An International Journal
computers & mathematics with applications

PERGAMON Computers and Mathematics with Applications 44 (2002) 943–955
www.elsevier.com/locate/camwa

Subdivision, Sampling, and Initialization Strategies for Simplical Branch and Bound in Global Optimization

J. CLAUSEN
Department of Mathematical Modelling
Building 321, Technical University of Denmark
DK 2800 Lyngby, Denmark
jc@imm.dtu.dk

A. ŽILINSKAS
Institute of Mathematics and Informatics
VMU, Vilnius, Lithuania
antanasz@klt.mii.lt

Abstract—We consider the problem of optimizing a Lipshitzian function. The branch and bound technique is a well-known solution method, and the key components for this are the subdivision scheme, the bound calculation scheme, and the initialization. For Lipschitzian optimization, the bound calculations are based on the sampling of function values.

We propose a branch and bound algorithm based on regular simplexes. Initially, the domain in question is covered with regular simplexes, and our subdivision scheme maintains this property. The bound calculation becomes both simple and efficient, and we describe two schemes for sampling points of the function: midpoint sampling and vertex sampling.

The convergence of the algorithm is proved, and numerical results are presented for the two dimensional case, for which also a special initial covering is presented.

Keywords—Global optimization, Branch and bound, Simplical decomposition.

1. INTRODUCTION

One of the most general assumptions enabling the construction of algorithms for global optimization guaranteeing prescribed accuracy is the assumption that an objective function has bounded rate of changing and is defined over a bounded domain. In the following, we restrict our attention to functions of this type. Hence, we assume that $f(x)$, $x \in A \subset R^n$ is Lipshitzian with constant L, and that A is bounded, i.e., that $|f(x) - f(y)| \leq L\|x - y\|$, $x, y \in A$, and that a B exists such that $\|x\| \leq B < \infty$, $x \in A$, where $\|\cdot\|$ denotes the Euclidean norm.

It is then obvious that if $\bigcup_{i=1}^{N} \mathrm{Sph}(x_i, \epsilon/L) \supseteq A$ with $\mathrm{Sph}(x, r)$ denoting the sphere with center x and radius r, then $\min_{1\leq i\leq N} f(x_i)$ estimates the global minimum with accuracy ϵ. Although $N(\epsilon)$, the number of points needed for the optimal algorithm to guarantee accuracy ϵ, is an exponential function of the dimension of the solution space, n, this poses no more pessimism regarding solution methods than the exponential number of feasible solutions in the

0898-1221/02/$ - see front matter Typeset by $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX
PII: S0898-1221(02)00205-5

case of an NP complete combinatorial problem. Algorithms based on the Lipshitzian model have been investigated by many authors: Baritompa, Evtushenko, Galperin, Hansen, Hansen, Horst, Jaumard, Pinter, Pardalos, Tuy, Wood, and others. A recent review of results with references to the original papers is presented in [1]. Our research was motivated by the following observations.

- In practice, there exist problems of modest dimensionality whose solution with guaranteed accuracy is important at the cost of any reasonable time.
- Normally, the objective functions are very different from the saw-tooth worst case functions.
- Investigation of algorithms with guaranteed accuracy may give rise to heuristics which are efficient for some types of objective functions although not guaranteed to solve any Lipshitz problem with prescribed accuracy.
- Parallel implementation of algorithms remarkably extended practical solvability in some classes of hard optimization problems [2–4].

Covering by spheres is mainly used as an abstract method for theoretical investigation. The constructive algorithms are defined by minimization of lower bounds of function values or by covering of the feasible region by hyperrectangles [1]. The disadvantages of both approaches are discussed and a new version of branch and bound search using a covering of the feasible region by regular simplexes is proposed.

There are two main reasons for using coverings of regular simplexes. First, if a regular simplex from the current covering is to be branched on and a covering for it constructed, several schemes can be devised, such that the resulting cover consists of a number of regular simplexes proportional to the dimension of the domain. Furthermore, for a sequence of simplexes $S_1 \supset S_2 \cdots \supset S_n \supset \dots$ constructed by our algorithm, the edge length converge exponentially to zero ensuring convergence of the algorithm.

Second, regular simplexes leave open several possibilities for function value estimation based on the Lipschitzian constant of the function. We propose a midpoint sampling scheme and a vertex sampling scheme, and investigate the properties of these in the two-dimensional case.

Finally, we address the question of initially covering the domain of the function to be optimized by a set of regular simplexes. We also address the relation of our algorithm to another algorithm based on regular simplexes by Wood.

2. A NEW VERSION OF THE BRANCH AND BOUND TECHNIQUE

Two main approaches are used to construct global minimization algorithms for Lipshitz continuous functions. The algorithms implementing the first approach define the point of $f(\cdot)$ to be evaluated next by global minimization of the lower bound of values of $f(\cdot)$. This technique aims for the maximal possible improvement at the current step. Experimental testing shows rather high efficiency of these algorithms for several one-dimensional and two-dimensional test functions with respect to the number of function evaluations [1,5]. However, even in the two-dimensional case, the auxiliary problem of minimization of the lower bound often becomes practically unsolvable [1].

