Mathematical Modelling and Analysis
Volume 13 Number 1, 2008, pages 145–159
© 2008 Technika ISSN 1392-6292 print, ISSN 1648-3510 online

BRANCH AND BOUND WITH SIMPLICIAL PARTITIONS FOR GLOBAL OPTIMIZATION

J. ŽILINSKAS

Institute of Mathematics and Informatics

Akademijos 4, LT-08663 Vilnius, Lithuania

E-mail: `julius.zilinskas@mii.lt`

Received September 29, 2007; revised December 12, 2007; published online February 15, 2008

Abstract. Branch and bound methods for global optimization are considered in this paper. Advantages and disadvantages of simplicial partitions for branch and bound are shown. A new general combinatorial approach for vertex triangulation of hyper-rectangular feasible regions is presented. Simplicial partitions may be used to vertex triangulate feasible regions of non rectangular shape defined by linear inequality constraints. Linear inequality constraints may be used to avoid symmetries in optimization problems.

Key words: global optimization, branch and bound, simplicial partitions.

1 Introduction

Many problems in engineering, physics, economics and other subjects may be reduced to problems of global minimization. Mathematically the problem of global optimization is formulated as

$$f^* = \min_{\mathbf{x} \in \mathbf{D}} f(\mathbf{x}),$$

where $f(\mathbf{x})$ is a nonlinear objective function of continuous variables $f : \mathbf{R}^n \to \mathbf{R}$, $\mathbf{D} \subset \mathbf{R}^n$ is a feasible region, n is a number of variables. Besides of the global minimum f^* one or all global minimizers $\mathbf{x}^* : f(\mathbf{x}^*) = f^*$ should be found. No assumptions on unimodality are included into formulation of the problem – many local minima may exist. Branch and bound methods for global optimization are considered in this paper.

An iteration of a classical branch and bound algorithm processes a node in the search tree representing a not yet explored subspace of the solution space. The iteration has three main components: selection of the node to process, branching of the search tree and bound calculation. Subspaces which cannot contain a global minimum are discarded from further search pruning the branches of the search tree. The rules of initial covering and branching depend on the type of partitions used. Although hyper-rectangular partitions are usually used in global optimization, other types of partitions may be more suitable for some problems. Advantages and disadvantages of simplicial partitions are shown in this paper.

Simplex is a polyhedron in n-dimensional space with the minimal number of vertices $(n + 1)$. Therefore simplicial partitions are preferable when the values of an objective function at the vertices of partitions are used to compute bounds. Usually feasible regions of global optimization problems are hyper-rectangles defined by intervals of variables. A disadvantage of simplicial partitions is the requirement to cover a feasible region of the problem by simplices. A new general combinatorial approach for vertex triangulation of hyper-rectangle is presented.

Another advantage of simplicial partitions is that they may be used to vertex triangulate feasible regions of non rectangular shape defined by linear inequality constraints. In this case the constraints are managed by the initial covering. The advantage is illustrated by examples. Linear inequality constraints may be used to avoid symmetries in optimization problems. For example, if exchange of variables does not impact the objective function of the problem, the linear inequality constraints may be set: $x_1 \leq x_2 \leq \cdots \leq x_n$. In this case the initial hyper-rectangular feasible region is reduced to $n!$ times smaller simplex and may be tackled by branch and bound algorithms with simplicial partitions.

2 Branch and Bound for Global Optimization

Classification of global optimization methods is given in [14]:

- Methods with guaranteed accuracy:
    - Covering methods;
- Direct methods:
    - Random search methods,
    - Clustering methods,
    - Generalized descent methods;
- Indirect methods:
    - Methods approximating level sets,
    - Methods approximating objective function.

One of the classes of global optimization methods are covering methods. Covering methods can solve global optimization problems of some classes with guaranteed accuracy. Covering methods detect the sub-regions not containing the global minimum and discard them from further search. The partitioning of the sub-regions stops when the global minimizers are bracketed in small sub-regions guaranteeing the prescribed accuracy. A lower bound for the objective

