Journal of Global Optimization **27:** 193–213, 2003.

Cutting Angle Method and a Local Search ⋆

A.M. BAGIROV and A.M. RUBINOV
School of Information Technology and Mathematical Sciences, The University of Ballarat, Vic 3353, Australia (E-mail: amr@ballarat.edu.au)

Abstract. The paper deals with combinations of the cutting angle method in global optimization and a local search. We propose to use special transformed objective functions for each intermediate use of the cutting angle method. We report results of numerical experiments which demonstrate that the proposed approach is very beneficial in the search for a global minimum.

Key words: Global optimization, cutting angle method, Lipschitz programming, discrete gradient

1. Introduction

We study the following problem of global optimization:

$$f(x) \longrightarrow \min \text{ subject to } x \in S, \tag{1.1}$$

where

$$S = \{x = (x_1, \dots, x_n) \in \mathbb{R}^n : \sum_{i=1}^{n} x_i = 1, x_1 \geqslant 0, \dots, x_n \geqslant 0\} \tag{1.2}$$

is the unit simplex and f is a Lipschitz function defined on S. Problems of unconstrained global optimization with known boundaries of variables can be presented in the form (1.1) (see Section 5 for details).

The cutting angle method and its various versions for the solution of problem (1.1) have been proposed and studied in [1, 2, 6, 20, 21].

It was shown in [21] that the problem (1.1) can be reformulated as the global minimization problem of the so-called IPH function over the unit simplex (IPH is an abbreviation for an increasing positively homogeneous function of degree one). Taking into account this result, we firstly study the problem (1.1) with an IPH objective function f. The cutting angle method reduces this problem to the sequence of auxiliary problems of the form:

$$h_j(x) \longrightarrow \min \text{ subject to } x \in S \tag{1.3}$$

⋆ This research was supported by the Australian Research Council

where

$$h_j(x) = \max_{k \leqslant j} \min_{i \in I(l^k)} l_i^k x_i, \tag{1.4}$$

$$l^k = (l_1^k, \dots, l_n^k) \in \mathbb{R}^n, \; l_i^k \geqslant 0, \; i = 1, \dots, n, \; I(l^k) = \{i : l_i^k > 0\},$$

$$k = 1, \dots, j, \; j \geqslant n.$$

The auxiliary problem (1.3) is essentially of combinatorial nature.

Note that the cutting angle method starts from a subset of $m \geqslant n$ initial points, which includes all vertices of the simplex S.

Let x^{j+1} be a solution of problem (1.3) and $\lambda_j = \min_{x \in S} h_j(x) = h_j(x^{j+1})$ be the value of the problem. It can be shown (see, for example, [2, 5]) that $\lambda_j = h_j(x^{j+1}) \longrightarrow f_* := \min_{x \in S} f(x)$ as $j \to +\infty$ and any limit point of the sequence (x^j) is a global minimizer of the function f over the simplex S. The cutting angle method produces as a rule a sequence $(f(x^j))$, which is not decreasing.

Assume now that the objective function f of problem (1.1) is Lipschitz. It was proved in [21] that there exists a constant $c' > 0$ such that for all $c \geqslant c'$ the function $f_1(x) = f(x) + c$ can be considered as the restriction of an IPH function to S, and so the cutting angle method can be applied to the function f_1. Numerical experiments demonstrated that the method is rather sensitive to the choice of the constant c.

An auxiliary function h_j defined by (1.4) is the maximum of a number of min-type functions of the form $x \mapsto \min_i l_i x_i$. If this number is fairly large, then the global minimization of the function h_j is time-consuming. So the cutting angle method produces fairly quickly several first iterations and much more time is required for the next iterations. This observation leads to a fruitful idea to change sometimes the objective function of the problem. In particular, it is convenient to do such changes, using a combination of cutting angle method with a local search.

Consider the global minimization problem of a Lipshitz function f over the unit simplex. Assume that a stationary point y of this function has been already found (we can find it by a method of local optimization). In order to leave this point we apply the cutting angle method. However, since this method does not produce a decreasing sequence we cannot take advantage of the known value $f(y)$ of the objective function, even if we include y into the set of initial points. In order to use this value, we transform the objective function f, that is, instead of f we consider a certain *transformed* objective function ψ. As a rule we consider a transformed function ψ such that $\psi(x) \leqslant f(y)$ for all x, so we exclude all stationary points y' of the function f, for which $f(y') > f(y)$. Applying the cutting angle method to a transformed function ψ and exploiting properties of this function, we can find a new point $\bar{y}$ such that $f(\bar{y}) < f(y)$. Numerical experiments show that applying appropriate transformed functions, we can sufficiently quickly leave a known stationary point y of the function f. Note that we use the cutting angle method only for reduction of the known value of a transformed function, so we do not need to find the global minimum of this function.

As a rule we consider non-smooth thransformed functions, even if the objective function f is smooth, so we cannot use smooth methods of local optimization for a local search.

In this paper we carry out a local search by a derivative free discrete gradient (briefly DG) method (see [4]). Numerical exprerimemts show that this method can jump over some stationary points, which are not local minimizers, so we can reduce the number of stationary points, which we meet.

For applications of the cutting angle method to the minimization of a transformed function ψ we need to add a constant c to ψ in order to obtain a function, which can be considered as the restriction of an IPH function to S. Our numerical experiments show that the cutting angle method is not very sensitive to the choice of c, if we use this method only for reducing a value of an objective function. Thus a combination with a local search allows us fairly quickly solve many problems.

The paper has the following structure. In Section 2 we provide some brief preliminary definitions and results related to IPH functions. One of the versions of the cutting angle method is described in Section 3. Transformed functions are introduced and studied in Section 4. Problem of unconstrained optimization are discussed in Section 5. Results of numerical experiments are presented in Section 6.

2. Preliminaries

Consider the n-dimensional space $\mathbb{R}^n$. We shall use the following notation:

- $I = \{1, \dots, n\}$;
- x_i is the i-th coordinate of a vector $x \in \mathbb{R}^n$;
- $[l, x] = \sum_{i \in I} l_i x_i$ is the inner product of vectors l and x;
- if $x, y \in \mathbb{R}^n$ then $x \geqslant y \iff x_i \geqslant y_i$ for all $i \in I$;
- if $x, y \in \mathbb{R}^n$ then $x \gg y \iff x_i > y_i$ for all $i \in I$;
- $\mathbb{R}^n_+ := \{x = (x_1, \dots, x_n) \in \mathbb{R}^n : \ x_i \geqslant 0 \text{ for all } i \in I\}$ (the nonnegative orthant);
- $S = \{x \in \mathbb{R}^n_+ : \ \sum_{i \in I} x_i = 1\}$ (the unit simplex).