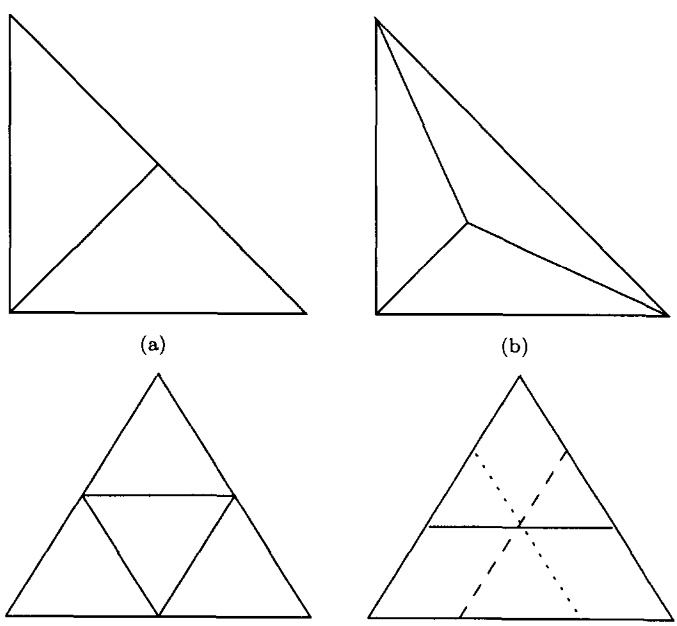
The second approach is based on the covering of A by elementary subsets, i.e., subsets of some prespecified type as spheres, hyperrectangles or simplexes. An initial covering of A is the subject to iterative refinement by subdivision of a selected subset D into a finite number smaller elementary subsets. A subdivision may result in a partition of D into mutually disjoint subset or it may result in a covering, i.e., a set of elementary subsets, for which the union of these contains D. A branch and bound technique controls the process of refinement. For each elementary subset, a lower bound estimate is calculated using only the values of $f(\cdot)$ at normally few points of this subset. If the estimate is worse than the known upper bound for $\min_{x \in A} f(x)$, then this subset may be left out from further consideration, i.e., *discarded* or *fathomed*. If this is not the case, the elementary subset under consideration is, as already mentioned, subdivided into several smaller subsets of the same type, i.e., *branching* takes place. Different types of elementary subsets may be used, as well as different strategies for selecting the next subset to process and for subdivision of a subset. Several implementations of this technique are discussed in [1].

It is interesting to note that some one-dimensional algorithms, e.g., Pijavskij-Shubert [1] and Brent [6] may be considered as representatives of both approaches. The testing in [1] shows that for two-dimensional functions, the best implementation of the branch and bound technique requires twice as many evaluations of $f(\cdot)$ as the algorithms using the first approach. But the auxiliary computations of branch and bound algorithms are essentially less time consuming. Some problems too hard for algorithms using the first approach have been solved by branch and bound algorithms.

The elementary covering subsets are normally hyperrectangles (boxes). In case interval methods are used, boxes are very natural because they correspond to the definition of operations in interval arithmetics [2,7]. The use of boxes for other covering algorithms has serious disadvantages. For example, the use of symmetric boxes (hypercubes) is normally out of question because of the exponential number (with respect to n) of objective function evaluations occurring when dividing a parent hypercube into smaller ones. Several schemes for dividing of a box into, e.g., two, smaller boxes have been proposed, however, the implied algorithms were rather slow in computational experiments [1].

Our idea is to construct a branch and bound algorithm based on potentially tight lower bounds evaluated by not too complicated auxiliary computations. The idea to choose regular simplexes as elementary covering subsets seems prospective. Different simplex based covering techniques have been proved advantageous in [8–12]. The idea pursued in the following is that Lipshitzean bound calculation for regular simplexes is a rather easy task, and hence, using a covering by regular simplexes may speed-up simplex based branch and bound algorithms considerably.

In the two-dimensional case, a regular simplex corresponds to a equilateral triangle. The latter may be subdivided into four equilateral triangles with edges half as long as the original ones as shown in Figure 1c. This corresponds to a covering, where a simplex is defined for each vertex v



(a) (b) (c) (d)

Figure 1. Three possible subdivisions of a simplex.

of the parent simplex by points subdividing each of the edges corresponding to v in the proportions 1 : 1, and defining the fourth simplex by the midpoints of each of the facets of the parent simplex. Other schemes of subdivision have also been suggested in the literature, cf., [8], in which Figures 1a and 1b are mentioned, and where also the subdivision (c) is proposed. The purpose of introducing subdivision (c) in [8] is, however, entirely different and relates to convergence properties rather than bound calculation. The subdivision (b) has an obvious disadvantage due to the possible degeneration of a triangle resulting from several subdivision steps. Subdivision (a) might be competitive with (c) with respect to easy bound calculation. The number of function evaluations in the worst case may, however, be estimated as $0.5(L/\epsilon)^2$ for (a) and $(2/3\sqrt{3})(L/\epsilon)^2$ for (c). Hence, the estimated number of function evaluations is approximately 27% larger for (a) than for (c). The subdivision into equilateral triangles has an advantage because the minimum of lower bound may be calculated by means of an analytical formula as shown below. The bounds calculated using the Lipshitz constant and the function values of the vertices of equilateral triangles covering a feasible region will in general be better than those calculated from the vertices of the irregular triangles covering the same region. Some comparison of two-dimensional subdivisions is presented below.

Another possibility exists, namely, to choose a covering of the parent simplex rather than a subdivision. This can be accomplished by defining the simplex for each of the vertices of the parent simplex by points subdividing the corresponding edge in proportions 2 : 1, cf., Figure 1d. The disadvantage of such a scheme is the overcovering introduced, which is 30% in the two-dimensional case.