Algorithm 1 General branch and bound algorithm.

```
1: Cover $\mathbf{D}$: $\mathbf{L} \leftarrow \{\mathbf{L}_j | \mathbf{D} \subseteq \bigcup \mathbf{L}_j, j = 1, \dots, m\}$ using **covering rule**
2: $\mathbf{S} \leftarrow \emptyset$, $UB(\mathbf{D}) \leftarrow \infty$
3: **while** $\mathbf{L} \neq \emptyset$ **do**
4:   Choose $\mathbf{I} \in \mathbf{L}$ using **selection rule**, $\mathbf{L} \leftarrow \mathbf{L} \setminus \mathbf{I}$
5:   **if** $LB(\mathbf{I}) < UB(\mathbf{D}) + \epsilon$ **then**
6:     Branch $\mathbf{I}$ into $p$ subsets $\mathbf{I}_j$ using **branching rule**
7:     **for all** $\mathbf{I}_j, j = 1, \dots, p$ **do**
8:       Find $UB(\mathbf{I}_j \bigcap \mathbf{D})$ and $LB(\mathbf{I}_j)$ using **bounding rules**
9:       $UB(\mathbf{D}) \leftarrow \min(UB(\mathbf{D}), UB(\mathbf{I}_j \bigcap \mathbf{D}))$
10:      **if** $LB(\mathbf{I}_j) < UB(\mathbf{D}) + \epsilon$ **then**
11:        **if** $\mathbf{I}_j$ may be a solution **then**
12:          $\mathbf{S} \leftarrow \mathbf{I}_j$
13:        **else**
14:          $\mathbf{L} \leftarrow \{\mathbf{L}, \mathbf{I}_j\}$
15:        **end if**
16:      **end if**
17:    **end for**
18:  **end if**
19: **end while**
```

function over a sub-region may be used to indicate the sub-regions which can be discarded. Some methods are based on a lower bound constructed as convex envelope of an objective function [8]. Lipschitz optimization is based on assumption that the slope of an objective function is bounded [10]. Interval methods estimate the range of an objective function over a sub-region defined by a multidimensional interval using interval arithmetic [9]. Statistical models [15] or heuristic estimates [11, 17] may be used to evaluate sub-regions. Although guaranteed accuracy is lost in this case, global optimization algorithms may be applied to solve "black box" problems. In the "black box" situation the values of an objective function are assumed to be given by an oracle, usually an objective function is given by means of a computer program and an analytical expression is not known, therefore the properties of the objective function are difficult to elicit.

A branch and bound technique can be used for managing the list of sub-regions and the process of discarding and partitioning. An iteration of a classical sequential branch and bound algorithm processes a node in the search tree representing a not yet explored sub-region of the feasible region. Each iteration has three main components: selection of the node to process, branching of the search tree by dividing the selected sub-region and pruning of the branches by discarding non-promising sub-regions. The rules of selection, branching and bounding differ from algorithm to algorithm. The general branch and bound algorithm for global optimization is shown in Algorithm 1. Before the cycle, a feasible region is covered by one or several partitions whose are added to the list of candidates $\mathbf{L}$. The branch and bound scheme aims to reduce $\mathbf{L}$ and makes it converge to $\mathbf{x}^*$.

The rules of covering and branching depend on the type of partitions used. Partitions may be hyper-rectangular, simplicial, hyper-conic or hyper-spherical. Example rules of covering rectangular feasible region and branching are shown in Fig. 1: rectangular partitions are shown in the first row, simplicial in the second row and spherical in the third and fourth rows. The first column shows covering rules and the others show branching. Partitions obtained with branch and bound algorithms for global optimization differ from those used in combinatorial optimization in that the number of possible partitions is infinite and that partitions may overlap. Usually feasible regions of general global optimization problems are hyper-rectangles. All interval and most of Lipschitz global optimization branch and bound algorithms use hyper-rectangular partitions. In this case initial covering is simple: $\mathbf{L} = \{\mathbf{D}\}$ (see first row in Fig. 1). Covering by hyper-spheres causes over-covering of a feasible region as well as overlapping of spheres themselves (see third and fourth row in Fig. 1). Use of regular simplices causes over-covering of a feasible region and non-overlapping branching is not known in more than two dimensions. The use of irregular simplices enables non-over-covering of feasible region as well as non-overlapping branching (see second row in Fig. 1).

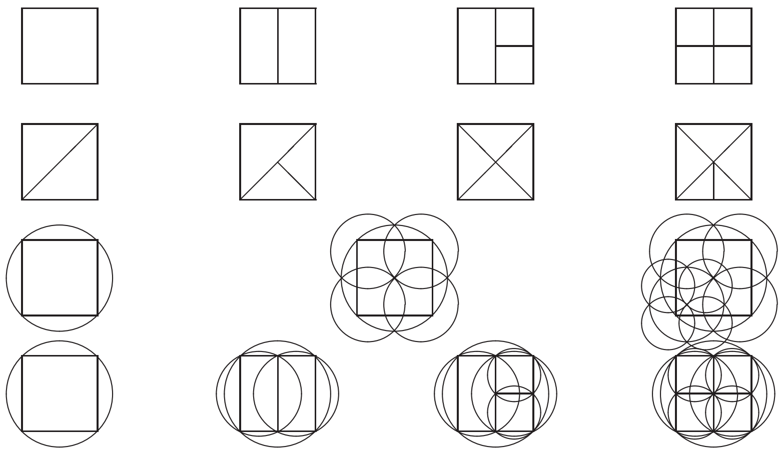


Figure 1. Example rules of covering rectangular feasible region and branching.