Recall that a function f defined on $\mathbb{R}^n_+$ is called increasing if $x \geqslant y$ implies $f(x) \geqslant f(y)$; the function f is positively homogeneous of degree one if $f(\lambda x) = \lambda f(x)$ for all $x \in \mathbb{R}^n_+$ and $\lambda > 0$. We shall consider IPH (increasing positively homogeneous of degree one) functions.

Let $l = (l_i)_{i \in I} \in \mathbb{R}^n_+$. Consider the set of indices $I(l) = \{i \in I : l_i > 0\}$. The vector l defines the so-called min-type function $x \mapsto \min_{i \in I(l)} l_i x_i$, which we denote by the same symbol l. Clearly a min-type function is IPH.

We shall use the following notation for $c \in \mathbb{R}$ and $l \in \mathbb{R}^+_n$:

$$(c/l)_i = \begin{cases} c/l_i & \text{if } i \in I(l), \\ 0 & \text{if } i \notin I(l). \end{cases} \tag{2.1}$$

The vector $f(x^0)/x^0$ is called the support vector of a function f at a point $x^0 \in \mathbb{R}^n_+ \setminus \{0\}$.

Subsequently we will use the unit vectors $e^m = (0, \ldots, 0, 1, 0, \ldots, 0)$ with $I(e^m) = \{m\}$. Clearly the vector $l = f(e^m)/e^m$ can be represented in the form $l = f(e^m)e^m$. We also have $l(x) = f(e^m)x_m$ for $x \in \mathbb{R}^n_+$.

The following statement can be found in [20, 21].

THEOREM 2.1. *Let f be a Lipschitz function defined on the unit simplex S with a Lipschitz constant L. Let $c' = 2L - \min_{x \in S} f(x)$. Then for each $c \geqslant c'$ the function $f_c(x) = f(x) + c \ (x \in S)$ can be extended to an IPH function g_c defined on $\mathbb{R}^n_+$.*

Assume that the constant c' is known and consider the problem (1.1):

$$f(x) \longrightarrow \min \quad \text{subject to } \ x \in S.$$

Let $c \geqslant c'$. Then the problem (1.1) is equivalent to the problem

$$f_c(x) \longrightarrow \min \quad \text{subject to } \ x \in S,$$

which, in turn, is equivalent to the problem

$$g_c(x) \longrightarrow \min \quad \text{subject to } \ x \in S.$$

Indeed, since g_c is an extension of f_c, numbers $f_c(x)$ and $g_c(x)$ coincide for $x \in S$.

Thus, the global minimization of a Lipschitz function over the unit simplex S can be transformed to the global minimization of an IPH function over S.

3. Cutting angle method

In this section we give a brief description of a version of the cutting angle method for solving problem (1.1) with an IPH objective function f. Note that an IPH function is nonnegative on $\mathbb{R}^n_+$. We assume that $f(x) > 0$ for all $x \in S$. It follows from positiveness of f that $I(l) = I(x)$ for all $x \in S$ and $l = f(x)/x$.

The cutting angle method

Step 0. (Initialization) Take points $x^k \in S$, $k = 1, \ldots m$, where $m \geqslant n$ and $x^k = e^k$ for $k = 1, \ldots, n$ and $x^k \gg 0$ for $k = n+1, \ldots, m$. Let $l^k = f(x^k)/x^k$, $k = 1, \ldots, m$. Define the function h_m :

$$h_m(x) = \max_{k \leqslant m} \min_{i \in I(l^k)} l^k_i x_i = \max \left\{ \max_{k \leqslant n} l^k_k x_k, \ \max_{n+1 \leqslant k \leqslant m} \min_{i \in I(l^k)} l^k_i x_i \right\}$$

and set $j = m$.

Step 1. Find a solution x^* of the problem

$$h_j(x) \longrightarrow \min \quad \text{subject to} \quad x \in S. \tag{3.1}$$

Step 2. Set $j = j + 1$ and $x^j = x^*$.
Step 3. Compute $l^j = f(x^j)/x^j$, define the function

$$h_j(x) = \max\{h_{j-1}(x), \ \min_{i \in I} l_i^j x_i\} \equiv \max_{k \leqslant j} \ \min_{i \in I(l^k)} l_i^k x_i$$

and go to Step 1.

REMARK 3.1. An interpretation of the term “cutting angle method" can be found, for example, in [20].

This algorithm can be considered as a version of the cutting angle method ([1, 2]). A more general version of this algorithm, known as the Φ - bundle method, has been discussed in [18]. Convergence of the Φ-bundle method was proved in [18] under very mild assumptions.

The cutting angle method provides a sequence of lower estimates for the global minimum f_* of (1.1) with an IPH objective function, which converges to f_*. Theoretically this sequence can be used for establishment of a stopping criterion (see [20] for details). Let

$$\lambda_j = \min_{x \in S} h_j(x) = h_j(x^{j+1}) \tag{3.2}$$

be the value of the problem (3.1). Then

$$\lambda_j \equiv \min_{x \in S} h_j(x) \leqslant \min_{x \in S} f(x).$$

Thus λ_j is a lower estimate of the global minimum f_*. It is known (see, for example, [20]), that λ_j is an increasing sequence and $\lambda_j \to f_*$.

Unfortunately the cutting angle method constructs the sequence, which is not necessary decreasing: it is possible that $f(x^{j+1}) > f(x^j)$ for some j.

The most difficult and time-consuming part of the cutting angle method is solving the auxiliary problem (3.1). A method for the solution of this problem was proposed in [5]. Some modifications of this method (and corresponding modifications of the cutting angle method) are discussed in detail in [6]. We use these modifications for numerical experiments.

Only one value of the objective function is used at each iteration of the cutting angle method. (Some modifications of this method require to evaluate a few values of the objective function at each iteration.) This observation shows that it is beneficial to apply cutting angle method for the minimization of functions, whose evaluation is very time-consuming. In particular, this method can be used for the minimization of marginal functions

$$f(x) = \max_{y \in a(x)} \varphi(x, y),$$

where $a(x)$ is a set-valued mapping with compact images and φ is a continuous function. (See [7] for details.)

Numerical experiments show that the cutting angle method is able to find only approximate global minimizers. One of the main drawbacks of this method is a slow convergence of the lower estimates λ_j for the global minimum. Often an approximate global minimizer can be found sufficiently quickly, however we need to spend a lot of time in order to confirm that a point, which we have, provides a good approximation of the global minimum.

