Journal of Global Optimization **18:** 265–274, 2000.

On Covering Methods for D.C. Optimization ⋆

RAFAEL BLANQUERO and EMILIO CARRIZOSA
Departamento de Estadística e Investigación Operativa, Universidad de Sevilla, Sevilla, Spain
(e-mail: rblanque@cica.es; ecarriz@cica.es)

(Received 21 January 1999; accepted in revised form 19 May 2000)

Abstract. Covering methods constitute a broad class of algorithms for solving multivariate Global Optimization problems. In this note we show that, when the objective f is d.c. and a d.c. decomposition for f is known, the computational burden usually suffered by multivariate covering methods is significantly reduced. With this we extend to the (non-differentiable) d.c. case the covering method of Breiman and Cutler, showing that it is a particular case of the standard outer approximation approach. Our computational experience shows that this generalization yields not only more flexibility but also faster convergence than the covering method of Breiman-Cutler.

Key words: Covering methods, d.c. Functions, Outer approximation, Power diagrams

1. Introduction

Given a polytope $X \subset \mathbb{R}^m$ and a real-valued function f defined on X, we consider the optimization problem

$$\max_{x \in X} f(x) \tag{1}$$

Covering algorithms, [4, 7, 10, 12, 17, 19], constitute an important class of methods for solving problems of type (1) for very large classes of functions f. In essence, covering methods build a feasible sequence $\{x_n\}$ and a sequence of *upper envelopes* $\{E^{(n)}\}$, i.e., functions satisfying

$$\begin{array}{lcll} E^{(n)} & \geqslant & f & \text{for all } n \\ E^{(n)} & \geqslant & E^{(n+1)} & \text{for all } n \\ E^{(n)}(x_j) & = & f(x_j) & \text{for all } j = 1, \ldots, n \end{array}$$

If the detection of an ε-optimal solution is used as stopping rule, the prototype covering algorithm is given in Algorithm 1, [16].

ALGORITHM 1. *Initialization*
Take $x_1 \in X$
Set $\underline{f} = f(x_1)$, $x_* = x_1$ *and construct* $E^{(1)}$

⋆ This research has been supported by Grant PB96-1416-C02-02 of DGES, Spain

Iteration n = 1,2, . . .
Set $\overline{f} = \max_{x\in X} E^{(n)}(x)$ *and select* $x_{n+1} \in \arg\max_{x\in X} E^{(n)}(x)$.
If $\overline{f} - \underline{f} \leq \varepsilon$ *then Stop.*
Construct $E^{(n+1)}$ *from* $E^{(n)}$.
If $f(x_{n+1}) > \underline{f}$ *then set* $\underline{f} = f(x_{n+1})$ *and* $x_* = x_{n+1}$
Go to iteration $n + 1$.

Whereas the theoretical convergence of this algorithm can be established for general instances from e.g. [16] (Theorem II.2) or [19], (the algorithm finishes with x_* as an ε-optimal solution and $\underline{f}$ as a lower bound differing less than ε from the optimal value), care must be taken in order to make these procedure work in practice. Indeed, two critical aspects must be considered, namely, how the upper envelopes $E^{(n)}$ are defined, and then how the subproblems (Q_n),

$$\max_{x\in X} E^{(n)}(x), \tag{Q_n}$$

are solved.

The upper envelopes $E^{(n)}$ are usually constructed as pointwise minimum of more simple functions k_i,

$$E^{(n)} = \min_{1\leq i\leq n} k_i \tag{2}$$

For instance, if f is Lipschitz with known Lipschitz constant L, Piyavskii algorithm, [19, 14, 16] uses k_i^{PIY} given by

$$k_i^{PIY}(x) = f(x_i) + L\|x - x_i\|$$

Another example of covering method based on (2) is provided in [3, 4, 7, 10]: assuming $f : X \subset \mathbb{R}^m \longmapsto \mathbb{R}$ is continuously differentiable and satisfies the condition

$$f(x) \leq f(y) + \nabla f(y)'(x - y) + K\,\|x - y\|^2 \quad \forall\, x, y \in X \tag{3}$$

where $K \geq 0$ is a known constant, one defines k_i^{BrC} as

$$k_i^{BrC}(x) = f(x_i) + \nabla f(x_i)'(x - x_i) + K\,\|x - x_i\|^2 \tag{4}$$

This is generalized in a later paper, [4], to

$$k_i^{BC}(x) = f(x_i) + \nabla f(x_i)'(x - x_i) + q(x - x_i)$$

where $q(x) = x'Hx$ is a quadratic form with positive definite matrix H and k_i^{BC} satisfies, for every choice of $x_i \in X$, the condition

$$k_i^{BC}(x) \geq f(x) \quad \forall\, x \in X$$

that is,

$$f(x_i) + \nabla f(x_i)'(x - x_i) + q(x - x_i) \geq f(x) \quad \forall\, x_i, x \in X \tag{5}$$

Note that (3) and k_i^{BrC} are particular cases of condition (5) and k_i^{BC}, respectively, taking as H the diagonal matrix with value K on the diagonal.