The bounding rule describes how the bounds for minimum are found. For the upper bound for minimum over feasible region $UB(\mathbf{D})$ the best currently found value of the objective function might be accepted. The lower bound for values of the objective function over considered sub-region $LB(\mathbf{I})$ may be estimated using convex envelopes, Lipschitz condition or interval arithmetic.

There are three main strategies of selection:

- Best first – select an element of $\mathbf{L}$ with the minimal lower bound. Candidate list can be implemented using heap and priority queue.
- Depth first – select the youngest element of $\mathbf{L}$. First-In-Last-Out structure is used for candidate list which can be implemented using stack.

- Breadth first – select the oldest element of **L**. First-In-First-Out structure is used for candidate list which can be implemented using queue.

Although covering, selection, branching and bounding rules differ in different branch and bound algorithms, the structure of the algorithm remains the same. This allows implementation of generalized branch and bound templates [2, 3]. Standard parts of branch and bound algorithms are implemented in the template, only specific rules should be implemented by the user. The template eases implementation of branch and bound algorithms for combinatorial optimization and for covering methods of continuous global optimization [4]. When computing power of usual computers is not sufficient to solve a practical global optimization problem, high performance parallel computers may be helpful. An algorithm is more applicable in case its parallel implementation is available, because larger practical problems may be solved by means of parallel computers. Because of that tools for parallelization of global optimization algorithms have been included in the template. Parallel versions can be obtained automatically using sequential program implemented using the template. Parallelization tools include master-slave and distributed versions of parallel branch and bound [19].

3 Simplex Based Branch and Bound

An n-simplex is the convex hull of a set of $(n+1)$ affinely independent points in n-dimensional Euclidean space. An one-simplex is a segment of line, a two-simplex is a triangle, and a three-simplex is a tetrahedron (see Fig. 2). A simplex is a polyhedron in n-dimensional space, which has the minimal number of vertices $(n+1)$. Therefore simplicial partitions are preferable when the values of an objective function at the vertices of partitions are used to evaluate sub-regions. Numbers of function evaluations in Lipschitz global optimization with rectangular and simplicial partitions is experimentally investigated in [16]. The experiments have shown, that simplicial partitions are preferable.

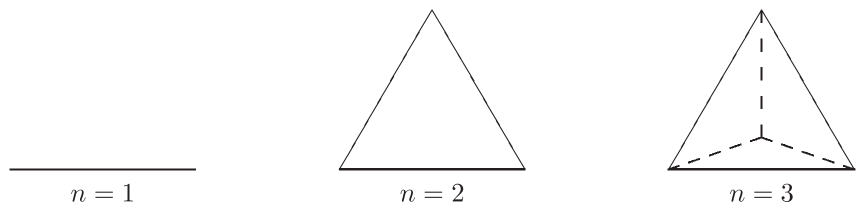


Figure 2. One-, two- and three-simplices.

Branching is carried out by means of partitioning a simplex into sub-simplices. It is known that tight bounds for function values cannot be constructed for a perverted simplex. An irregular triangle (two-dimensional simplex) may be divided into 4 similar triangles, a right equilateral triangle may be divided into 2 similar triangles, see Fig. 3.

If other branching strategies are used, the perversion of simplices must be prevented. One way of prevention is to divide simplices by a hyper-plane passing through the middle point of the longest edge and the vertices not belonging to the longest edge. This ensures that the longest edge of sub-simplices is not more than two times longer than other edges. The examples of such a division are shown in Fig. 4. Experiments in [16] have shown that this partitioning is preferable over division into several simplices.

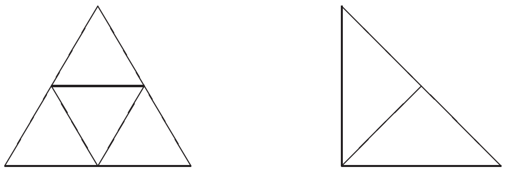


Figure 3. Subdivision of triangles into similar triangles.

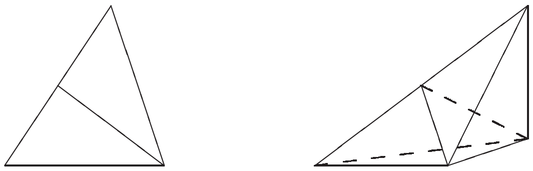


Figure 4. Subdivision of simplices into two through the middle of the longest edge.