4. A combination of the cutting angle method and a local search

In this section we propose a combination of the cutting angle method and a local search for the global minimization of a Lipschitz function f over the unit simplex. Theorem 2.1 shows that such a problem can be transformed to a problem of the global minimization of an IPH function over the unit simplex. We assume that for each initial point $\bar{y}$ a local method under consideration constructs a sequence (y^k) such that

1) (y^k) tends to a stationary point of f;
2) the sequence $(f(y^k))$ is strictly decreasing: $f(y^{k+1}) < f(y^k)$ for all k.

Starting from an arbitrary point $\bar{y}^0$ we use the local search in order to obtain a stationary point y^1. Then the cutting angle method allows us to obtain a point $\bar{y}^1$ such that $f(\bar{y}^1) < f(y^1)$. Starting a local search from the point $\bar{y}^1$ we obtain a new stationary point y^2 such that $f(y^2) < f(\bar{y}^1) < f(y^1) < f(\bar{y}^0)$ and so on.

Applying the cutting angle method to the minimization of the function f we are not able to take into account the known value of this function at a stationary point y, which was found by a local search. In order to use this value we shall transform a function f into a new function ψ. A function ψ is called a *transformed function* of f (with respect to a point y) if

1) $\psi(y) \leqslant f(y)$;
2) $\min_{x \in S} \psi(x) = \min_{x \in S} f(x)$.

It follows from 2) that the set $T(x^*) = \{x \in S : f(x) \leqslant \psi(x^*)\}$ is nonempty for all x^*. Indeed if $\bar{x}$ is a global minimizer of f over S then $f(\bar{x}) = \min_{x \in S} \psi(x) \leqslant \psi(x^*)$ so $\bar{x} \in T(x^*)$.

REMARK 4.1. It is assumed (for the implementation of the proposed below algorithm), that at least one point from the set $T(x^*)$ can be easily computed.

The simplest example of a transformed function is the function $\psi(x) = f(x)$. We now give some more complicated examples.

PROPOSITION 4.1. *The following functions are transformed ones for a function f with respect to a point y:*

1) $\psi_1(x) = \min(f(x), f(y))$;
2) $\psi_2(x) = \min_{k=1,\dots,m} \min_{\alpha \in A_k} \psi_1(\alpha x + (1-\alpha)x^k)$, *where* $1 \in A_k \subset [0, 1]$, $x^k \in S$, $k = 1, \dots, m$;
3) $\psi_3(x) = \min_{k=1,\dots,m} \min_{\alpha \in A_k} f(\alpha x + (1-\alpha)x^k)$, *where* A *and* x^k *are as in item 2);*
4) $\psi_4(x) = \min_{\beta \in B} \min_{\alpha \in A} \sum_{k \in K} \beta_k f(\alpha x + (1-\alpha)x^k)$, *where* K *is a finite set of indices,* $B = \{(\beta_k)_{k \in K} : \sum_{k \in K} \beta_k = 1,\ \beta_k \geqslant 0\ (k \in K)\}$, $1 \in A \subset [0, 1]$, $x^k \in S\ (k \in K)$;
5) $\psi_5(x) = \min_{\beta \in B} \min_{\alpha \in A} \sum_{k \in K} \beta_k \psi_1(\alpha x + (1-\alpha)x^k)$, *where* K, B, A *and* x^k *are as in item 4);*
6) $\psi_6(x) = \alpha \psi_1(x) + (1-\alpha) f(x)$ *where* $\alpha \in (0, 1)$.

Proof. We consider only functions ψ_1, ψ_2 and ψ_4. A proof for other functions is similar.

1) Function ψ_1. Clearly $\psi_1(y) \leqslant f(y)$; we have also

$$\min_{x \in S} f(x) = \min_{x \in S} \min(f(y), f(x)) = \min_{x \in S} \psi_1(x).$$

2) Function ψ_2. Since $1 \in A_k$ for all k, it follows that

$$\psi_2(y) \leqslant \psi_1(y) = f(y).$$

Consider a point $x^* \in S$. Let k be an index and $\alpha_k \in A_k$ be a number such that $\psi_1(\alpha_k x^* + (1-\alpha_k)x^k) = \psi_2(x^*)$. Denote the point $\alpha_k y + (1-\alpha_k)x^k$ by $\tilde{x}$. Then

$$\psi_2(x^*) = \psi_1(\tilde{x}) = \begin{cases} f(y) & f(\tilde{x}) \geqslant f(y), \\ f(\tilde{x}) & f(\tilde{x}) \leqslant f(y). \end{cases}$$

Hence, if $f(\tilde{x}) \geqslant f(y)$ then $f(y) = \psi_2(x^*)$, if $f(\tilde{x}) \leqslant f(y)$ then $f(\tilde{x}) = \psi_2(x^*)$. It means that $y \in T(x^*)$ if $f(x) \geqslant f(y)$ and $\tilde{x} \in T(x^*)$ if $f(\tilde{x}) \leqslant f(y)$.

Let f_* be the global minimum of f. Then we have for all $x \in S$:

$$\psi_2(x) = \min_k \min_{\alpha \in A_k} f(\alpha x + (1 - \alpha x^k)) \geqslant f_*,$$

so $\min_{x \in S} \psi_2(x) \geqslant f_*$. Let z be a global minimizer of f. Since $1 \in A_k$ for all k it follows that $\psi_2(z) = f(z) = f_*$ so $\min_{x \in S} \psi_2(x) \leqslant f_*$. Thus $\min_{x \in S} \psi_2(x) = \min_{x \in S} f(x)$.

3) Function ψ_4. Since $1 \in A$, we conclude that $\psi_4(y) \leqslant f(y)$. Let $x^* \in S$. Consider a set $(\beta_k)_{k \in K} \in B$ and a number $\alpha \in A$ such that

$$\psi_4(x^*) = \sum_k \beta_k f(\alpha x^* + (1-\alpha)x^k).$$

If $f(\alpha x^* + (1-\alpha)x^k) > \psi_4(y)$ for all $k \in K$, then also $\psi_4(x^*) > \psi_4(x^*)$ which is impossible. So there exists k' such that $\alpha x^* + (1-\alpha)x^{k'} \in T(x^*)$ (In particular an index k' such that $f(\alpha x^* + (1-\alpha)x^{k'}) = \min_{k\in K} f(\alpha x^* + (1-\alpha)x^k)$ enjoys this property.) Let z be a global minimizer of the function f. Then $\psi_4(z) = f(z)$. On the other hand $\psi_4(x) \geqslant f(z)$ for all $x \in S$, hence $\min_{x\in S} f(x) = \min_{x\in S} \psi_4(x)$. $\triangle$

REMARK 4.2. a) functions $\psi_2 - \psi_5$ depend on many parameters.
b) All functions $\psi_1 - \psi_6$ are nonsmooth.