Solving the subproblems (Q_n) seems to be the bottleneck point of the procedures. In fact, each (Q_n) is itself a Global-Optimization (in general non-differentiable) problem, whose resolution seems to be far from trivial. Indeed, it is mentioned in [16] (page 596) that the Piyavskii algorithm seems to be not very promising in dimensions higher than 2 due to the increasing difficulty in solving (Q_n). The structure of the problems (Q_n) is more simple for those coverings based on (4); however, bounds on the first and second derivatives of f are required, which, as mentioned in [4], is often an equally difficult global optimization problem as the original.

In the next section we show that, when a d.c. decomposition of f is known, a new class of upper envelopes can be described. These envelopes will yield subproblems (Q_n) with a structure similar to those based on (4), and, at the same time, are more versatile since they can avoid the calculus of the bounds of derivatives and can also be used for non-differentiable problems.

2. Covering algorithm for d.c. functions

Let f be a real-valued d.c. function on X, i.e., f can be represented as the difference of two functions convex on X, and assume that a d.c. decomposition of f is known,

$$f(x) = f_1(x) + f_2(x) \text{ for each } x \in X,$$

where f_1 is convex on X and f_2 is concave on a convex open set $\Omega \supset X$.

We recall the reader that, although for an arbitrary d.c. function f, it is not easy to obtain such a d.c. decomposition, a series of rules enables one to obtain d.c. decompositions of functions f which are obtained (via summation, pointwise maximum, linear combinations, ...) from more simple d.c. functions, [16, 21].

2.1. THE UPPER ENVELOPE AND PARTICULAR CASES

In this case, by majorizing the concave part of the decomposition, we obtain the following convex bounding function:

$$k_i^{dc}(x) = f_1(x) + (f_2(x_i) + \xi_2(x_i)'(x - x_i)) \tag{6}$$

where $\xi_2(x_i)$ is an arbitrary subgradient of f_2 at x_i.

In particular, the method of Breiman–Cutler [7] and the generalization due to Baritompa–Cutler [4] are particular cases of our approach (6). Indeed, one has

PROPOSITION 2.1. *Let $f : X \subset \mathbb{R}^m \to \mathbb{R}$ satisfy (5) for a quadratic form q with symmetric positive semidefinite matrix H. Then, f is a d.c. function and*

$$f(x) = x'Hx + (f(x) - x'Hx) \tag{7}$$

is a d.c. decomposition for f. Moreover, if (7) is used as d.c. decomposition of f, then Algorithm 1 with envelopes k_i^{dc} yields the algorithm of Baritompa–Cutler.

Proof. Since $f_1(x) = x'Hx$ is a convex function, it suffices to show that the differentiable function $f_2(x) = f(x) - x'Hx$ is concave.

By (5), it follows that

$$\begin{aligned} f(x) &\leqslant f(y) + \nabla f(y)'(x-y) + (x-y)'H(x-y) \\ &= f(y) + \nabla f(y)'(x-y) + x'Hx - y'Hy - 2y'H(x-y) \end{aligned}$$

Rearranging terms,

$$\begin{aligned} f(x) - x'Hx &\leqslant f(y) - y'Hy + (\nabla f(y) - 2Hy)'(x-y) \\ f_2(x) &\leqslant f_2(y) + \nabla f_2(y)'(x-y) \end{aligned},$$

from which we conclude that f_2 is concave.