A disadvantage of simplicial partitions is requirement to cover a feasible region of the problem by simplices. A feasible region is face-to-face vertex triangulated: it is partitioned into n-simplices, where the vertices of simplices are also the vertices of the feasible region, see Fig. 5.

Very often a feasible region in global optimization is a hyper-rectangle defined by intervals of variables. A rectangle may be vertex triangulation into two triangles. A three-dimensional rectangle may be vertex triangulated into five simplices as it is shown in Fig. 5.

However such a triangulation is not general. The general (any dimensional) algorithm for combinatorial vertex triangulation of hyper-rectangle is proposed in Algorithm 2. Here D_{j1} and D_{j2} represent the ends of interval of j-th variable defining hyper-rectangular feasible region; v_{ij} represents j-th coordinate of i-th vertex of the current simplex. This approach is deterministic, the number of simplices is known in advance, it is equal to $n!$. All simplices are of equal hyper-volume, i.e. $1/n!$ of the hyper-volume of the hyper-rectangle. By adding

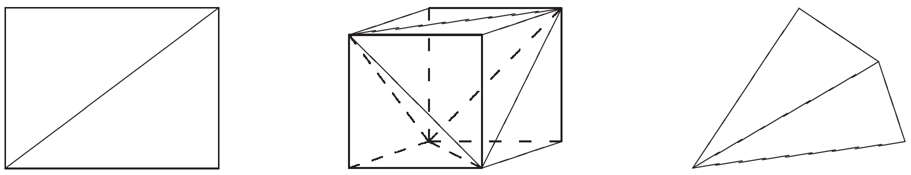


Figure 5. Examples of face-to-face vertex triangulation of feasible regions.

Algorithm 2 Combinatorial vertex triangulation of hyper-rectangle.

```
1: for $\tau$ = equals one of all permutations of $1, \dots, n$ do
2:     for $j = 1, \dots, n$ do
3:         $v_{1j} \leftarrow D_{j1}$
4:     end for
5:     for $i = 1, \dots, n$ do
6:         for $j = 1, \dots, n$ do
7:             $v_{(i+1)j} \leftarrow v_{ij}$
8:         end for
9:         $v_{(i+1)\tau_i} \leftarrow D_{\tau_i 2}$
10:    end for
11: end for
```

just one point at the middle of diagonal of the hyper-rectangle each simplex may be subdivided into two.

For example, a unit cube is vertex triangulated into six simplices:

$$\tau:\ \{1,2,3\} \quad \{1,3,2\} \quad \{2,1,3\} \quad \{2,3,1\} \quad \{3,1,2\} \quad \{3,2,1\}$$

$$\mathbf{v}: \left\{\begin{matrix}0\,0\,0\\1\,0\,0\\1\,1\,0\\1\,1\,1\end{matrix}\right\}, \left\{\begin{matrix}0\,0\,0\\1\,0\,0\\1\,0\,1\\1\,1\,1\end{matrix}\right\}, \left\{\begin{matrix}0\,0\,0\\0\,1\,0\\1\,1\,0\\1\,1\,1\end{matrix}\right\}, \left\{\begin{matrix}0\,0\,0\\0\,1\,0\\0\,1\,1\\1\,1\,1\end{matrix}\right\}, \left\{\begin{matrix}0\,0\,0\\0\,0\,1\\1\,0\,1\\1\,1\,1\end{matrix}\right\}, \left\{\begin{matrix}0\,0\,0\\0\,0\,1\\0\,1\,1\\1\,1\,1\end{matrix}\right\},$$

where τ represents a permutation and a corresponding simplex is represented by a matrix $\mathbf{v}$ which rows define the coordinates values of the vertices. Examples of combinatorial vertex triangulation of two- and three-dimensional hyper-rectangles are shown in Fig. 6.

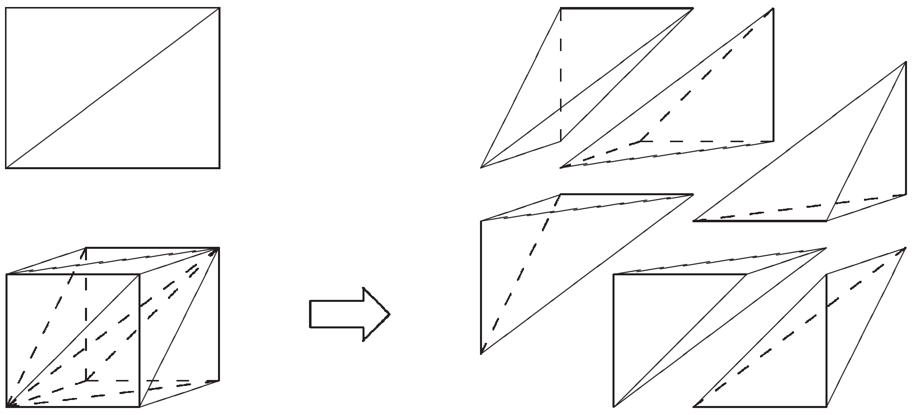


Figure 6. Examples of combinatorial vertex triangulation of two- and three-dimensional hyper-rectangles.