Consider the following problem of global optimization:

$$f(x) \longrightarrow \min \quad \text{subject to } \; x \in S. \tag{4.1}$$

We propose an algorithm for its solution. This algorithm is based on a combination of the cutting angle method and a local search and on a choice of transformed functions ψ. We consider an algorithm for a local search, which possesses the following properties: for an arbitrary initial point z^0 this algorithm constructs a sequence (z^k) such that

$$f(z^0) > f(z^1) > \dots f(z^k) > \dots \tag{4.2}$$

and each limit point of this sequence is a stationary point of the function f over the simplex S.

Algorithm

Step 0. (Initialization) Choose an arbitrary starting point $\bar{y}^0$. Set $i = 0$.

Step 1. Find a stationary point of f over S by the local method, starting from the point $\bar{y}^i$. Denote this stationary point by y^i and let $f_i = f(y^i)$.

Step 2. Construct a transformed function ψ^i of the function f with respect to the point y^i.

Step 3. Take points $x^k = e^k$, $k = 1, \dots, n$, $x^{n+1} = y^i$. Let $l^k = \psi^i(x^k)/x^k$, $k = 1, \dots, n+1$ and set $j = n+1$. Construct the function h_j, defined by (1.4).

Step 4. Solve the problem

$$h_j(x) \longrightarrow \min \quad \text{subject to} \quad x \in S. \tag{4.3}$$

Step 5. Let x^* be a solution of the problem (4.3). Set $j = j+1$ and $x^j = x^*$.

Step 6. Compute $\psi^* = \psi^i(x^*)$. If $\psi^* < f_i$ then find a point $x' \in T(x^*)$, set $i = i+1$, $\bar{y}^i = x'$ and go to Step 1.

Step 7. Otherwise compute $l^j = \psi^i(x^j)/x^j$, define the function

$$h_j(x) = \max\{h_{j-1}(x), \min_{i\in I(l^j)} l_i^j x_i\} \equiv \max_{k\leqslant j} \min_{i\in I(l^k)} l_i^k x_i$$

and go to Step 4.

REMARK 4.3. We use different stopping criterions for a local method and the cutting angle method. When we apply a local method first we reduce the constrained minimization problem to the unconstrained one using exact penalty functions. A local method terminates when difference between the values of objective function (the penalty function in the case of the constrained minimization) in two successive iterations is less than a given tolerance $\epsilon > 0$. We take $\epsilon = 10^{-4}$ in numerical experiments. Theoretically we can use the stopping criterion proposed in [20] for the cutting angle method. But the achievement of this criterion is time-consuming for many large-scale problems. Therefore we used the following stopping criterion in our numerical experiments: the number of iterations generated by the cutting angle method is restricted by a number N: if after N iterations the cutting angle method cannot leave a local minimizer, we accept this minimizer as a surrogate of a global minimum. For example, in [9] where the proposed method has been applied for solving data classification problems, the number of iterations generated by the cutting angle method is restricted by 70.

PROPOSITION 4.2. *Assume that the function f has a finite number of stationary points. Then the algorithm terminates after finite number of iterations at a global minimizer of f.*

Proof. Consider the iteration (i). Step 1 of Algorithm at this iteration leads to a stationary point y^i of the function f. If y^i is a global minimizer of f then the transformed function ψ^i of the function f with respect to y^i is the constant $f(y^i)$. Using the cutting angle method, we can discover that the global minimum of ψ^i is equal to $f(y^i)$. Since $\min\{\psi^i(x) : x \in S\} = \min\{f(x) : x \in S\}$, we can assert that y^i is a global minimizer.

Now let us consider the case where y^i is not a global minimizer of f. First assume that y^i is a global minimizer of the transformed function ψ^i. We can discover it by applying the cutting angle method. Let $\tilde{x}^i \in T(y^i)$. Then we have

$$f(\tilde{x}^i) \leqslant \psi^i(y^i) = \min\{\psi^i(x) : x \in S\} = \min\{f(x) : x \in S\},$$

so $\tilde{x}^i$ is a global minimizer of f. Assume now that y^i is not a global minimizer of ψ^i. Since the cutting angle method converges to a global minimizer, its application to the transformed function ψ^i leads to a point $\bar{y}_*^{i+1}$ such that $\psi(y_*^{i+1}) < \psi(y^i) \leqslant f(y^i)$. Let $\bar{y}^{i+1} \in T(y_*^{i+1})$. Then $f(\bar{y}^{i+1}) \leqslant \psi(y_*^{i+1})$. Starting a local search from the point $\bar{y}^{i+1}$ we obtain a new stationary point y^{i+1}. Due to (4.2) we have $f(y^{i+1}) < f(y^i)$, so $y^{i+1} \neq y^i$. Since the function f has a finite number of stationary points, the algorithm terminates after a finite number of steps. $\triangle$

The implementation of this algorithm requires to clarify some points.
1) Step 1 of the Algorithm includes the search for a stationary point. We use the so-called discrete gradient (DG) method (see [3, 4]) for a local search. This is a derivative free method of nonsmooth optimization. Our numerical experiments allow us to suppose that the DG method with a certain adjustment of parameters

can jump over some stationary points, which are not local minima. The possibility to jump over stationary points is one of the main reasons for exploiting of DG method. However, different local methods can be useful for a local search for some classes of objective functions.
2) We consider only transformed functions ψ, which enjoys the following property: $\psi(x) \leqslant f(x)$ for all $x \in S$. This property allows us to exploit the known (record) value $f(y^i)$ of the function f at the iteration i and to avoid stationary points y such that $f(y) > f(y^i)$. For many transformed functions ψ^i the point y^i is a global maximizer. So we use the cutting angle method in order to leave the global maximum (it is impossible if y^i is a global minimizer, since in such a case the function ψ^i is constant.)
3) Only two types of transformed functions have been used in numerical experiments. First, at each iteration i we consider the function ψ_1^i:

$$\psi_1^i(x) = \min(f(x), f(y^i)).$$

This function does not depend on parameters. Its plot contains many flat pieces, which correspond to its global maximum. We also consider the function ψ_2^i:

$$\psi_2^i(x) = \min_{k \in I} \min_{\alpha \in A_k} \psi_1(\alpha x + (1-\alpha)e^k),$$

where $A_k = \{0.2, 0.4, 0.6, 0.8, 1\}$ for all $k = 1, \dots, n$. Both of these types of functions were examined in Proposition 4.1. For functions ψ_2^i we choose vertices e^k of the unit simplex as points x^k in Proposition 4.1 (this is the most natural choice for the simplex).