In the n-dimensional case with $n > 2$, neither of the coverings described above result in subdivision of the simplex in question into a set of pairwise disjoint regular simplexes, and indeed, the existence of such a subdivision scheme is not known. The two schemes for covering described above can both be generalized, and the issue is discussed in more detail in Section 7. In the case of the first scheme, a parent simplex is covered by means of $n+2$ descendant regular simplexes. The edge length of $n+1$ of these subsimplexes equals $(n-1)/n$ of the parent edge length, and edge length of the last equals to $1/n$ of this. The three-dimensional case is illustrated by Figure 2. Let $S(z) \subset R^n$ be a regular simplex with center z and edge length d. Let a set of simplexes covering A be given, i.e., $\bigcup_{i=1}^{m} S(z_i) \supseteq A$. Simply choosing a suitable simplex S circumscribing A may be too inefficient, e.g., the volume ratio between A and S may be less than $(2/\sqrt{n(n+1)})^n$ for hypercubic regions. We propose a possible covering for the two-dimensional

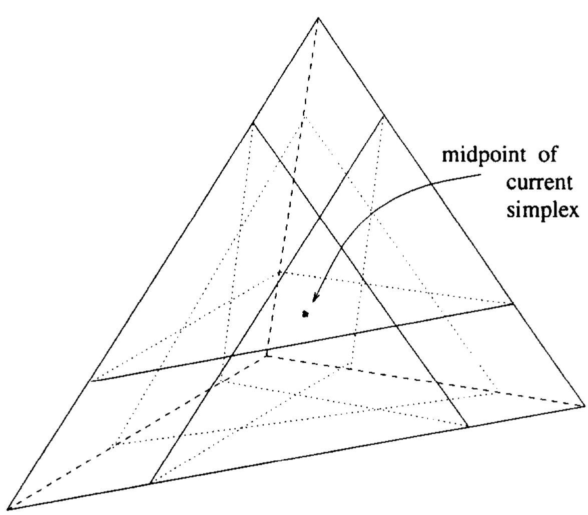


Figure 2. A subdivision of a three-dimensional simplex.

rectangular domains in Section 5. A set of standard coverings should be developed for standard domains as, e.g., hyperrectangels, but the design of an efficient initial covering for a complicated domain of a practical problem will in general constitute a challenge.

In each $S(z_i)$, at least one point x_j is chosen for calculation of the value of $f(\cdot)$. Using the available values of $f(\cdot)$ for each $S(z_i)$, a lower bound b_i for the function values of $f(\cdot)$ over $S(z_i)$ is calculated using the Lipshitz constant. Let y_{0k} be an upper bound on the global minimum after k evaluations of $f(\cdot)$, i.e., $y_{0k} \geq \min_{x \in A} f(x)$. Normally, the equality $y_{0k} = \min_{1 \leq i \leq k} f(x_i)$ holds. The simplex $S(z_i)$ is discarded if $b_i \geq y_{0k} - \epsilon$. The remaining simplexes are kept in a priority queue ordered by the values of the bounds b_i.

At each iteration of a branch and bound algorithm, the simplex $S(z_i)$ from the front of the queue is chosen to be covered. The function values of $f(\cdot)$ are calculated at the points of $S(z_i)$ defined by the algorithm, and y_{0k} is updated. The bounds for the descendant simplexes are estimated and each simplex is discarded or inserted in the priority queue depending on the relation of its bound value to y_{0k}. The algorithm terminates when the priority queue is empty. The branch and bound algorithm is, hence, an eager best-first algorithm.

Two versions of the algorithm are considered.

VERSION 1. For each simplex, the values of $f(\cdot)$ are calculated at the center of gravity of the vertexes.

VERSION 2. The values of $f(\cdot)$ are calculated at the vertexes and $(n+1)n$ points on the edges of the initial simplexes. For a selected simplex, the new function values are calculated at all or some of new $(n+1)n$ vertexes of descendant simplexes depending whether they coincide with the vertexes of previously generated simplexes (in which case, the function values are already known).

In the first version of the algorithm, the estimation of bounds is very simple:

$$b_i = f(z_i) - L \cdot \delta_{i\ \max},$$

where $\delta_{i\ \max}$ denotes the maximal distance between the center and the vertexes of the i^{th} simplex. In the second version, b_i is the solution to the minimization problem

$$b_i = \min_{x \in S(z_i)} \max_{x_j \in S(z_i)} \left(f(x_j) - L \cdot \|x - x_j\| \right), \tag{1}$$

where $x_j \in S(z_i)$ are the trial points corresponding to the vertexes of i^{th} simplex. It is shown in the following, that problem (1) is easy, at least for a two-dimensional case. Before going into details with our branch and bound method, we briefly describe the multidimensional bisection method (MB) of Wood in the context of branch and bound in order to contrast it to our branch and bound algorithm. The MB algorithm is also based on regular simplexes, and hence, may be seen as an alternative way of exploiting the properties of those. For a more detailed description, the reader is referred to [9,10,13].

3. MULTIDIMENSIONAL BISECTION

We consider the minimization of an n-dimensional function $f(\cdot)$ and its epigraph E, which is an $n+1$-dimensional body. We want to determine a point in E with the minimum value of the $n+1$ coordinate, which is the function value of f in the domain point given by the first n coordinates. The idea of MB is to maintain a collection of $n+1$-dimensional regular simplexes, whose union contains all points of E with minimal function value. Each such simplex has a unique apex as the point with lowest value of the $n+1^{\text{st}}$ coordinate, an axis parallel to the $n+1^{\text{st}}$ coordinate axis, and a top parallel to the domain space in question. Hence, for each simplex, the $n+1^{\text{st}}$ coordinate of the apex provides a lower bound for the function values for all points of the domain of f, located in the projection of top onto the domain space. In Figure 3, the simplex S determines a lower bound of $f(x)$ for $x \in [s_1, s_2]$ by the second coordinate of its apex A_S.