As the number of simplices is known in advance, efficient parallel enumeration of all simplices may be performed. Factoradic numbers can be used to assign unique numbers to permutations, such that given a factoradic of k one can quickly find the corresponding permutation. A parallel algorithm for combinatorial vertex triangulation based on factoradic numbers is given in Algorithm 3, where *size* is the number of processors, *rank* is the number of the current processor from 0 to $size - 1$, τ is the permutation of $1, \dots, n$ corresponding to the current simplex. Such an algorithm may be considered for covering of hyper-rectangular feasible regions and initial distribution of work in parallel branch and bound algorithms with simplicial partitions.

One of the advantages of simplicial partitions is that a more general feasible region defined by linear inequality constraints may be vertex triangulated. In this way constraints are managed by the initial covering.

For example, let the optimization problem be defined as

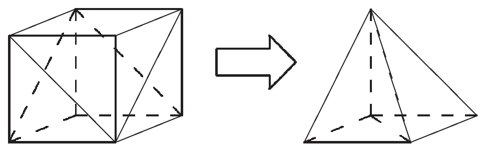
$$\min_{s.t.\,\mathbf{Q}} f(\mathbf{x}), \qquad \mathbf{Q} = \begin{cases} 0 \le x_i \le 1, \\ x_1 + x_2 \le 1, \\ x_2 - x_3 \le 0. \end{cases}$$

In this case five vertices of the hyper-rectangle defined by $0 \le x_i \le 1$ form the feasible region:

$$\left\{ \begin{matrix} x_1 = 0, x_2 = 0, x_3 = 0 \\ x_1 = 0, x_2 = 0, x_3 = 1 \\ \xcancel{x_1 = 0, x_2 = 1, x_3 = 0} \\ x_1 = 0, x_2 = 1, x_3 = 1 \\ x_1 = 1, x_2 = 0, x_3 = 0 \\ x_1 = 1, x_2 = 0, x_3 = 1 \\ \xcancel{x_1 = 1, x_2 = 1, x_3 = 0} \\ \xcancel{x_1 = 1, x_2 = 1, x_3 = 1} \end{matrix} \right\},$$

which can be vertex triangulated by two simplices

$$\mathbf{v} = \left\{ \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{matrix} \right\}, \ \mathbf{v} = \left\{ \begin{matrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{matrix} \right\}.$$



Here the origin (000) is represented by the leftmost lower vertex of the cube, the first coordinate axis is directed right, the second is directed up, and the third is directed away from the reader.

Another example:

$$\min_{s.t.\,\mathbf{Q}} f(\mathbf{x}), \qquad \mathbf{Q} = \begin{cases} 0 \le x_i \le 1, \\ x_1 + x_2 \le 1, \\ -x_2 + x_3 \le 0. \end{cases}$$

In this case four vertices of the hyper-rectangle defined by $0 \le x_i \le 1$ form

Algorithm 3 Parallel algorithm for combinatorial vertex triangulation.

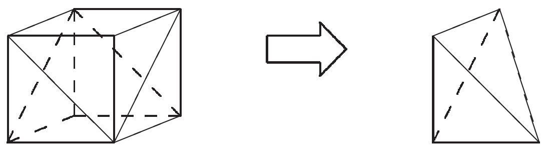
```
**for** $k = \lfloor n! rank/size \rfloor$ **to** $\lfloor n!(rank+1)/size \rfloor - 1$ **do**
    **for** $j = 1, \ldots, n$ **do**
        $\tau_j \leftarrow j$
    **end for**
    $c \leftarrow 1$
    **for** $j = 2, \ldots, n$ **do**
        $c \leftarrow c(j-1)$
        swap $\tau_{j-\lfloor k/c \rfloor \% j}$ with $\tau_j$
    **end for**
    **for** $j = 1, \ldots, n$ **do**
        $v_{0j} \leftarrow D_{j1}$
    **end for**
    **for** $i = 1, \ldots, n$ **do**
        **for** $j = 1, \ldots, n$ **do**
            $v_{(i+1)j} \leftarrow v_{ij}$
        **end for**
        $v_{(i+1)\tau_i} \leftarrow D_{\tau_i 2}$
    **end for**
**end for**
```

the feasible region:

$$\left\{\begin{array}{c} x_1 = 0, x_2 = 0, x_3 = 0 \\ \xcancel{x_1 = 0, x_2 = 0, x_3 = 1} \\ x_1 = 0, x_2 = 1, x_3 = 0 \\ x_1 = 0, x_2 = 1, x_3 = 1 \\ x_1 = 1, x_2 = 0, x_3 = 0 \\ \xcancel{x_1 = 1, x_2 = 0, x_3 = 1} \\ \xcancel{x_1 = 1, x_2 = 1, x_3 = 0} \\ \xcancel{x_1 = 1, x_2 = 1, x_3 = 1} \end{array}\right\}$$

which is a simplex

$$\mathbf{v} = \left\{\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{array}\right\}.$$



