
<td>1</td>
</tr>
<tr>
<td>Casado</td>
<td>21, 122</td>
</tr>
<tr>
<td>Chehayeb</td>
<td>49</td>
</tr>
<tr>
<td>Csallner</td>
<td>24</td>
</tr>
<tr>
<td>Csendes</td>
<td>26, 122</td>
</tr>
<tr>
<td>Dür</td>
<td>7, 34, 65</td>
</tr>
<tr>
<td>de Klerk</td>
<td>7</td>
</tr>
<tr>
<td>Doye</td>
<td>30</td>
</tr>
<tr>
<td>Draper</td>
<td>41</td>
</tr>
<tr>
<td>Ershov</td>
<td>36</td>
</tr>
<tr>
<td>Feichtinger</td>
<td>38</td>
</tr>
<tr>
<td>Festa</td>
<td>39</td>
</tr>
<tr>
<td>Floudas</td>
<td>40</td>
</tr>
<tr>
<td>Fouskakis</td>
<td>41</td>
</tr>
<tr>
<td>García</td>
<td>21, 58, 97, 122</td>
</tr>
<tr>
<td>Gaviano</td>
<td>84</td>
</tr>
<tr>
<td>Giraud</td>
<td>49</td>
</tr>
<tr>
<td>Gutmann</td>
<td>44</td>
</tr>
<tr>
<td>Hamma</td>
<td>47, 49</td>
</tr>
<tr>
<td>Hart</td>
<td>52</td>
</tr>
<tr>
<td>Hendrix</td>
<td>3, 58</td>
</tr>
<tr>
<td>Hichert</td>
<td>62</td>
</tr>
<tr>
<td>Horst</td>
<td>65</td>
</tr>
<tr>
<td>Jelasity</td>
<td>97</td>
</tr>
<tr>
<td>Jimenez</td>
<td>49</td>
</tr>
<tr>
<td>Kallrath</td>
<td>66</td>
</tr>
<tr>
<td>Kampe</td>
<td>62</td>
</tr>
<tr>
<td>Kanemitsu</td>
<td>70</td>
</tr>
<tr>
<td>Khainskov</td>
<td>36, 77</td>
</tr>
<tr>
<td>Komiya</td>
<td>70</td>
</tr>
<tr>
<td>Konno</td>
<td>70</td>
</tr>
<tr>
<td>Kuznetsova</td>
<td>118</td>
</tr>
<tr>
<td>Le Thi</td>
<td>86</td>
</tr>
<tr>
<td>Leary</td>
<td>82</td>
</tr>
<tr>
<td>Lera</td>
<td>84</td>
</tr>
<tr>
<td>Locatelli</td>
<td>88, 103</td>
</tr>
<tr>
<td>Luo</td>
<td>132</td>
</tr>
<tr>
<td>Meyer</td>
<td>91</td>
</tr>
<tr>
<td>Nowak</td>
<td>95</td>
</tr>
<tr>
<td>Ortigosa</td>
<td>58, 97</td>
</tr>
<tr>
<td>Pardalos</td>
<td>39, 101</td>
</tr>
<tr>
<td>Pinter</td>
<td>102</td>
</tr>
<tr>
<td>Pullan</td>
<td>130</td>
</tr>
<tr>
<td>Quist</td>
<td>7</td>
</tr>
<tr>
<td>Raber</td>
<td>88</td>
</tr>
<tr>
<td>Roos</td>
<td>7</td>
</tr>
<tr>
<td>Schoen</td>
<td>103</td>
</tr>
<tr>
<td>Sergeyev</td>
<td>106</td>
</tr>
<tr>
<td>Shimba</td>
<td>70</td>
</tr>
<tr>
<td>Souza de Cursi</td>
<td>109</td>
</tr>
<tr>
<td>Strelakovsky</td>
<td>114, 118</td>
</tr>
<tr>
<td>Szabó</td>
<td>122</td>
</tr>
<tr>
<td>Terflaky</td>
<td>7</td>
</tr>
<tr>
<td>Thoai</td>
<td>65, 127</td>
</tr>
<tr>
<td>Tuttle</td>
<td>132</td>
</tr>
<tr>
<td>Wood</td>
<td>130</td>
</tr>
<tr>
<td>Zabinsky</td>
<td>132</td>
</tr>
<tr>
<td>Zhigljavsky</td>
<td>136</td>
</tr>
<tr>
<td>Zilinskas</td>
<td>14, 139</td>
</tr>
<tr>
<td>Zirilli</td>
<td>141</td>
</tr>
</tbody>
</table>