In each iteration of MB, a simplex with apex of lowest $n+1^{\text{st}}$ coordinate is chosen for processing. The function value of the domain point corresponding to the apex is computed, and based on this function value and the Lipshitz constant, a simplical cone is generated, in which no points of the epigraph of f can be located. This is called the removal cone. Three possibilities now exist. If the removal cone contains the simplex, the simplex is fathomed (just as in ordinary branch and bound). Otherwise, some parts of the simplex falls outside the removal cone, and these are either disjoint simplexes with tops being disjoint subsets of the top of the originating simplex or can be described as the union of simplexes each of which is a subsimplex of the originating simplex. In Figure 3, the removal cone corresponding to the domain point a of A_S is illustrated, and the effect of using this to replace S with two smaller simplices is illustrated.

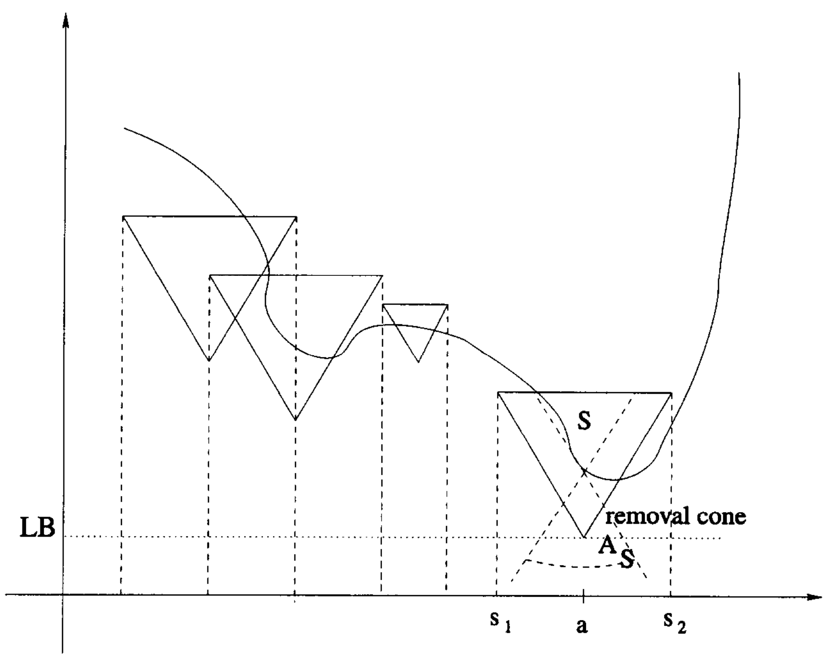


Figure 3. The $n + 1^{\text{st}}$ coordinate of the apex constitutes a lower bound.

In both of the latter cases, some of the (up to) $n + 1$ simplexes are added to the collection of simplexes to be processed, namely, those for which the $n + 1$ coordinate of the apex is not larger than or equal to the minimum value of f, in the previously evaluated points. Also, the tops of all simplexes in the collection may be lowered if the calculated function value is the best value of f found so far, possibly fathoming some of these.

In terms of branch and bound, the projections of the tops of the simplexes in the current collection onto the domain space constitute the live subspaces of the domain, and values of the $n + 1$ coordinates of their apexes constitute the bounds of these subspaces. The function evaluation for the apex of a simplex is connected to the branching process, i.e., used to generate new subspaces, and implicitly also used to fathom complete subspaces or parts hereof. The new resulting subspaces may overlap and will normally be of differing size.

In contrast, the first version of our branch and bound method, in which we use the simple bound calculation based on the evaluation of f in the center of the simplex in question, uses the function evaluations explicitly to calculate bounds, and the subdivision of a simplex is done in a regular fashion using no information related to the function evaluation. Hence, even if the MB method can be phrased in a branch and bound context with simplexes used as subspaces, the method differs substantially from both versions of our branch and bound algorithm. Note also that our second bound calculation can be expected to produce bounds substantially stronger than the apex-bounds of the MB-method.

4. CONSTRUCTION OF A SIMPLEX

We define the vertexes of the standard simplex with edge length equal to one iteratively. The $k+1$ vertices in dimension k are denoted $\chi_1^k, \chi_2^k, \dots, \chi_{k+1}^k$, and the center $\chi_c^k = (1/(k+1)) \sum_{i=1}^{k+1} \chi_i$.

- Dimension 1. The two vertices are $\chi_1^1 = 0$ and $\chi_2^1 = 1$. The center is $\chi_c^1 = 0.5$.
- Dimension k. The first k vertices are constructed by locating the k vertices of the simplex of dimension $k-1$ according to their coordinates in the hyperplane defined by $x_k = 0$. The vertex χ_{k+1}^k has the first k coordinates equal to those of χ_c^k and the $k+1^{\text{st}}$ coordinate is determined by the equality $\|\chi_{k+1}^k - \chi_1^k\| = 1$.

It follows immediately that $\chi_1^2 = (0,0)$, $\chi_2^2 = (1,0)$, and $\chi_3^2 = (1/2, \sqrt{3}/2)$; in three dimensions, we have $\chi_1^3 = (0,0,0)$, $\chi_2^3 = (1,0,0)$, $\chi_3^3 = (1/2, \sqrt{3}/2, 0)$, and $\chi_4^3 = (1/2, \sqrt{3}/6, \sqrt{6}/3)$, i.e., the one-dimensional standard simplex is the unit interval, the two-dimensional simplex is the equilateral triangle, etc. The simplexes in dimensions one, two, and three are shown in Figures 4a–c.