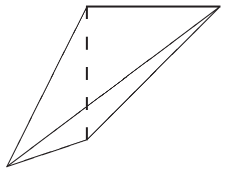
If exchange of values of the variables x_i and x_j does not change the value of the objective function, it is symmetric over the hyper-plane $x_i = x_j$. In this case equivalent solutions and equivalent sub-regions of the feasible region exist [20]. The search space can be restricted by linear inequality constraints to avoid equivalent solutions and sub-regions in the search space, and to find only one of the equivalent solutions. This is ensured by constraining the sequence of values of exchangeable variables by setting linear constraints: $x_i \leq x_j$. The resulting constrained search space may be vertex triangulated. The search space and the numbers of local and global minimizers may be reduced by avoiding such symmetries.

For example, let the optimization problem be defined as

$$f(\mathbf{x}) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos(x_i) + 1, \quad \mathbf{D} = [-500, 700]^n.$$

The objective function is symmetric over hyper-planes $x_i = x_j$. Constraints may be set to avoid symmetries: $x_1 \leq x_2 \leq \ldots \leq x_n$. The resulting simplicial search space is

$$\mathbf{D} = \left\{ \begin{array}{ccccc} -500 & -500 & \ldots & -500 & -500 \\ -500 & -500 & \ldots & -500 & 700 \\ \vdots & & \ddots & & \vdots \\ -500 & 700 & \ldots & 700 & 700 \\ 700 & 700 & \ldots & 700 & 700 \end{array} \right\}.$$



The example of three-dimensional restricted search space is illustrated. The search space and the numbers of local and global minimizers are reduced $n!$ times with respect to the original feasible region.

Lipschitz [12, 13, 16], statistical [15] or heuristic estimates presented below may be used to evaluate simplicial sub-regions. A heuristic attraction based subdivision method has been proposed in [17]. It has been developed for multidimensional global optimization with hyper-rectangular partitions. Subdivision is controlled by the information acquired during local searches. There is no guarantee that the global minimum is found with a prescribed accuracy, but the user should specify the time limit. The method is applicable in a "black box" situation. The algorithm has been applied for multidimensional scaling, many-body problems, growth model of human mandible problem [18], grillage-type foundation problem. The algorithm is presented in Algorithm 4.

Algorithm 4 Attraction based subdivision algorithm.

```
1: while not time-limit do
2:   while sub-region list is not empty do
3:     remove the best sub-region
4:     apply local searches from the sample points
5:     reject or subdivide
6:   end while
7:   for all all rejected sub-regions do
8:     subdivide and add to sub-region list
9:   end for
10: end while
```

Parallel version of the algorithm has been developed [11]. The feasible region is initially divided into sub-regions which are investigated by different parallel processors working independently. The algorithm corresponds to the distributed paradigm of parallel programming with the static load balance strategy. The termination of the algorithm is the same as the stopping condition in the sequential case – the time limit. The currently known best value of

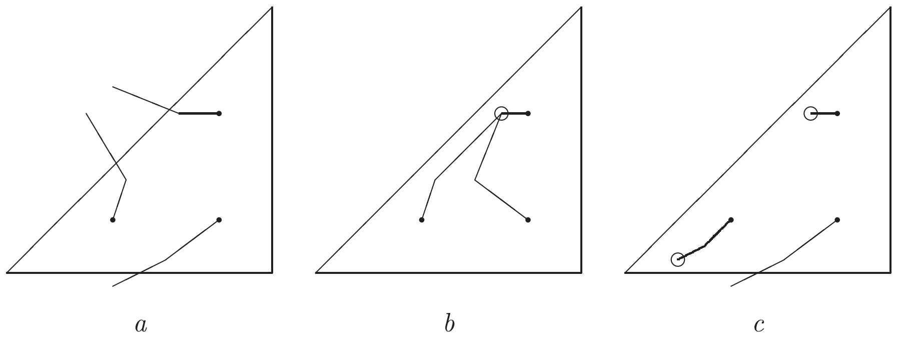


Figure 7. Heuristic evaluation of simplicial sub-region based on the results of local searches.

the objective function is exchanged between processors. The existence of the outer loop ensures that all processors always have work to do, but it is difficult to know if processors perform computations on equally promising sub-regions.