REMARK 4.4. The choice of the sets $A_k, k = 1, \dots, n$, depends on the problem under consideration and in particular, on the number of variables. The number of elements of A_k should be large enough in order to obtain a good minorant for the objective function f. On the other hand it should not be too large, otherwise we will have a large number of objective function evaluations at each iteration of the cutting angle method. Numerical experiments show that the best choice in this situation is to consider sets A_k, which contain 4-7 points. In our numerical experiments A_k consists of 5 elements for all $k = 1, \dots, n$ as was shown above.

Some numerical experiments were carried out with the proposed Algorithm. Description of these experiments and results obtained can be found in Section 6.

5. Unconstrained minimization

Consider the following problem of unconstrained global optimization

$$f(x) \longrightarrow \min \quad \text{subject to} \quad x \in \mathbb{R}^n. \tag{5.1}$$

Let x^* be a global minimizer of (5.1). Assume that we know a vector of lower bounds $(a_i)_{i \in I}$ and a vector of upper bounds $(b_i)_{i \in I}$ of the unknown point x^*. Let

$x_i' = x_i - a_i$ and $t = \sum_{i \in I}(b_i - a_i)$. Then $x_i' \geqslant 0$ and $\sum_{i \in I} x_i' \leqslant t$. Let $z_i = x_i'/t$, $(i = 1, \ldots, n)$ and $z_{n+1} = 1 - \sum_{i \in I} z_i$. Then $z_1 \geqslant 0, \ldots, z_n \geqslant 0, z_{n+1} \geqslant 0$ and $\sum_{i=1}^{n+1} z_i = 1$. Thus, the change of variables allows us to substitute the problem (5.1) for the following problem of global minimization over the unit simplex $S_* \subset \mathbb{R}^{n+1}$:

$$g(z_1, \ldots, z_n) \longrightarrow \min \quad \text{subject to} \quad z = (z_1, \ldots, z_n, z_{n+1}) \in S_*.$$

This problem can be solved by the proposed Algorithm.

REMARK 5.1. It follows from the aforesaid that the problem

$$f(x_1, \ldots, x_n) \longrightarrow \min \quad \text{subject to} \quad \sum_i x_i \leqslant 1, \quad x_i \geqslant 0, \; i = 1, \ldots, n \tag{5.2}$$

can also be solved by the proposed Algorithm.

We now consider the following problem, which arises in cluster analysis (see [8] for details).

Let $A = \{a^p\}_{p \in P}$ be a given finite subset of $\mathbb{R}^n$. Consider a set $X = \{x^q\}_{q \in Q} \subset \mathbb{R}^n$, where Q is a finite set of indices. The deviation from a point $a^p \in A$ to the set X is equal to $d(a^p, X) = \min_{q \in Q} \|x^q - a^p\|$. The total deviation from the set A to the set X is equal to

$$d(A, X) = \sum_{p \in P} d(a^p, X) = \sum_{p \in P} \min_{q \in Q} \|x^q - a^p\|.$$

We are interested in the situation, where the cardinality $|Q|$ of the set X is much less then the cardinality $|P|$ of the set A. In such a case we can consider X as a certain approximator of A. The collection of points $(\bar{x}^q)_{q \in Q}$ can be considered as the best approximator of the cardinality $|Q|$ if

$$\sum_{p \in P} \min_{q \in Q} \|\bar{x}^q - a^p\| = \min_{x^1, \ldots, x^{|Q|} \in \mathbb{R}^n} \sum_{p \in P} \min_{q \in Q} \|x^q - a^p\|.$$

If $|Q| = 1$ then the best approximator $\bar{x}$ (of cardinality one) can be found as a solution of the following problem of convex programming:

$$\sum_{p \in P} \|x - a^p\| \longrightarrow \min \quad \text{subject to} \quad x \in \mathbb{R}^n.$$

We can consider the vector $\bar{x}$ as a centre of the set A. If $|Q| = 2$, we have the following complicated problem of global optimization:

$$\sum_{p \in P} \min(\|x^1 - a^p\|, \|x^2 - a^p\|) \longrightarrow \min \quad \text{subject to} \quad (x^1, x^2) \in \mathbb{R}^n \times \mathbb{R}^n. \tag{5.3}$$

The objective function of this problem is saw-shape. Let

$$a_i = \min_{p\in P} a_i^p, \quad b_i = \max_{p\in P} a_i^p, \qquad (i \in I).$$

and $(\bar{x}^1, \bar{x}^2)$ is a solution of problem (5.3). It is clear that $a_i \leqslant (\bar{x}^q)_i \leqslant b_i$ for $q = 1, 2$. Hence we can use the proposed algorithm in order to find the $2n$-dimensional vector $((\bar{x}^1)_1, \dots (\bar{x}^1)_n, (\bar{x}^2)_1, \dots , (\bar{x}^2)_n)$.

The proposed method for the solution of the problem (5.3) has been applied in the study of Wisconsin Diagnostic Breast Cancer ([16]) and BUPA Liver Disorders ([11]) databases. First a special procedure for feature selection was used, which allowed us to reduce the number of variables (see [8] for details). We consider the first database with three variables and the second one with four variables, so the dimension of problem (5.3) was 7 and 9, respectively. In both cases the method allows us to obtain the descriptions of the sets under consideration which are as good as known results.

6. Results of numerical experiments

In this section we discuss results of numerical experiments for some known test problems. The section consists of two parts. First of them contains some small scale problems and detailed discussions of the results of numerical experiments with these problems. In particular we compare results obtained by means of transformed functions ψ_1 and ψ_2. Examples of problems with parametric dimension (in particular, large scale problems) and corresponding results are presented in the second part.

We used a personal computer IBM Pentium-S with CPU 150 MHz for numerical experiments. The codes have been written in Fortran-90 for MS-DOS.

REMARK 6.1. We report, in particular, the number of iterations produced by the cutting angle method. Note that the initialization of this method consists of $n + 1$ iterations. (See Step 0 of the cutting angle method and Step 0 of the proposed Algorithm.) Assume, for example, that $n = 50$ and there are 52 iterations of the cutting angle method. It means that the cutting angle method has been implemented only once after Initialization.

Small scale problems

Below we give detailed description of results of numerical experiments for some well-known small scale problems of global optimization. The function ψ_1 has been used in these experiments.