The general solution of the recurrence relation gives

$$\chi_{k+1:i}^k = \chi_{c:i}^{k-1}, \quad \chi_{k+1:k}^k = \sqrt{\frac{(k+1)}{2k}}, \qquad \chi_{c:i}^k = \chi_{c:i}^{k-1},$$
$$\chi_{c:k}^k = \frac{1}{\sqrt{2k(k+1)}}, \qquad i = 1, \dots, k-1. \tag{2}$$

The regular simplexes are obtained from a standard simplex by contraction/extension of R^n equally in all coordinates and orthogonal transformation of R^n.

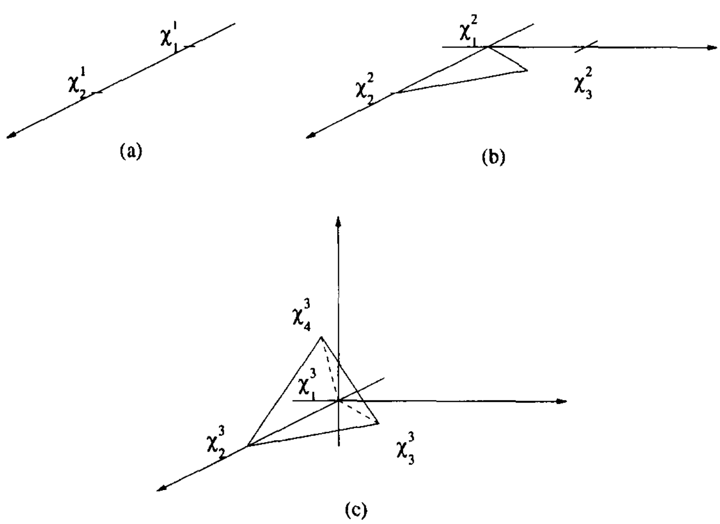


Figure 4. The simplices of dimensions one, two, and three.

5. CONVERGENCE

The conditions of general theorems on convergence given in [8] may be checked for the proposed versions of the algorithm. However, both versions may be treated uniformly by an extremely short direct proof, which also gives some quantitatives estimates. Let us denote the number of simplexes of the initial cover by N_0 and the longest edge of these simplexes by $d_{\max}$.

THEOREM 1. *The requested accuracy ϵ is achieved by the algorithm after at most $N = (n+1) \cdot n \cdot N_0 \cdot \lceil \log(d_{\max} L/\epsilon)/\log(n/(n-1)) \rceil$ calculations of objective function values*

$$y_{0N} = \min_{1 \le i \le N} f(x_i) \le \min_{x \in A} f(x) + \epsilon.$$

PROOF. After $N_C = \lceil \log(d_{\max} L/\epsilon)/\log(n/(n-1)) \rceil$ cuts of the largest simplex (let its number be h), the edge will become shorter than ϵ/L implying $b_h \ge \min_{x_i \in S(z_h)} f(x_i) - \epsilon$. Termination now appears no later than after $N_0 \cdot N_C \cdot (n+1) \cdot n$ function evaluations since then all generated simplexes have been discarded. The accuracy of evaluation of the global minimum is bounded by ϵ by the rule for discarding a simplex.

THEOREM 2. *If the algorithm runs with $\epsilon = 0$, then only the points of global minimum are the accumulating points of the sequence x_i.*

PROOF. Let $z \in A$ be a point not belonging to the set of global minimum points. Let us denote $\theta = f(z) - \min_{x \in A} f(x)$. The simplex generated by the algorithm $S(z_h) \ni z$ will be discarded after at most $N = (n+1) \cdot n \cdot N_0 \cdot \lceil \log(3 d_{\max} L/\theta)/\log(1/(n/(n-1)) \rceil$ calculations of objective function values, because for the h^{th} simplex with edge length $\theta/3L$, it holds that

$$b_h \ge \min_{x_i \in S(z_h, d_h)} f(x_i) - \frac{\theta}{3} \ge f(z) - \frac{2\theta}{3} = \min_{x \in A} f(x) + \frac{\theta}{3} \ge y_{0N}.$$

Therefore, z cannot be an accumulating point of the sequence x_i.

6. TWO-DIMENSIONAL IMPLEMENTATION

The feasible region is supposed to be rectangular. We propose a possible initial cover, which is only slightly larger than the feasible region. The construction of the cover is rather *ad hoc*, but in pilot experiments the cover proved to be efficient. Of course, other covers are possible. In the solution of combinatorial optimization problems using parallel branch and bound, similar considerations regarding initial subdivisions occur, and *ad hoc* solutions are also common.

Let A be reduced to a standard region: $0 \le x_{:1} \le 1$, $0 \le x_{:2} \le 2\sqrt{3}/3$. The original problem must be scaled to the standard feasible region. The latter is covered by the set of 28 triangles whose edge length is equal to 1/3 and whose vertexes χ_i, $i = 1, \ldots, 22$ have the coordinates $\chi_{i:1} \in \{j/6,\ j = -1, \ldots, 7\}$, $\chi_{i:2} \in \{k \cdot \sqrt{3}/6,\ k = 1, 3$, if j is odd and $k = 0, 2, 4$, if j is even$\}$. The covering is shown in Figure 5.