In this paper the version of the algorithm with simplicial partitions is proposed. Similarly as in the previous version of the algorithm, three cases of the results of local searches are considered:

- All local searches go outside of the sub-region, Fig. 7a. It is assumed that there is no minimum point in the sub-region and therefore it may be rejected.
- All local searches converge to the same point, Fig. 7b. It is assumed, that there is only one minimum point in the sub-region and it is already known, therefore the sub-region may be rejected.
- In the other cases, Fig. 7c, the lower bound is estimated using the largest gradient norm and the smallest objective function value in the sub-region obtained during local searches. If the estimated lower bound for the minimum over the sub-region is larger than the smallest function value found before, the sub-region may be rejected. Otherwise the sub-region is subdivided.

4 Case Study: Optimization Problems of Grillage-Type Foundations

The grillage-type foundations are the most conventional and effective scheme of foundations, especially in the case of weak grounds. Grillage consists of separate beams, which are supported by piles or reside on other beams. As piles may reach length of tens meters, reducing the number of piles will lead to substantial savings. The optimal scheme of grillage should possess the minimum possible number of piles. Theoretically, reactive forces in all piles should approach the limit magnitudes of reactions for those piles [5]. These goals can be achieved by choosing appropriate pile positions. The piles should be positioned minimizing the largest difference between reactive forces and limit magnitudes of reactions.

A designer may arrive at the acceptable pile placement scheme by engineering tests algorithms. However obtaining of optimal schemes is likely only in the case of simple geometries, simple loadings and the limited number of design parameters. Practically, this is difficult to achieve for grillages of complex geometries. To be on the safe side, the number of piles in design schemes is usually overestimated.

The problems may be approached using global optimization [6]. These are "black box" optimization problems: the values of objective function are evaluated by an independent package which models reactive forces in the grillage using finite element method. Gradient may be estimated using sensitivity analysis implemented in the modelling package. The number of piles is n, usually $n \geq 10$. The position of a pile is defined by a real number, which is mapped to a two-dimensional position by the modelling package. Possible values are in the range $[0, 77]$. The feasible region of the problems is $[0, 77]^n$. The experiments have shown that the problems are difficult and parallel optimization is helpful [1, 7].

If characteristics of all piles are equal, their interchange does not change the value of the objective function. The problem may be constrained to avoid symmetries of the objective function: $x_1 \leq x_2 \leq \ldots \leq x_n$. In this case a search space is the simplex

$$\mathbf{D} = \left\{ \begin{array}{ccccc} 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & \ldots & 0 & 77 \\ \vdots & & \ddots & & \vdots \\ 0 & 77 & \ldots & 77 & 77 \\ 77 & 77 & \ldots & 77 & 77 \end{array} \right\}.$$

The search space and the numbers of local and global minimizers are reduced $n!$ times with respect to the original feasible region.

Two piles may not be at the same position. Let us define the minimal distance between two piles by δ. The problem may be constrained to avoid symmetries of the objective function and coincidence of the piles:

$$x_1 \leq x_2 - \delta, \ldots, x_{n-1} \leq x_n - \delta.$$

In this case the search space is the simplex

$$\mathbf{D} = \left\{ \begin{array}{ccccc} 0 & \delta & \ldots & (n-2)\delta & (n-1)\delta \\ 0 & \delta & \ldots & (n-2)\delta & 77 \\ \vdots & & \ddots & & \vdots \\ 0 & 77-(n-2)\delta & \ldots & 77-\delta & 77 \\ 77-(n-1)\delta & 77-(n-2)\delta & \ldots & 77-\delta & 77 \end{array} \right\}.$$

The search space and the numbers of local and global minimizers are reduced approximately $n!$ times with respect to the original feasible region. Moreover

Table 1. Performance of attraction based subdivision algorithm.

n	$\min f^*$	$\overline{f^*}$	$\max f^*$
10	214.129	236.901	257.796
15	179.448	186.946	195.213
20	221.171	262.024	298.273
37	3998.47	5323.85	6475.33
40	1201.02	1330.1	1464.3

in this case there are no unfeasible regions in the search space where piles coincide.

Grillage-type foundation problems of various dimensionality have been optimized using attraction based subdivision algorithm with simplicial partitions. A trust region algorithm with BFGS update has been used for local optimization. 10 runs 10000s each have been performed. The results are given in Table 1, where the best ($\min f^*$), mean ($\overline{f^*}$) and the worst ($\max f^*$) estimates of global minimum in 10 runs are given. The best function value 214.129 of the problem with $n = 10$ piles has been found in one of the runs after 199s. This means that good function values are found quickly but then the solution is not necessarily improved during long period of time.

