REVUE D'ANALYSE NUMÉRIQUE ET DE THÉORIE DE L'APPROXIMATION
Rev. Anal. Numér. Théor. Approx., vol. 37 (2008) no. 1, pp. 59–70
ictp.acad.ro/jnaat

NUMERICAL METHODS FOR SOLVING UNIMODAL MULTIPLE CRITERIA OPTIMIZATION PROBLEMS – A SYNTHESIS†

LIANA LUPŞA* and IOANA CHIOREAN*

Abstract. In this paper we give a general method to approximate the set of all efficient solutions and the set of all weakly-efficient solutions for a multiple criteria optimization problem involving generalized unimodal objective functions on the feasible sets. This type of problems appear frequently in Economy, Mathematics, sometimes in Medico-Economics studies, etc.

MSC 2000. 90C29, 90C10, 65H05, 66Y05.

Keywords. Generalized unimodal functions, minimum points, multiple criteria programming, efficient points, weakly efficient points, serial and parallel calculus.

1. INTRODUCTION

Starting with an overview of the existing algorithms, the present paper intends to give a general algorithm which compute the set of optimal solution of an optimization problem, when the set on which the function is unimodal is any compact subset of $\mathbb{R}$ (not necessarily interval or discreet set). A parallel approach for this algorithm is also given.

Let $f = (f_1, \dots, f_m) : D \to \mathbb{R}^m$ ($m \in \mathbb{N}^*$, $m \geqq 2$) be a vector-valued function defined on a nonempty set $D \subset \mathbb{R}$ and S a subset of D. Consider the multiple criteria optimization problem:

$$(1) \qquad (MOP) \begin{cases} \text{Minimize} & f(x) \\ \text{subject to} & x \in S, \end{cases}$$

where the partial ordering in the image space of the objective function is understood to be induced by the standard ordering cone $\mathbb{R}^m_+$. More precisely, denoting $I := \{1, \dots, m\}$, we have for any $x = (x_1, \dots, x_m)$, $y = (y_1, \dots, y_m) \in \mathbb{R}^m$:

$$x \leq y \quad \Leftrightarrow \quad x_i \leqq y_i, \text{ for all } \quad i \in I, \text{ and } \quad \sum_{i \in I} x_i < \sum_{i \in I} y_i.$$

* "Babeş-Bolyai" University, Faculty of Mathematics and Computer Science, 400084, Cluj-Napoca, Romania, email: {llupsa,ioana}@math.ubbcluj.ro.

†This paper is supported by the scientific grant CEEX 125/31.07.2006.

Recall (see e.g. [6]) that the set of *efficient solutions*, called by us the efficient set, and the set of *weakly-efficient solutions*, called by us the weakly-efficient set, of problem (1) are given, respectively, by:

$$\mathrm{Eff}(S;f) := \{x \in S \mid \nexists\, y \in S \text{ such that } f(y) \leq f(x)\},$$

$$\mathrm{WEff}(S;f) := \{x \in S \mid \nexists\, y \in S \text{ such that } f(y) < f(x)\}.$$

In what follows we give a general method to approximate the sets $\mathrm{Eff}(S;f)$ and $\mathrm{WEff}(S;f)$ in the hypothesis that the function f is lower unimodal on S.

We mention that the generalization of "unimodal function" notion was made, until now, in the following three directions:

a) By replacing the real interval which was the definition domain of the unimodal property (see [5]) with an arbitrary set of real numbers (see [7]);

b) By working with multivariate functions, due to the fact that the domain of definition for the unimodal property is a compact interval in $\mathbb{R}^n$ (see [1] and [4]);

c) By replacing the real univariate functions with vectorial univariate function (see [8]). We mention, also, that the obtained results in the third case contain, as particular cases, properties already known from the first case. Analogously, the given algorithms in the case c) may be successfully used in the case a), too.

2. UNIMODAL VECTORIAL FUNCTIONS ON A SET AND SOME OF THEIR PROPERTIES

In the following we suppose that D is a non empty subset of the real numbers set $\mathbb{R}$ and m is a natural number, $m \geqq 2$.

Definition 1. (see [8]) *A function $\varphi : D \to \mathbb{R}$ is said to be lower unimodal on $S \subset D$ if there exist $u, v \in S$ satisfying the following conditions:*

(LU1) $\varphi(u) = \varphi(v)$;

(LU2) $\varphi(x) > \varphi(y)$ *whenever* $x, y \in S,\ x < y \leqq u$;

(LU3) $\varphi(x) < \varphi(y)$ *whenever* $x, y \in S,\ v \leqq x < y$;

(LU4) $S \cap [u, v] = \{u, v\}$.