PROBLEM 6.1. (see [13], p. 246).

$$f(x) = \max\{\varphi_1(x), \varphi_2(x)\}$$

where

$$\varphi_1(x) = -1.0 + 8x_1 + 8x_2 - 32x_1x_2,$$

$$\varphi_2(x) = 3.6 - 12x_1 - 4x_3 + 4x_1x_3 + 10x_1^2 + 2x_3^2,$$

$$x \in S = \left\{ x \in \mathbb{R}^3 \mid x \geqslant 0,\ \sum_{i=1}^{3} x_i = 1 \right\}.$$

Numerical results for Problem 6.1: An initial point is $x^0 = (0, 0, 1)$ with $f(x^0) = 1.60000$. An approximate stationary point $x^1 = (0.1784, 0.0001, 0.8215)$ with $f(x^1) = 0.42742$ has been found by the DG method within 36 iterations. Then the cutting angle method was applied. This method required five iterations in order to leave the stationary point and to find a new point $x^2 = (0.4286, 0.4286, 0.1428)$ with $f(x^2) = 0.00816$. Then a global minimizer $x^* = (0.5387, 0.3705, 0.0908)$ with $f(x^*) = -0.11344$ was found by the DG method within 35 iterations.

PROBLEM 6.2. (see [13], subsection 5.5.2).

$$f(x) = -\sum_{i=1}^{10} \frac{1}{\|x - a^i\|^2 + c_i}$$

where

$$x \in S^* = \{x \in \mathbb{R}^2_+ \ : \ x_1 + x_2 \leqslant 20\}.$$

The vectors $a^i = (a_1^i, a_2^i),\ i = 1, \ldots, 10$ and the vector $c = (c_1, \ldots, c_{10})$ can be found in [13], p.256.

Numerical results for Problem 6.2: An initial point is $x^0 = (1, 1)$, $f(x^0) = -0.83971$. An approximate stationary point $x^1 = (7.9749, 1.0323)$ with $f(x^1) = -1.47140$ has been found by the DG method within 28 iterations. The cutting angle method was applied, which required 37 iterations in order to leave the stationary point and to find a new point $x^2 = (3.4189, 3.2441)$ with $f(x^2) = -1.54842$. Then a global minimizer $x^* = (3.9176, 3.9814)$ with $f(x^*) = -2.14522$ was reached by the DG method within 29 iterations.

PROBLEM 6.3. (Shubert function, see [15, 17]).

$$f(x) = \prod_{i=1}^{2} \left(\sum_{j=1}^{5} j \cos((j+1)x_i + j) \right),$$

$$-10 \leqslant x_i \leqslant 10,\ i = 1, 2.$$

Numerical results for Problem 6.3: An initial point is $x^0 = (2, 2)$, $f(x^0) = 0.67721$. The DG method calculated the first approximate stationary point $x^1 = (9.0691, -2.0072)$ with $f(x^1) = -16.2861$ within 36 iterations. A new point $x^2 = (-7.0461, 1.7874)$ with $f(x^2) = -38.66532$ has been found by the cutting angle method within 27 iterations. Then the DG method was again applied. A new approximate stationary point $x^3 = (-7.0835, 1.8057)$ with $f(x^3) = -39.58874$ was found by this method within 39 iterations. The cutting angle method required 40 iterations in order to leave the stationary point x^3 and to find a new starting point for the DG method $x^4 = (-1.6642, -7.0859)$ with $f(x^4) = -79.13844$. Finally a global minimizer $x^* = (-1.4251, -7.0853)$ with $f(x^*) = -186.73091$ was reached by the DG method within 45 iterations.

REMARK 6.2. It is known (see [15, 17] and references therein) that the Shubert function has 760 local minimizers. Different initial points lead to the calculation of different local minimizers. We have checked many initial points and for all of them the number of local minimizers which appeared, when the combination of the DG method and the cutting angle method was applied, does not exceed 2.

PROBLEM 6.4. (Shekel function, see [10, 17]).

$$f(x) = -\sum_{j=1}^{N} \frac{1}{\sum_{i=1}^{4}(x_i - a_{ij})^2 + c_j},$$

$$0 \leqslant x_i \leqslant 10, \ i = 1, 2, 3, 4.$$

We consider $N = 5, 7, 10$. An initial point is $x^0 = (0, 0, 0, 0)$. The values of $a_{ij}, i = 1, 2, 3, 4, \ j = 1, \ldots, N$ and $c_j, \ j = 1, \ldots, N$ are given, for example, in [17]. Note that the objective function of Problem 6.2 is similar to the objective function of Problem 6.4.

Results of numerical experiments for $N = 5, 7, 10$ are presented in Table 1. First the DG method was applied in order to find an approximate stationary point, then the cutting angle method was exploited and then again the DG method, which leads to a global minimizer. Correspondingly Table 1 consists of three parts. We use the following notation: DG is the DG method, CAM is the cutting angle method, m - number of iterations and f the value of the objective function.

In Table 2 results of numerical experiments with Problems 6.1, 6.2, 6.3 and 6.4 for functions ψ_1 and ψ_2 are presented. We use the following notation: *itloc* is the total number of iterations by the local method, *itcut* is the total number of iterations by the cutting angle method, *loc* is the number of computed local minimizers, *fun* is the number of the objective function evaluations, *cut* is the number of the objective function evaluations, which are used by the cutting angle method, t is CPU time. We give a CPU time which was necessary for the computation of the known global

Table 1. Numerical results for Problem 6.4

	$N = 5$		$N = 7$		$N = 10$	
	m	f	m	f	m	f
DG	1	−0.27312	1	−0.29362	1	−0.32173
	81	−5.05520	79	−5.08767	82	−5.12848
CAM	1	−5.05520	1	−5.08767	1	−5.12848
	7	−10.15320	7	−10.40282	7	−10.53628
DG	1	−10.15320	1	−10.40282	1	−10.53628
	2	−10.15320	35	−10.40294	38	−10.53640

Table 2. Comparison of results for the functions ψ_1 and ψ_2.

Problem	Function	itloc	itcut	loc	fun	cut	t
6.1	ψ_1	57	5	1	606	5	0.76
	ψ_2	46	5	1	751	15	0.05
6.2	ψ_1	35	37	1	323	78	16.23
	ψ_2	30	4	1	301	60	0.05
6.3	ψ_1	97	67	2	892	189	56.02
	ψ_2	58	5	1	540	105	0.05
6.4 ($N = 5$)	ψ_1	55	6	1	449	6	0.06
	ψ_2	55	6	1	593	150	0.06
6.4 ($N = 7$)	ψ_1	62	6	1	530	6	0.05
	ψ_2	62	6	1	674	150	0.06
6.4 ($N = 10$)	ψ_1	64	6	1	527	6	0.06
	ψ_2	64	6	1	671	150	0.06

minimum. All problems have been solved with the precision $\delta = 10^{-4}$, that at last point x^k:

$$f(x^k) - f^* \leqslant 10^{-4}$$

where f^* is the global minimum of the function f.