Moreover,

$$\begin{aligned} k_i^{dc}(x) &= f_1(x) + f_2(x_i) + \nabla f_2(x_i)'(x - x_i) \\ &= x'Hx + (f(x_i) - x_i'Hx_i) + (\nabla f(x_i) - 2Hx_i)'(x - x_i) \\ &= f(x_i) + \nabla f(x_i)'(x - x_i) + x'Hx - x_i'Hx_i - 2x_i'H(x - x_i) \\ &= f(x_i) + \nabla f(x_i)'(x - x_i) + (x - x_i)'H(x - x_i) \\ &= k_i^{BC}(x), \end{aligned}$$

showing that k^{dc} yields the envelope of Baritompa–Cutler when (7) is used as d.c. decomposition of f. □

Hence, even for functions f satisfying (5), different envelopes can be obtained as soon as a d.c. decomposition, other than (7), is provided. This may avoid the costly process of bounding derivatives, [4] and may affect the convergence, as depicted in Example 2.2 and Section 3 (see [6] for further details).

EXAMPLE 2.2. *In the paper [7], Breiman and Cutler use $f : I = [-0.2, 1] \longmapsto \mathbb{R}$ defined as*

$$f(x) = \cos(5\pi x) - x^2$$

This function is twice continuously differentiable, thus it admits a decomposition of the form

$$f(x) = Kx^2 - (Kx^2 - f(x)) \tag{8}$$

In particular, Breiman and Cutler propose such decomposition taking $K = 12.5\pi^2 - 1$ (the smallest value for which $Kx^2 - f(x)$ is convex).

However, an alternative d.c. decomposition is provided in [5]: given $\hat{x} \in I$,

$$f(x) = f_1(x) + f_2(x) \tag{9}$$

where

$$f_1(x) = f(\hat{x}) + f'(\hat{x})(x - \hat{x}) + \int_{\hat{x}}^{x} (x - t)[f''(t)]^+ dt$$

$$f_2(x) = -\int_{\hat{x}}^{x} (x - t)[f''(t)]^- dt$$

being $A^+ = \max(A, 0)$ *and* $A^- = \min(-A, 0)$.

This yields the d.c. decomposition

$$f(x) = (f(x) + h(x)) - h(x),$$

where

$$h(x) = \begin{cases} h_1(x) & x \in [z_1, a_1] \\ h_2(x) & x \in [a_1, b_1] \\ h_3(x) & x \in [b_1, a_2] \\ h_4(x) & x \in [a_2, b_2] \\ h_5(x) & x \in [b_2, a_3] \\ h_6(x) & x \in [a_3, b_3] \\ h_7(x) & x \in [b_3, z_2] \end{cases}$$

$$\begin{aligned}
h_1(x) &= 1 + 5\pi(x - a_1)\sin(5\pi a_1) - \cos(5\pi a_1) + a_1(2x - a_1) \\
h_2(x) &= 1 - \cos(5\pi x) + x^2 \\
h_3(x) &= 1 + 5\pi(x - b_1)\sin(5\pi b_1) - \cos(5\pi b_1) + b_1(2x - b_1) \\
h_4(x) &= h_2(x) + h_3(x) - 1 - 5\pi(x - a_2)\sin(5\pi a_2) + \cos(5\pi a_2) - a_2(2x - a_2) \\
h_5(x) &= 1 + h_4(x) - h_2(x) + 5\pi(x - b_2)\sin(5\pi b_2) - \cos(5\pi b_2) + b_2(2x - b_2) \\
h_6(x) &= h_2(x) + h_5(x) - 1 - 5\pi(x - a_3)\sin(5\pi a_3) + \cos(5\pi a_3) - a_3(2x - a_3) \\
h_7(x) &= 1 + h_6(x) - h_2(x) + 5\pi(x - b_3)\sin(5\pi b_3) - \cos(5\pi b_3) + b_3(2x - b_3)
\end{aligned}$$

$$\begin{aligned}
z_1 &= -0.2 \qquad & a_i &= \tfrac{-1}{5\pi}\arccos(\tfrac{-0.08}{\pi^2}) + \tfrac{2}{5}(i-1) \quad i = 1, 2, 3 \\
z_2 &= 1 \qquad & b_i &= \tfrac{1}{5\pi}\arccos(\tfrac{-0.08}{\pi^2}) + \tfrac{2}{5}(i-1) \quad i = 1, 2, 3
\end{aligned}$$

This alternative envelope yields sharper bounds than (8), as depicted in Figure 1, which shows the envelopes derived from (8) and (9) after four iterations of the algorithms using $x_1 = 0.5$ *as starting point.*

For the sake of completeness, we have compared in Figure 2 the d.c. envelope from (9) and the Piyavskii envelope (using the quite sharp Lipschitz constant $L = 17.51$*) again after four iterations using* $x_1 = 0.5$ *as starting point.*