Remark 2. (see [8])

1) By (LU1)–(LU2) it follows that $u \leqq v$, so (LU4) makes sense.
2) As a direct consequence of (LU1)–(LU4) we can easily deduce that for any $x, y \in S$ the following implications hold:

If $x < y \leqq v$ then $\varphi(x) \geqq \varphi(y)$;

If $u \leqq x < y$ then $\varphi(x) \leqq \varphi(y)$. □

Definition 3. (see [8]) *A function $f = (f_1, ..., f_m) : D \to \mathbb{R}^m$ is said to be lower unimodal on $S \subset D$ if all its scalar components f_i, $i \in I = \{1, ..., m\}$, are lower unimodal on S.*

If $f = (f_1, ..., f_m) : D \to \mathbb{R}^m$ is a lower unimodal function on $S \subseteq D$, by u_i, v_i we denote, for every $i \in I$, the points u and v from Definition 1.

In the following we remember some properties of the sets $\mathrm{Eff}(S; f)$ and $\mathrm{WEff}(S; f)$ in the circumstances that the function f is lower unimodal on S. The problem was studied in [9], where the authors showed that both the sets $\mathrm{Eff}(S; f)$ and $\mathrm{WEff}(S; f)$ can be completely determined by only using the numbers $u_1, v_1, \ldots, u_m, v_m$. As in mentioned paper, we denote:

$$\underline{u} := \min_{i\in I} u_i, \quad \underline{v} := \min_{i\in I} v_i, \quad \overline{u} := \max_{i\in I} u_i \quad \text{and} \quad \overline{v} := \max_{i\in I} v_i.$$

Theorem 4. (see [9], Theorems 2.1, 2.2)

i) *The set of weakly efficient solutions of problem* (1) *admits the following representation:*

$$\mathrm{WEff}(S; f) \quad = \quad [\underline{u}, \overline{v}] \cap S. \tag{2}$$

ii) *The set of efficient solutions of problem* (1) *is given by the following representation:*

$$\mathrm{Eff}(S; f) \quad = \quad [\min\{\underline{v}, \overline{u}\}, \max\{\underline{v}, \overline{u}\}] \cap S. \tag{3}$$

Other important properties of lower unimodal functions will be given in that follows. Let us suppose that S is a nonempty subset of D and $f : D \to \mathbb{R}^m$, is a lower unimodal function on S. If c and d are elements of S, we introduce the notations: $I^-_{c,d} = \{i \in I \mid f_i(c) < f_i(d)\}$, $I^0_{c,d} = \{i \in I \mid f_i(c) = f_i(d)\}$, $I^+_{c,d} = \{iI \mid f_i(c) > f_i(d)\}$.

Theorem 5. *If $a, b, c, d \in S$ are such that $a \leqq c < d \leqq b$, $a \leqq \underline{u}$,* and $\overline{v} \leqq b$, *and*

$$\theta = \inf\{|x - y| \mid x, y \in [a,\, b] \cap S,\ x \neq y\}, \tag{4}$$

then the following sentences are true:

(i) *If $I^-_{c,d} \neq \emptyset$, then $\{\underline{u}, \underline{v}\} \subset [a,\, d - \theta] \cap S$;*
(ii) *If $I^-_{c,d} = \emptyset$ and $I^0_{c,d} \neq \emptyset$, then $\{\underline{u}, \underline{v}\} \subset [c,\, d] \cap S$;*
(iii) *If $I^-_{c,d} = \emptyset$ and $I^0_{c,d} = \emptyset$, then $\{\underline{u}, \underline{v}\} \subset [c + \theta,\, b] \cap S$;*
(iv) *If $I^+_{c,d} \neq \emptyset$, then $\{\overline{u}, \overline{v}\} \subset [c + \theta,\, b] \cap S$;*
(v) *If $I^+_{c,d} = \emptyset$ and $I^0_{c,d} \neq \emptyset$, then $\{\overline{u}, \overline{v}\} \subset [c,\, d] \cap S$;*
(vi) *If $I^+_{c,d} = \emptyset$ and $I^0_{c,d} = \emptyset$, then $\{\overline{u}, \overline{v}\} \subset [a,\, d - \theta] \cap S$.*

Proof. (i) As $I^-_{c,d} \neq \emptyset$, there is an $i \in I$ such that $f_i(c) < f_i(d)$ and, in this case, Remark 2 implies $u_i,\, v_i \in [a,\, d[\cap S$. Then $v_i < d$ and (4) implies $\theta \leqq d - v_i$ or $v_i \leqq d - \theta$. As $u_i \leqq v_i$, we get $u_i \leqq d - \theta$. Therefore,

$$\underline{u} \leqq u_i \leqq d - \theta \quad \text{and} \quad \underline{v} \leqq v_i \leqq d - \theta.$$

These imply $\underline{u} \in [a, d-\theta] \cap S$ and $\underline{v} \in [a, d-\theta] \cap S$, i.e. (i) holds.