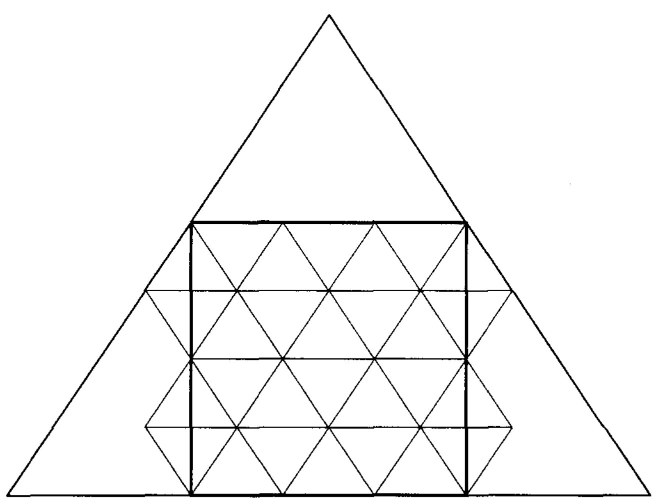


Figure 5. A simple initial covering consisting of a circumscribing regular simplex, and a tailored initial covering with regular simplexes of the unit cube in two dimensions.

The implementation of Version 1 is straightforward. Points outside of the feasible region are handled as follows. The function value of the point in question is substituted by the function value at the point obtained by an orthogonal projection onto the region boundary plus L multiplied by the distance between these two points. If all points are outside of the region, the simplex is discarded.

The implementation of Version 2 includes solving the minimization problem (1). Let us consider the standard triangle defined by (2) and suppose that $f(\chi_3) \geq f(\chi_2) \geq f(\chi_1) = 0$. By sufficient scaling, the calculation of the bound value for each triangle may be reduced to the minimization on the standard triangle S,

$$b = \min_{x \in S} \max(-\|x\|, \psi_2 - \|x - \chi_2\|, \ \psi_3 - \|x - \chi_3\|),$$

where $\psi_i = f(\chi_i)/L$, $i = 2, 3$.

THEOREM 3. *For the lower bound b, the following inequality holds:*

$$b \geq \beta_1 = \frac{(1 - \psi_2^2)}{(2\psi_2 + \sqrt{3})}.$$

PROOF. Since b is monotone in ψ_2, ψ_3, we have

$$\begin{aligned} b \geq \beta_1 &= \min_{x \in S} \max\left(-\|x\|, \psi_2 - \|x - \chi_2\|, \psi_2 - \|x - \chi_3\|\right) \\ &= \min_{h \geq 0} \min_{\|x\| = h} \max\left(-h, \psi_2 - \|x - \chi_2\|, \psi_2 - \|x - \chi_3\|\right). \end{aligned}$$

Because of symmetry of the correct triangle, the condition $\|x\| = h$ implies that the minimum of $\max(\psi_2 - \|x - \chi_2\|, \ \psi_2 - \|x - \chi_3\|)$ is achieved on the symmetry axis. Therefore, the solution to the minimization problem is

$$\beta_1 = \min_{h \geq 0} \max\left(-h, \psi_2 - \sqrt{1 + h^2 - h\sqrt{3}}\right) = \frac{1 - \psi_2^2}{2\psi_2 + \sqrt{3}},$$

and the minimum point is $x_1 = \sqrt{3}(1 - \psi_2^2)/(4\psi_2 + 2\sqrt{3})$, $x_2 = (1 - \psi_2^2)/(4\psi_2 + 2\sqrt{3})$.

THEOREM 4. *For the lower bound b, the following inequality holds:*

$$b \geq \beta_2 = \begin{cases} \left(\psi_3 - \dfrac{\sqrt{3}}{2} + \dfrac{1}{4\left(\psi_3 - \sqrt{3}/2\right)} \Big/ 2,\right) & \textit{if } \psi_3 \leq \dfrac{(-1 + \sqrt{3})}{4}, \\ -\dfrac{(1 - \psi_3^2)}{(1 + 2\psi_3)}, & \textit{otherwise.} \end{cases}$$

PROOF. Since b is monotone in ψ_2, ψ_3, we have the inequality

$$b \geq \beta_2 = \min_{x \in S} \max\left(-\|x\|, -\|x - \chi_2\|, \ \psi_3 - \|x - \chi_3\|\right). \tag{3}$$

Because of symmetry with respect to $x_1 = 1/2$, the latter minimization problem may be reduced to one-dimensional minimization in the variable $0 \leq x_2 \leq \sqrt{3}/2$ with $x_{.1}$ fixed to $1/2$ or in the variable $0 \leq x_1 \leq 1/2$ with x_2 fixed to 0. In the first case, a necessary condition for the minimum of (3),

$$\psi_3 - \left(\frac{\sqrt{3}}{2} - x_2\right) = -\sqrt{\frac{1}{4} + x_2^2}$$

is satisfied in the interior of the feasible region (at the point x where ($x_1 = 1/2$, $x_2 = -(\psi_3 - \sqrt{3}/2)/2 + 1/(8(\psi_3 - \sqrt{3}/2))$), if $\psi_3 \leq (-1+\sqrt{3})/4$. The minimal value is equal to $(\psi_3 - \sqrt{3}/2 + 1/(4(\psi_3 - \sqrt{3}/2)))/2$.

In the case $\psi_3 > (-1+\sqrt{3})/4$, the equality $x_2 = 0$ is satisfied, and the necessary minimum condition is $-x_1 = \psi_3 - \sqrt{3/4 + (1/2 - x_1)^2}$ implying the bound $\beta_2 = -(1-\psi_3^2)/(1+2\psi_3)$ with the minimum point for (3) $x = ((1-\psi_3^2)/(1+2\psi_3),\ 0)$.