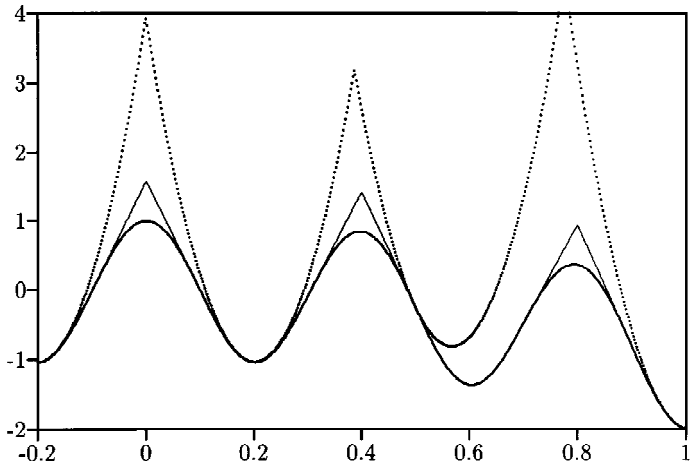

Figure 1. Comparison of envelopes I: Example function (double-wide line), Breiman–Cutler envelope (dotted line), d.c. envelope (single-wide line)

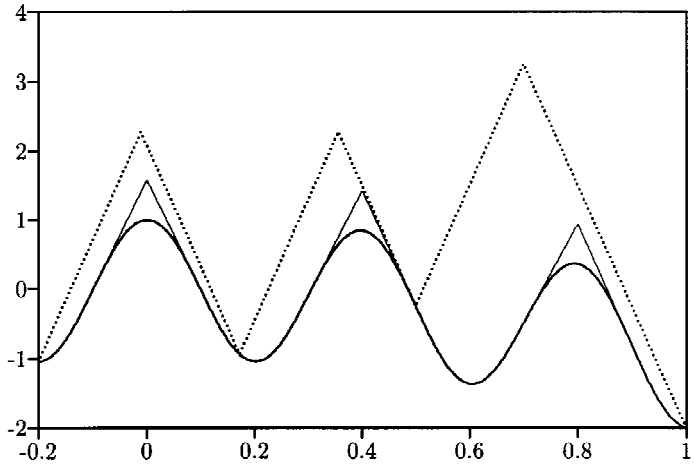

Figure 2. Comparison of envelopes II: Example function (double-wide line), Piyavskii envelope (dotted line), d.c. envelope (single-wide line)

2.2. SOLVING (Q_n)

2.2.1. *Convex maximization*

For the upper envelope (6), (Q_n) can be formulated as a convex maximization problem with linear constrains, e.g. [8]. Indeed, (Q_n) is equivalent to

$$\max_{(x,t)\in B^{(n)}} \{f_1(x) + t\},$$

where

$$B^{(n)} = \left\{(x,t) : x \in X \ , \ t \leq f_2(x_i) + \xi_2(x_i)'(x - x_i) \quad \forall\, i \ , \ 1 \leq i \leq n\right\}$$

Hence, solving each (Q_n) amounts to maximizing a convex function over the polyhedron $B^{(n)}$. Moreover, since we obtain $B^{(n+1)}$ from $B^{(n)}$ by adding a linear constraint, the resolution of (Q_n) can be used to solve (Q_{n+1}) if procedures for on-line enumeration of vertices, [9], are used.

Moreover, the insertion of each new constraint can be done in polynomial time (the dimension m considered to be fixed), since, at the n-th iteration, $O(n^{m-1})$ time suffices to update the geometry of the feasible region, see [1, 11].

In other words, the covering Algorithm 1 for the d.c. case (and, in particular, the Breiman–Cutler method) is simply an outer approximation algorithm.

2.2.2. *Power diagrams*

A different though equivalent strategy for optimizing (Q_n) is proposed in [7] for the particular case (4): defining for each i, n, $D_{i:n}^{BrC}$ as

$$D_{i:n}^{BrC} = \left\{x \in \mathbb{R}^m : k_i^{BrC}(x) \leq k_j^{BrC}(x) \quad \forall\, j\ ,\ 1 \leq j \leq n\right\},$$

one immediately obtains that each $D_{i:n}^{BrC}$ is polyhedral. Moreover, since (Q_n) can be rewritten as

$$\max_{1\leq i\leq n}\left\{\max_{x\in X\cap D_{i:n}^{BrC}} k_i^{BrC}(x)\right\}, \tag{10}$$

solving (Q_n) amounts to inspecting the vertices of each polytope $X \cap D_{i:n}^{BrC}$, for which a geometrical method is detailed in [7].