Results presented in Table 2 show that for Problems 6.1, 6.2 anf 6.3 the function ψ_2 allows one quickly to leave local minimizers and to obtain a good starting point for the computation of a global minimizer. For the Problem 6.4 we got almost the same results for both functions ψ_1 and ψ_2. The version of the algorithm with the function ψ_1 requires more iterations of the cutting angle method in order to leave the stationary points. Therefore it needs much more CPU time to compute a global minimizer with given precision. The version of the algorithm with the function ψ_2

requires a smaller number of iterations in order to leave the stationary points and as a rule provides a good starting point for a local search. However, it requires much more function evaluations.

Problems with parametric dimension

Here we consider three well-known test problems and also a test problem, which was introduced by the authors in [5]. We use the following notation for the description of the test problems: f is the objective function, x^0 is the starting point, x^* is the global minimum point.

PROBLEM 6.5. *(Griewank, see [12] and also [17])*

$$f(x) = \frac{1}{d}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1,$$

$$\sum_{i=1}^{n} x_i \leqslant 400,\ x_i \geqslant -50,\ i = 1,\dots,n,\ d = 4000,$$

$$x^0 = (x_1^0,\dots,x_n^0),\ x_i^0 = -n,\ i = 1,\dots,n,\ x^* = (0,\dots,0),\ f(x^*) = 0.$$

PROBLEM 6.6. (Levy, see [14] and also [17]).

$$\begin{aligned} f(x) = {}& \sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2(1 + 10\sin^2(\pi y^i + 1)) \\ & + (y_n - 1)^2(1 + \sin^2(2\pi x_n)), \end{aligned}$$

$$y_i = 1 + \frac{x_i - 1}{4},\quad \sum_{i=1}^{n} x_i \leqslant 70,\ x_i \geqslant -1,\ i = 1,\dots,n,$$

$$x^0 = (0,\dots,0),\ x^* = (1,\dots,1),\ f(x^*) = 0.$$

PROBLEM 6.7. (Bagirov and Rubinov, see [5]).

$$f(x) = \sum_{i=1}^{n}\min\{0, 10\|x - a^i\| - b_i\}, \qquad (n \geqslant 2), \tag{6.1}$$

$$\sum_{i=1}^{n} x_i = 1,\ x_i \geqslant 0,\ i = 1,\dots,n,$$

where $\|\cdot\|$ *is 2-norm,* a^i, $i = 1, \dots, n$ *are* n*-vectors with coordinates*

$$a^i_j = \begin{cases} (n+1)/2n & \text{if} \quad j = i, \\ 1/2n & \text{if} \quad j \neq i. \end{cases}$$

$$b_1 = 4, \; b_i = b_{i-1} - \frac{2}{n-1}, \; i = 2, \dots, n.$$

$$x^0 = (0, \dots, 0, 1), \; x^* = a^1 \equiv \left(\frac{n+1}{2n}, \frac{1}{2n} \dots, \frac{1}{2n} \right), \; f(x^*) = -4.$$

PROBLEM 6.8. (Pardalos, Ye and Han, see [19]).

$$f(x) = \sum_{i=1}^{k-1} \left(x_i - \frac{r_i}{r_{i+1}} x_{i+1} \right)^2 + \sum_{i=k+1}^{n} x_i^2 - \left(\sum_{i=1}^{k} x_i \right)^2,$$

$$\sum_{i=1}^{n} x_i = 1, \; x_i \geqslant 0, \; i = 1, \dots, n, k = [n/2] + 1,$$

$$r_i = \frac{r_i'}{\sum_{i=1}^{k} r_i'} \quad \textit{where} \quad r_i' = 5|sin(i)| + 0.1, \; i = 1, \dots, k.$$

$$x^0 = (0, \dots, 0, 1), \; x^* = (r_1, \dots, r_k, 0, \dots, 0), \; f(x^*) = -1.$$

REMARK 6.3. Problem 6.7 can serve as a certain simplification of problem (5.3). Note that the function (6.1) is a saw-shape and has very many local minimizers. For example, for $n = 3$ the number of the points of local minima is 7 and for $n = 4$ it is 15.

Results of numerical experiments are presented in Table 3. Here we use the same notation as in Table 2; n is the number of variables. We present results of numerical experiments only with the function ψ_2. For the function ψ_1 corresponding results are essentially worse. For example, for the Problem 6.7 with $n = 5$ the CPU time, related to the version of the Algorithm with the transformed function ψ_1, was 23.18 sec., which is about 210 times more than the corresponding time for the version, related to the transformed function ψ_2.

PROBLEM 6.9. *In [9] the proposed method has been applied for solving data classification problems. The minimization of (5.3) was used for this purpose. For a stopping criterion we restricted the number of iterations generated by the cutting angle method by 70 (see Remark 4.3). We cannot assert that we obtain a global solution to the problem (5.3). However, for all databases under consideration the results obtained by the proposed method are better than those obtained by only a local method.*

Table 3. Numerical results for Problems 6.5, 6.6, 6.7 and 6.8.

Problem	n	itloc	itcut	loc	fun	cut	t
6.5	5	119	5	1	1134	25	0.28
	10	192	12	1	4092	60	0.71
	20	1085	22	1	34605	120	3.95
	30	1121	30	1	50531	150	5.55
	50	801	50	1	61151	250	14.11
6.6	5	65	12	1	1848	1110	0.58
	10	78	12	1	2331	660	0.22
	20	71	22	1	4952	2310	0.43
	30	59	32	1	8076	4960	0.88
	50	61	52	1	27323	13260	8.45
6.7	5	45	5	1	936	25	0.11
	10	86	10	1	4150	50	0.50
	20	55	20	1	14419	100	3.19
	30	48	30	1	30637	150	11.36
	50	49	50	1	48334	250	41.31
6.8	5	72	1	1	1932	5	0.11
	10	486	5	2	37908	22	4.95
	20	173	0	0	29934	0	62.84
	30	156	0	0	39826	0	74.34
	70	1097	70	1	119961	350	286.56

7. Concluding remarks

Here we present some conclusions, which follow from the numerical experiments.
1. Results of numerical experiments show that the proposed method is able to leave stationary points and to find a global minimizer. Moreover the cutting angle method does not sort all local minimizers. The combination of DG and the cutting angle method is able to jump over many local minima. No more than two local minimizers different from the global one have appeared at each example under consideration.
2. When a stationary point y^i is known, we consider a new transformed function ψ^i (with respect to this point), which depends on y^i. A small amount of min-type functions is used at some first iterations of the cutting angle method. So this method can rather quickly find a point, where value of the objective function is less than at the stationary point. Nevertheless numerical experiments show that the cutting angle method is much more time-consuming than a local search by DG.