5 Conclusions

The presented approach for vertex triangulation of hyper-rectangular feasible regions of global optimization problems is general – for any dimensionality, and deterministic – the number of simplices $n!$ is known in advance.

Simplicial partitions allow reduction of the search space of optimization problems exposing symmetries. In the case of problems of grillage-type foundation with equal piles the search space is reduced $n!$ times avoiding symmetries. Moreover simplex partitions allow avoidance of coincidence of piles.

6 Acknowledgements

The research is partially supported by the NATO Reintegration grant CBP.EAP.RIG.981300 and Lithuanian State Science and Studies Foundation within the project B-03/2007 "Global optimization of complex systems using high performance computing and GRID technologies".

References

[1] M. Baravykaitė, R. Belevičius and R. Čiegis. One application of the parallelization tool of master-slave algorithms. *Informatica*, **13**(4):393–404, 2002.

[2] M. Baravykaitė and R. Čiegis. An implementation of a parallel generalized branch and bound template. *Mathematical Modelling and Analysis*, **12**(3):277–289, 2007.

[3] M. Baravykaitė, R. Čiegis and J. Žilinskas. Template realization of generalized branch and bound algorithm. *Mathematical Modelling and Analysis*, **10**(3):217–236, 2005.

[4] M. Baravykaitė and J. Žilinskas. Implementation of parallel optimization algorithms using generalized branch and bound template. In I.D.L. Bogle and J. Žilinskas(Eds.), *Computer Aided Methods in Optimal Design and Operations*, volume 7 of *Series on Computers and Operations Research*, pp. 21–28. World Scientific, 2006.

[5] R. Belevičius, S. Valentinavičius and E. Michnevič. Multilevel optimization of grillages. *Journal of Civil Engineering and Management*, **8**(1):98–103, 2002.

[6] R. Čiegis. On global minimization in mathematical modelling of engineering applications. In A. Törn and J. Žilinskas(Eds.), *Models and Algorithms for Global Optimization*, volume 4 of *Springer Optimization and Its Applications*, pp. 299–310. Springer, 2007.

[7] R. Čiegis, M. Baravykaitė and R. Belevičius. Parallel global optimization of foundation schemes in civil engineering. *Lecture Notes in Computer Science*, **3732**:305–312, 2006.

[8] C.A. Floudas. *Deterministic Global Optimization: Theory, Methods and Applications*, volume 37 of *Nonconvex Optimization and its Applications*. Kluwer Academic Publishers, 2000.

[9] E. Hansen and G.W. Walster. *Global Optimization Using Interval Analysis*. Marcel Dekker, New York, 2nd edition, 2003.

[10] R. Horst, P.M. Pardalos and N.V. Thoai. *Introduction to Global Optimization*, volume 48 of *Nonconvex Optimization and its Applications*. Kluwer Academic Publishers, 2nd edition, 2001.

[11] K. Madsen and J. Žilinskas. Parallel branch-and bound attraction based methods for global optimization. In G. Dzemyda, V. Šaltenis and A. Žilinskas(Eds.), *Stochastic and Global Optimization*, volume 59 of *Nonconvex Optimization and its Applications*, pp. 175–187. Kluwer Academic Publishers, 2002.

[12] R. Paulavičius and J. Žilinskas. Analysis of different norms and corresponding Lipschitz constants for global optimization. *Technological and Economic Development of Economy*, **12**(4):301–306, 2006.

[13] R. Paulavičius and J. Žilinskas. Analysis of different norms and corresponding Lipschitz constants for global optimization in multidimensional case. *Information Technology and Control*, **36**(4):383–387, 2007.

[14] A. Törn and A. Žilinskas. Global optimization. *Lecture Notes in Computer Science*, **350**:1–252, 1989.

[15] A. Žilinskas and J. Žilinskas. Global optimization based on a statistical model and simplicial partitioning. *Computers & Mathematics with Applications*, **44**(7):957–967, 2002.

[16] J. Žilinskas. Optimization of lipschitzian functions by simplex-based branch and bound. *Information Technology and Control*, **1**(14):45–50, 2000.

[17] J. Žilinskas. Black box global optimization inspired by interval methods. *Information Technology and Control*, **4**(21):53–60, 2001.

[18] J. Žilinskas. Application of black box global optimization algorithm inspired by interval methods for practical problems. *Information Technology and Control*, **3**(24):76–82, 2002.

[19] J. Žilinskas. Parallel algorithms for Lipschitz global optimization with simplicial partitioning. *Information Technology and Control*, **4**(25):32–36, 2002.

[20] J. Žilinskas. Reducing of search space of multidimensional scaling problems with data exposing symmetries. *Information Technology and Control*, **36**(4):377–382, 2007.