This strategy is applicable not only to covering methods derived from k^{BrC}, but also for the more general case in which the envelopes are derived from k^{dc}. Indeed,

$$\begin{aligned} k_i^{dc}(x) \leq k_j^{dc}(x) \ &\Leftrightarrow\ f_2(x_i) + \xi_2(x_i)'(x - x_i) \leq f_2(x_j) + \xi_2(x_j)'(x - x_j) \\ &\Leftrightarrow\ \|x - C_i\|^2 + \beta_i \leq \left\|x - C_j\right\|^2 + \beta_j \end{aligned}$$

where

$$C_k = \frac{-1}{2}\xi_2(x_k) \qquad \beta_k = f_2(x_k) - \xi_2(x_k)'x_k - \frac{1}{4}\left\|\xi_2(x_k)\right\|^2 \quad k = i, j$$

thus, defining $D_{i:n}^{dc}$ as

$$D_{i:n}^{dc} = \left\{x \in \mathbb{R}^m : k_i^{dc}(x) \leq k_j^{dc}(x) \quad \forall j\ ,\ 1 \leq j \leq n\right\},$$

it turns out that $\left\{\{D_{i:n}^{dc}\}_{1\leq i\leq n}\right\}$ is a *power diagram*, with centers $C_1, \ldots, C_n$ and (additive) weights $\beta_1, \ldots, \beta_n$, [1, 2, 11, 18].

The key property of power diagrams is the fact that they induce a *polyhedral* subdivision of $\mathbb{R}^m$ (as the particular case $\{D_{i:n}^{B_rC}\}$), and can be constructed by on-line procedures, [2]. In fact, the geometrical procedure described in [7] for solving (Q_n) is basically the same than the on-line procedure of [2] for the (at first glance unrelated) problem of describing the geometrical structure of a power diagram.

Table 1. Number of iterations for unidimensional test problems

No.	Piyavskii	Breiman–Cutler	DC covering
1	149	25	12
2	155	21	14
3	195	103	54
4	413	24	14
5	151	32	15
6	129	37	11
7	153	24	15
8	185	85	53
9	119	24	14
10	203	24	14
11	373	44	28
12	327	42	29
13	993	264	15
14	145	29	13
15	629	80	16
16	497	88	15
17	549	67	26
18	303	19	14
19	131	20	14
20	493	30	11

A geometric-based strategy is also possible for envelopes based on k^{PIY} for Lipschitz optimization. Indeed, one can define $D_{i:n}^{PIY}$, and obtain an expression similar to (10). This is the basis of the Jaumard–Herrmann–Ribault algorithm described in [14]. The so defined sets $D_{i:n}^{PIY}$ may still be identified with geometrical objects, namely, *additive Voronoi diagrams*, [1, 11, 18], but (for dimension greater or equal to 2) are no longer polyhedral, thus more complicated data structures are needed in order to store and update the geometry of such diagram.

3. Computational experience

In this section we present some numerical results, which show that the d.c. decomposition chosen in (6) may have a great influence on the convergence of the algorithm. See also [6] for further results.

We have studied the performance of Algorithm 1 using a set of well-known test problems from the literature, collected in [7, 13]. Table 1 shows the number of iterations spent by the three covering methods addressed throughout the paper

Table 2. Computational results for multidimensional test problems

Example	Breiman–Cutler			DC covering		
	Iterations	CPU standard time	CPU seconds	Iterations	CPU standard time	CPU seconds
COS2	77	2.6404	0.0059	27	1.7696	0.0040
COS4	1392	1416.7967	3.1776	201	250.8309	0.5748
H3	2575	443.2186	1.0094	442	117.6784	0.2648

for twenty one-dimensional test problems (see Table 1 of [13]). The computational results concerning to the Piyavskii and Breiman–Cutler methods are taken from [20], where the exact value of the associated parameters L and K have been used, as well as an accuracy of value $\varepsilon = 0.0001\ (b-a)$, with $[a, b]$ being the interval where the test function is defined.

The computational implementation of Algorithm 1 has been carried out in every case using the d.c. decomposition given in Section 6 of [15] for the non-$\mathcal{C}^2$ function 18 and the d.c. decomposition (9) for the remaining instances. The starting point for the algorithm has been always taken equal to the middle point of the interval. We just provide the number of iterations since the CPU times for our algorithm are negligible.