(ii) As $I^{-}_{c,d} = \emptyset$, we have $f_i(c) \geqq f_i(d)$, for all $i \in I$. Then Remark 2 implies $c \leqq u_i \leqq v_i$, for all $i \in I$. Therefore, $c \leqq \underline{u}$ and $c \leqq \underline{v}$. On the other hand, as $I^{0}_{c,d} \neq \emptyset$, there is $k \in I$ such that $f_k(c) = f_k(d)$. In this case, Remark 2 gives $c \leqq u_k \leqq d$ and $c \leqq v_k \leqq d$. It follows that $\underline{u} \leqq u_k \leqq d$ and $\underline{v} \leqq v_k \leqq d$. Therefore (ii) holds.

(iii) If $I^{-}_{c,d} \cup I^{0}_{c,d} = \emptyset$, then $f_i(c) > f_i(d)$, for all $i \in I$. In this case, Remark 2 implies $c < u_i \leqq v_i \leqq b$, for all $i \in I$. Then $\underline{u} \leqq \underline{v} \leqq b$. Also, in view of (4), we have

$$\theta \leqq u_i - c \quad \text{and} \quad \theta \leqq v_i - c, \quad \text{for all} \quad i \in I.$$

These imply $c + \theta \leqq u_i$ and $c + \theta \leqq v_i$, for all $i \in I$. Therefore, we have $c + \theta \leqq \underline{u}$ and $c + \theta \leqq \underline{v}$. Hence (iii) holds.

In the same way we can prove that (iv)–(vi) are true. □

We will use the above results in the next section to elaborate a general method for approximating the efficient set and the weakly-efficient set in a unimodal vectorial optimization problems (i.e. in the problem (MOP) when the function f is lower unimodal on S).

3. THE (UMA) ALGORITHM

In what follows we suppose that:

H1. $D \subseteq \mathbb{R}$;

H2. S is a nonempty, compact subset of D, and $\text{card} S \geqq 2$;

H3. $f = (f_1, ..., f_m) : D \to \mathbb{R}^m$ is a lower unimodal function on S, where m is a natural and not a null number.

The following algorithm permits to obtain two sets WEF and EF. These sets approximate the sets $\text{Eff}(S; f)$ and $\text{WEff}(S; f)$, with a given error $\varepsilon > 0$.

We mention that an algorithm to approximate $\text{Eff}(S; f)$ and $\text{WEff}(S; f)$, in the case when the set S is a real compact interval, was given in [10] and other algorithm, in the some condition, but more performance, [3]. In the special case when S is a discrete set, the first algorithm to determine the sets $\text{Eff}(S; f)$ and $\text{WEff}(S; f)$ was given in [9] and other algorithm, in the some condition, but more performance, in [8].

In the UMA Algorithm, we use the following notations:

$$\underline{I}^{-}_{k} \ \text{ for } \ I^{-}_{\underline{c}_k, \underline{d}_k} \quad \text{and} \quad \underline{I}^{0}_{k} \ \text{ for } \ I^{0}_{\underline{c}_k, \underline{d}_k};$$

$$\overline{I}^{+}_{k} \ \text{ for } \ I^{+}_{\overline{c}_k, \overline{d}_k} \quad \text{and} \quad \overline{I}^{0}_{k} \ \text{ for } \ I^{0}_{\overline{c}_k, \overline{d}_k}.$$

The serial (UMA) Algorithm

Step 1. Set
$k := 0,\ h := 0,\ \underline{sw} := 1,\ \overline{sw} := 1,\ \underline{S}_1 := S,\ \overline{S}_1 := S,$
$\underline{a}_1 := \min S,\ \overline{a}_1 := \min S,\ \underline{b}_1 := \max S,\ \overline{b}_1 := \max S,$
and proceed.

Step 2. If $\underline{sw} = 0$, then go to step 12; else proceed.

Step 3. Increase k by 1 and proceed.