To implement the algorithm, bounds for each triangle may be obtained as rescaled values of $\max(\beta_1, \beta_2)$, where β_1 and β_2 are defined by the theorems above. The obtained bound may be sufficient to discard the current triangle. If not, the tighter bounds are calculated. The system of the following equations, expressing the necessary minimum conditions,

$$\|x\| + \psi_3 - \|x - x_3\| = 0, \qquad \|x\| + \psi_2 - \|x - x_2\| = 0,$$

is numerically solved by the Newton method. For the initial point, we use the better of the two points defined above. If the trajectory of the numerical solution crosses $x_2 = 0$, then the two separate equations obtained from the system by substituting x_2 with zero should be solved and the better point chosen for the bound evaluation.

7. THE MULTIDIMENSIONAL CASE

As pointed out in Section 2, there are at least two possibilities of covering a regular simplex of dimension n with subsimplexes, also for $n > 2$.

The parent simplex may be covered with $n + 1$ subsimplexes, each of which is defined by one vertex v of the parent simplex and n points located on each of the edges of the parent simplex incident to v. Each of these points should subdivide the parent simplex edge in the proportion $n : 1$, i.e., the edge length of the new simplex should be $n/(n+1)$ times the length of the parent simplex.

The new simplexes introduce a substantial overcovering of the parent simplex. The total volume of the $n+1$ new simplexes is $n^n/(n+1)^{(n-1)} \cdot V$, where V is the volume of the parent simplex. Note, however, that for a sequence of embedded simplexes, the edge length will converge exponentially to zero, and hence, a branch and bound algorithm based on such a covering is convergent. The number of function evaluations per branching is $n+1$ for Version 1 of our algorithm and $(n+1)n$ for Version 2.

The other covering scheme consists of $n + 2$ simplexes. Of these, $n + 1$ correspond each to a vertex of the parent simplex. These are constructed as indicated above, except that the length is now set to $(n-1)/n$ times the length of the edges of the parent simplex. The last simplex is defined by the centers of the $n + 1$ simplexes of dimension $n - 1$ constituting the facets of the parent simplex. As noted before, this covering scheme results in a partitioning in case the domain is two dimensional. The total volume of these simplexes is $(n+1)((n-1)/n)^n + (1/n)^n$, i.e., the overcovering is again with a factor of n in the limit. However, for $n = 3$, the overcovering is 22% and for $n = 4$, it is 60%, to be compared to 70%, respectively, 104% for the first covering scheme described. Again, the edge lengths converge exponentially to 0 with the number of subdivision. The number of function evaluations of Version 1, respectively, Version 2 of our algorithm is $n+1$, respectively, $(n+1)^2$. Regarding the construction of a tailored initial covering of an n-dimensional unit cube/hyperrectangel, this is a nontrivial task, and we leave this open for further research.

8. RESULTS OF EXPERIMENTAL TESTING

In the recent review [1], the results of experimental testing of known algorithms are presented. The same test functions with the same estimates of Lipshitz constants and prescribed accuracies have been optimized by the algorithm proposed in the present paper. We refer to [1] for a description of the conditions of the experiment. In Table 1, the number of function evaluations for both versions of our algorithm is presented corresponding to Table XI of [1]. Also, the results for our second version with the initial cover consisting of only one simplex is given. The results of the best algorithm according to [1] are included in Table 1 for comparison. The second version of our algorithm (using the advanced bound calculation and the constructed initial covering) is clearly superior.

Table 1. Number of function evaluations in optimizing the problems given in [1] using both versions of the proposed algorithm, and giving results for Version 2 with simple initialization and with the alternative (a)-subdivision. "*" indicates that the prescribed accuracy in the experimental set-up of [1] is not clear.

Test Case [1]	Best of [1]	Version 1 Simple Est.	Version 2a Adv. Est.	Version 2b w. Simple Int.	Version 3 Alt. Subdiv.
1	643	1006	489	479	489
2	167	217	137	139	126
3	3531	6622	2618	3051	3033
3.1	3953	7489	3245	3527	–
3.2	3035	6160	2665	2826	–
3.3	3689	7951	3387	3702	–
4.1	*45	88	49	53	49
4.2	*45	103	41	47	41
5	73	112	53	69	81
6	969	3253	629	770	855
7	7969	12574	6370	7210	6309
8	301	613	255	292	290
9.	13953	22270	8759	12320	12026
9.1	14559	23119	9531	12786	–
9.2	13281	21007	9002	11649	–
9.3	12295	22207	8917	12006	–
10	1123	2014	820	947	1022
11	2677	4591	2222	2292	2345
12	12643	24997	10851	12502	11388
13	15695	28672	10643	16308	8688

The current implementation of the algorithm (in C) is experimental and has been developed to evaluate the method with repect to the number of function evaluations, and the efficiency with respect to the time of auxiliary computations has not been a major concern. Hence, we do not tabulate the running times but merely note that these were less than a few seconds for all tests. The implementation is regarded as a step in the development of a parallel version of the algorithm, and in the parallel algorithm also the time eficiency will be addressed.

One remark should be made on the optimization of test function 4. In Table XI of [1], the value of requested accuracy is given 0.0141, while according to the conditions of the experiment, it should be equal to 0.0283. Therefore, it is not clear, with which value the calculations were performed in [1]. We have optimized this function with both values of requested accuracy and included in Table 1, Version 4.1 (with 0.0283) and Version 4.2 (with 0.0141).