Finally, we have considered three multidimensional test problems from [7, 10] (COS2, COS4 and H3 in Table 3 of [7]), where the d.c. representations used are based on decomposition (9) for each term in COS2 and COS4, and the decomposition suggested by Proposition 3.5 of [22] for H3. The stopping rule and the tolerance values ε_1 and ε_2 has been taken as in [7]. In Table 2 we show the number of iterations and the CPU time (measured in units of standard time and seconds) taken by our algorithm in order to solve these problems.

We just need to say that the implementation of Algorithm 1 used in these examples is directly based on the program listings of the Breiman–Cutler algorithm provided in [10]. Both of them have been programmed in Fortran 90 in a personal computer with a 200 Mhz. Pentium processor and 16 Megabytes of RAM memory.

4. Acknowledgements

The authors want to thank Dr. J. Fliege for his helpful remarks.

References

1. Aurenhammer, F. (1987), Power diagrams: Properties, algorithms and applications, *SIAM J. on Computing* 16: 78–96.
2. Aurenhammer, F. (1991), Voronoi Diagrams – A Survey of a Fundamental Geometric Data Structure, *ACM Computing Surveys* 23, 345–495.

3. Baritompa, W. (1993), Customizing Methods for Global Optimization – A Geometric Viewpoint, *J. of Global Optimization* 3, 193–212.
4. Baritompa, W. and Cutler, A. (1994), Accelerations for Global Optimization Covering Methods Using Second Derivatives, *J. of Global Optimization* 4: 329–341.
5. Bittner, L. (1970), Some Representation Theorems for Function and Sets and their Application to Nonlinear Programming, *Numerische Mathematik* 16: 32–51.
6. Blanquero, R. (1999), *Localización de servicios en el plano mediante técnicas de optimización d.c.*, Unpublished PhD. Thesis, Universidad de Sevilla, Spain.
7. Breiman, L. and Cutler, A. (1993), A deterministic algorithm for global optimization, *Mathematical Programming* 58: 179–199.
8. Carrizosa, E. and Plastria, F. (1998), Locating and undesirable facility by generalized cutting planes, *Mathematics of Operations Research* 23: 680–694.
9. Chen, P., Hansen, P. and Jaumard, B. (1991), On-line and Off-line Vertex Enumeration by Adjacency Lists, *Operations Research Letters* 10: 403–409.
10. Cutler, A. (1988), *Optimization methods in Statistics*, PhD. Thesis, University of California, Berkeley.
11. Edelsbrunner, H. (1987), *Algorithms in Combinatorial Geometry*, Springer-Verlag, Berlin.
12. Hansen, P., Jaumard, B. and Lu, S. (1992), Global optimization of univariate Lipschitz functions: I. Survey and properties, *Mathematical Programming* 55: 251–272.
13. Hansen, P., Jaumard, B. and Lu, S. (1992), Global optimization of univariate Lipschitz functions: II. New algorithms and computational comparison, *Mathematical Programming* 55: 273–292.
14. Hansen, P. and Jaumard, B. (1995), Lipschitz optimization, in Horst, R. and Pardalos, P.M. (eds.), *Handbook of Global Optimization*, Kluwer Academic Publishers, Dordrecht.
15. Hartman, P. (1959), On functions representable as a difference of convex functions, *Pacific J. of Mathematics* 9: 707–713.
16. Horst, R. and Tuy, H. (1996), *Global Optimization: Deterministic Approaches*, Springer-Verlag, Berlin.
17. Mladineo, R.H. (1986), An algorithm for finding the global maximum of a multimodal multivariate function, *Mathematical Programming* 34: 188–200.
18. Okabe, A., Boots, B. and Sugihara, K. (1995), *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, John Wiley & Sons, Chichester.
19. Piyavskii, S.A. (1972), An algorithm for finding the absolute extremum of a function, *USSR Computational Mathematics and Mathematical Physic* 12: 57–67.
20. Sergeyev, Ya.D. (1998), A Method Using Local Tuning for Minimizing Functions with Lipschitz Derivatives, in Bomze, I.M., Csendes T., Horst, R. and Pardalos, P.M. (eds.), *Developments in Global Optimization*, Kluwer Academic Publishers, Dordrecht.
21. Tuy, H. (1995), D.C. Optimization: Theory, Methods and Algorithms, in Horst, R. and Pardalos, P.M. (eds.), *Handbook of Global Optimization*, Kluwer Academic Publishers, Dordrecht.
22. Tuy, H. (1998), *Convex Analysis and Global Optimization*, Kluwer Academic Publishers, Dordrecht.