Step 4. Set $\mu_k := \inf\{|x-y| \quad | \quad x, y \in \underline{S}_k,\ x \neq y\}$, and proceed.

Step 5. Take $\underline{c}_k \in \underline{S}_k$ and $\underline{d}_k \in \underline{S}_k$, such that

$$\begin{array}{ll} \underline{a}_k < \underline{c}_k < \underline{d}_k < \underline{b}_k, & \text{if card } \underline{S}_k > 3, \\ \underline{a}_k = \underline{c}_k < \underline{d}_k < \underline{b}_k, & \text{if card } \underline{S}_k = 3, \\ \underline{a}_k = \underline{c}_k < \underline{d}_k = \underline{b}_k, & \text{if card } \underline{S}_k = 2, \end{array} \tag{5}$$

and proceed.

Step 6. Build the sets $\underline{I}_k^- = \{i \in I \mid f_i(\underline{c}_k) < f_i(\underline{d}_k)\}$ and
$\underline{I}_k^0 = \{i \in I \mid f_i(\underline{c}_k) = f_i(\underline{d}_k)\}$ and proceed.

Step 7. If $\underline{I}_k^- \neq \emptyset$ then set $\underline{S}_{k+1} := \underline{S}_k \cap [\underline{a}_k,\ \underline{d}_k - \mu_k]$,
$\underline{a}_{k+1} := \min \underline{S}_{k+1},\ \underline{b}_{k+1} := \max \underline{S}_{k+1},\ \underline{u}_k := \underline{c}_k,\ \underline{v}_k := \underline{c}_k,$
and go to step 10; else go to the next step.

Step 8. If $\underline{I}_k^0 \neq \emptyset$ then set $\underline{S}_{k+1} := \underline{S}_k \cap [\underline{c}_k,\ \underline{d}_k]$, $\underline{a}_{k+1} := \min \underline{S}_{k+1}$,
$\underline{b}_{k+1} := \max \underline{S}_{k+1}$, $\underline{u}_k := \underline{c}_k$, $\underline{v}_k := \underline{d}_k$, and go to step 10, else go to the next step.

Step 9. Set $\underline{S}_{k+1} := \underline{S}_k \cap [\underline{c}_k + \mu_k,\ \underline{b}_k]$, $\underline{a}_{k+1} := \min \underline{S}_{k+1}$, $\underline{b}_{k+1} := \max \underline{S}_{k+1}$,
$\underline{u}_k := \underline{d}_k,\ \underline{v}_k := \underline{d}_k$, and proceed.

Step 10. If $\underline{b}_k - \underline{a}_k < \varepsilon/2$, or card $\underline{S}_k = 2$, then proceed; else go to step 12.

Step 11. Set $\underline{sw} := 0$, and proceed.

Step 12. If $\overline{sw} = 0$, then go to step 22; else proceed.

Step 13. Increase h by 1 and proceed.

Step 14. Set $\nu_h := \inf\{|x-y| \quad | \quad x, y \in \overline{S}_h,\ x \neq y\}$, and proceed.

Step 15. Take $\overline{c}_h \in S$ and $\overline{d}_h \in S$, such that

$$\begin{array}{ll} \overline{a}_h < \overline{c}_h < \overline{d}_h < \overline{b}_h, & \text{if card } \overline{S}_h > 3, \\ \overline{a}_h < \overline{c}_h < \overline{d}_h = \overline{b}_h, & \text{if card } \overline{S}_h = 3, \\ \overline{a}_h = \overline{c}_h < \overline{d}_h = \overline{b}_h, & \text{if card } \overline{S}_h = 2, \end{array} \tag{6}$$

and proceed.

Step 16. Build the sets $\overline{I}_h^+ = \{i \in I \mid f_i(\overline{c}_h) > f_i(\overline{d}_h)\}$, and
$\overline{I}_h^0 = \{i \in I \mid f_i(\overline{c}_h) = f_i(\overline{d}_h)\}$, and proceed.

Step 17. If $\overline{I}_h^+ \neq \emptyset$ then set $\overline{S}_{h+1} := \overline{S}_h \cap [\overline{c}_h + \nu_h,\ \overline{b}_h]$,
$\overline{a}_{h+1} := \min \overline{S}_{h+1},\ \overline{b}_{h+1} := \max \overline{S}_{h+1},\ \overline{u}_h := \overline{d}_h,\ \overline{v}_h := \overline{d}_h,$
and go to step 20, else proceed.