Many tests used by the authors of [1] are close to the worst case conditions, e.g., the Lipshitz constant is very large, but it characterizes the behavior of the objective function in a very small subregion near to the border, implying that the variation of function values in the remaining part of the feasible region with respect to the constant is insignificant. Of course, it is important to demonstrate the behavior of the algorithms under conditions similar to the worst case. However, in our opinion, it is also important to present results with the functions which we consider to be typical representatives of difficult (but not worst case) problems. Hence, we have performed experiments with two such functions, the Rastrigin test function and a two-dimensional version of the Shubert function.

The Rastrigin test function is defined by

$$f(x) = \sum_{i=1}^{2} x_{\cdot i}^2 - \cos(18x_{\cdot i}), \qquad -1 \le x_{\cdot i} \le 1, \quad i = 1, 2.$$

It is widely used for testing of global optimization algorithms [5]. The two-dimensional version of the Shubert function [1,5] is defined by

$$f(x) = \sum_{i=1}^{2} \sum_{j=1}^{5} -j \cdot \sin((j+1)x_{\cdot i} + j), \qquad -10 \le x_{\cdot i} \le 10, \quad i = 1, 2.$$

Lipshitz constants of both functions were estimated using a quadratic mesh of 1000×1000 resulting in the values 27.7 and 96.8. The guaranteed accuracies of the passive algorithm defined by a quadratic mesh of 100×100 and the values of the Lipshitz constants above are 0.392 and 13.7. The numbers of function evaluations made by the algorithm with different levels of requested accuracy are presented in Table 2.

Table 2. Number of function evaluations of both versions of the algorithm for the Rastrigin and two-dimensional Schubert functions with differing prescribed accuracies.

Rastrigin					Two Dim. Schubert				
ϵ	V 1	V 2a	V 2b	V 3	ϵ	V 1	V 2a	V 2b	V 3
0.392	1168	521	543	541	13.7	6895	2766	3481	2536
0.0392	2752	1407	2910	1377	1.37	18196	7776	8192	8628
0.00392	4180	2195	4172	4349	0.137	25180	11492	11508	> 15000
0.01	3934	2057	3803	2433	0.1	25435	11640	11633	–

In all cases, the value -2.0000 and point $(0.0000, 0.0000)$ were found for the Rastrigin function. The value -24.0430 and point $(-6.7708, 5.7813)$ were found for the second function in the case of requested accuracy 13.7, and the value -24.0612 with point $(-6.7773, 5.7910)$ were found in the other cases. The results show that the actual accuracy is much higher than the requested one. The number of function evaluations grows much slower than in worst case.

9. CONCLUSIONS

The Lipshitz model is one of best understood models of global optimization. Different ideas have been proposed and many algorithms have been implemented in the framework of this model. However, the available testing results show that the generalization from two to higher dimensions may be as complicated as the step from the implementation of one-dimensional algorithms to two-dimensional algorithms. Hence, there is some pessimism regarding the perspective of the approach. In our view there is, however, also reasons for optimism. In our proposed version of the branch and bound Lipshitz technique, some specific difficulties were overcome. The two-dimensional version of the proposed technique compares favorably with other algorithms based on the Lipshitz model. A parallel version of the algorithm can immediately be implemented using the sequential C code. The multidimensional generalization seems promising, however, some experience with the two-dimensional algorithm has to be accumulated and new possibilities, e.g., adaptive Lipshitz model [14], should be investigated prior to the extension to higher dimensions.

REFERENCES

1. P. Hansen and B. Jaumard, Lipshitz optimization, In *Handbook of Global Optimization*, (Edited by R. Horst and P. Pardalos), pp. 404–493, KAP, (1995).
2. T. Hendriksen and K. Madsen, Parallel algorithms for global optimization, *Interval Computing* **5**, 88–95, (1992).
3. J. Clausen, Parallel branch and bound—Principles and personal experience, In *Parallel Computing in Optimization*, (Edited by A. Migdalas *et al.*), pp. 239–267, KAP, (1997).
4. K. Holmqvist, A. Migdalas and P. Pardalos, Parallelised heuristics for combinatorial search, In *Parallel Computing in Optimization*, (Edited by A. Migdalas *et al.*), pp. 269–294, KAP, (1997).
5. A. Tőrn and A. Žilinskas, *Global Optimization*, Springer, (1989).
6. R. Brent, *Algorithms for Minimization Without Derivatives*, Prentice-Hall, (1973).
7. E. Hansen, *Global Optimization Using Interval Analysis*, Marcel Dekker, (1992).
8. R. Horst and H. Tuy, *Global Optimization-Deterministic Approaches*, 2$^{\text{nd}}$ Edition, Springer, (1992).
9. G. Wood, Multidimensional bisection applied to global optimization, *Computers Math. Applic.* **21** (6/7), 161–172, (1991).
10. B. Zhang, G. Wood and W. Baritompa, Multidimensional bisection: The performance and the context, *J. Global Optimization* **3**, 337–358, (1993).
11. C. Perttunen, A computational geometric approach to feasible region division in constrained global optimization, In *Proceedings of the 1991 IEEE Conference on Systems, Man, and Cybernetics*, IEEE, (1991).
12. R. Horst, P. Pardalos and N. Thoai, *Introduction to Global Optimization*, KAP, (1995).
13. W. Baritompa, Customizing method for global optimization—A geometric viewpoint, *J. Global Optimization* **3**, 193–212, (1993).
14. A. Žilinskas, Adaptive version of branch and bound type global optimization algorithm, In *Poster Proceedings of the Third International Conference Adaptive Computing in Design and Manufacture*, (Edited by I.C. Parmee), pp. 34–37, University of Plymouth, (1998).