3. Let

$$c \geqslant 2L - \min_{x \in S} f(x) \tag{7.1}$$

where L is the Lipschitz constant of f. Then (see Theorem 2.1) the function $f(x)+c$ is the restriction of an IPH function to S. The lower bound $2L - \min_{x\in S} f(x)$ is unknown. It follows from (7.1) that we can consider a sufficiently large number as c. However, numerical experiments show that large values of c lead to slow convergence of the cutting angle method. Thus we need to find a not very large number c such that (7.1) holds. Numerical experiments also demonstrate that we can take a fairly small number c, when the cutting angle method is considered only as means for leaving a local minimizer. Let L_i be a Lipschitz constant of a transformed function ψ^i, which is constructed with respect to point y^i. Then $L_{i+1} \leqslant L_i \leqslant L$. If $L_{i+1} < L_i$ then we can decrease c.
4. A version of Algorithm has been studied, where the function f was used as a transformed function ψ. Numerical experiments show that this version requires much more iterations of the cutting angle method.
5. Various transformed functions can be used in order to replace the objective function. We considered functions ψ_1 and ψ_2 in numerical experiments. Results of numerical experiments allow us to assert that the effectiveness of the proposed method strongly depends on these functions. This dependence becomes obvious when the number of variables increases. The use of the second function allowed us to obtain a fairly good starting point for a local search. An interesting and important problem is describe and examine new transformed functions. This problem is the subject of the further research.
6. Numerical experiments show that the convergence of the sequence of lower estimates λ_k, which are constructed by the cutting angle method, is very slow, if the objective function is constant. So, if the global minimizer y_* has already been obtained, the confirmation of the fact that y_* is really a global minimizer, requires very much computational time. This is one of the main drawbacks of the proposed method. So, the question arises how to recognize that a given function is constant?

If the record value f_i of the function f at the i-th iteration is close to the global minimum f_*, then the relative range

$$\frac{\max_{x\in S} \psi^i(x) - \min_{x\in S} \psi^i(x)}{\max_{x\in S} \psi^i(x)}$$

of the function ψ^i is small, (especially, if the number c, which we add to this function in order to apply the cutting angle method, is large) so this function is almost constant. The cutting angle method works slowly in such a situation.
7. Numerical experiments with some well-known Lipschitz problems demonstrate that the proposed approach allows us to solve fast enough many problems up to $n = 50$ variables, using a personal computer IBM Pentium-S with CPU 150 MHz.

Acknowledgement

The authors would like to thank two anonymous referees for their comments which improved the quality of the paper.

References

1. Andramonov, M. Yu, Rubinov, A.M. and Glover, B.M. 1997, Cutting angle method for minimizing increasing convex-along-rays functions, Research Report 97/7, SITMS, University of Ballarat.
2. Andramonov, M.Yu, Rubinov, A.M. and Glover, B.M. 1999, Cutting angle method in global optimization, *Applied Mathematics Letters* 12, 95–100.
3. Bagirov, A.M. 1999, Minimization methods for one class of nonsmooth functions and calculation of semi-equilibrium prices. in Ser. *Applied Optimization*, Vol. 30: *Progress in Optimization: Contributions from Australasia*, A. Eberhard et al. (eds.), Kluwer Academic Publishers, 147–175.
4. Bagirov, A.M. 1999, Derivative-free methods for unconstrained nonsmooth optimization and its numerical analysis. *Investigacao Operacional*, 19, 75–93.
5. Bagirov, A.M. and Rubinov, A.M. 2000, Global minimization of increasing positively homogeneous functions over the unit simplex, *Annals of Operations Research* 98, 171–187.
6. Bagirov, A.M. and Rubinov, A.M. 2001, Modified versions of the cutting angle method, In: *Nonconvex Optimization and Its Applications*, Vol. 54, *Advances in Convex Analysis and Global Optimization*, N. Hadjisavvas and P.M. Pardalos (eds.), Kluwer Academic Publishers, Dordrecht, 245–268.
7. Bagirov, A.M. and Rubinov, A.M. 2001, Global optimization of marginal functions with applications to economic equilibrium. Journal of Global Optimization, 20(3–4), 215–237.
8. Bagirov, A.M., Rubinov, A.M., Stranieri, A. and Yearwood, J. 1999, The global optimization approach to the clustering analysis. Research Report, 45/99, University of Ballarat, Australia.
9. Bagirov, A.M., Rubinov, A.M. and Yearwood, J. 2001, A global optimization approach to classification. Research Report, 01/4, University of Ballarat, Australia.
10. Brachetti, P., De Felice Ciccoli, M., Di Pillo, G. and Lucidi, S. 1997, A new version of the Price's algorithm for global optimization. *Journal of Global Optimization*, 10, 165–184.
11. BUPA liver disorders database. Publicly available from: www.sgi.com/Technology/mlc/db.
12. Griewank, A. 1981, Generalized descent for global optimization. *Journal of Optimization Theory and Applications* 34, 11–39.
13. Horst, R., Pardalos, P.M. and Thoai, N.V. 1995, *Introduction to Global Optimization*, Nonconvex Optimization and Its Applications, Vol. 3, Kluwer Academic Publishers, Dordrecht.
14. Levy, A.V. and Montalvo, A. 1985, The tunnelling algorithm for the global minimization of functions. *SIAM Journal on Scientific and Statistical Computing* 6, 15-29.
15. Lucidi, S. and Piccioni, M. 1989, Random tunnelling by means of acceptance-rejection sampling for global optimization. *Journal of Optimization Theory and Applications* 62(2).
16. Mangasarian, O.L., Street, W.N. and Wolberg, W. H., Breast cancer diagnosis and prognosis via linear programming, *Operations Research* 43(4), 570–577.
17. Neumaier, A., Optimization test problems. Publicly available from: solon.cma.univie.ac.at/ñeum/glopt.html.
18. Pallaschke, D. and Rolewicz, S. 1997, *Foundations of Mathematical Optimization (Convex Analysis without Linearity)*, Kluwer Academic Publishers, Dordrecht.
19. Pardalos, P.M., Yinyu Ye and Chi-Geun Han, 1991, Algorithms for the solution of quadratic Knapsack problems. *Linear Algebra and its Applications* 152, 69–91.

20. Rubinov, A.M. 2000, *Abstract Convexity and Global Optimization*, Kluwer Academic Publishers, Dordercht.
21. Rubinov, A. and Andramonov, M. 1999, Lipschitz programming via increasing convex-along-rays functions, *Optimization Methods and Software* 10, 763–781.