Step 18. If $\overline{I}_h^0 \neq \emptyset$ then set $\overline{S}_{h+1} := \overline{S}_h \cap [\overline{c}_h,\ \overline{d}_h]$,
$\overline{a}_{h+1} := \min \overline{S}_{h+1},\ \overline{b}_{h+1} := \max \overline{S}_{h+1},\ \overline{u}_h := \overline{c}_h,\ \overline{v}_h := \overline{d}_h,$
and go to step 20, else proceed.

Step 19. Set $\overline{S}_{h+1} := \overline{S}_h \cap [\overline{a}_h, \overline{d}_h - \nu_h]$,
$\overline{a}_{h+1} := \min \overline{S}_h$, $\overline{b}_{h+1} := \max \overline{S}_{h+1}$, $\overline{u}_h := \overline{c}_h$, $\overline{v}_h := \overline{c}_h$, and proceed.
Step 20. If $\overline{b}_h - \overline{a}_h < \varepsilon/2$, or card $\overline{S}_h \leqq 2$, then proceed; else go back to step 2.
Step 21. Set $\overline{sw} := 0$.
Step 22. If $\underline{sw} \neq 0$, then go back to step 2.
Step 23. Set $\text{WEF} := \{x \in S \mid \underline{u}_k \leqq x \leqq \overline{v}_h\}$ and proceed.
Step 24. If $\underline{v}_k \leqq \overline{u}_h$ then set $\text{EF} := \{x \in S \mid \underline{v}_k \leqq x \leqq \overline{u}_h\}$;
else set $\text{EF} := \{x \in S \mid \overline{u}_h \leqq x \leqq \underline{v}_k\}$.
Step 25. Stop.

THEOREM 6. *In the hypotheses* H1–H3, *if k is a natural number and the numbers $\underline{a}_1, \ldots, \underline{a}_k$, $\underline{b}_1, \ldots, \underline{b}_k$, $\underline{u}_1, \ldots, \underline{u}_k$, $\underline{v}_1, \ldots, \underline{v}_k$ are the points given by the UMA Algorithm, one have*

$$\underline{u},\ \underline{v} \in \underline{S}_j, \text{ for all } j \in \{1, \ldots, k\}. \tag{7}$$

If, in addition, card $\underline{S}_k \leqq 2$, *then*

$$\underline{u}_k = \underline{u},\ \underline{v}_k = \underline{v}. \tag{8}$$

Proof. First we prove that (7) holds. The proof is by induction. Step 1 gives $\underline{S}_1 = S$. As $\underline{u}, \underline{v} \in S$, obviously $\underline{u}, \underline{v} \in \underline{S}_1$. Therefore, if $k = 1$, then (7) is true. Let now consider $k > 1$, and let be $j \in \{1, \ldots, k-1\}$. We prove that if $\underline{u}, \underline{v} \in \underline{S}_j$, then

$$\underline{u},\ \underline{v} \in \underline{S}_{j+1}. \tag{9}$$

From the algorithm it follows that $\underline{b}_i - \underline{a}_i \geqq \varepsilon/2$, and card $\underline{S}_i > 2$, for all $i \in \{1, \ldots, k-1\}$. Therefore card $\underline{S}_j \geqq 3$. If $\underline{c}_j$ and $\underline{d}_j$ are the points chosen at the j^{th} iteration, three cases are possible:

$$1)\ \underline{I}_j^- \neq \emptyset; \quad 2)\ \underline{I}_j^- = \emptyset, \text{ and } \underline{I}_j^0 \neq \emptyset; \quad 3)\ \underline{I}_j^- = \emptyset, \text{ and } \underline{I}_j^0 = \emptyset.$$

If $\underline{I}_j^- \neq \emptyset$, from Step 7 we have $\underline{S}_{j+1} := \underline{S}_j \cap [\underline{a}_j, \underline{d}_j - \mu_j]$. On the other hand, Theorem 5 gives $\underline{u}, \underline{v} \in [\underline{a}_j, \underline{d}_j - \mu_j] \cap S$. But, in view of the induction hypothesis, we have $\underline{u}, \underline{v} \in \underline{S}_j$. Therefore (9) holds. By analogy, it can be proved that (9) is true, in the other two cases. Therefore, because (7) is true for $j = 1$, by induction, we can conclude that (7) holds for all $j \in \{1, \ldots, k\}$.

Now we prove that, if, in addition, card $\underline{S}_k = 2$, then $\underline{u}_k = \underline{u}$ and $\underline{v}_k = \underline{v}$.

First we mention that, from (7), we get $\underline{u}, \underline{v} \in \underline{S}_k$. As card $\underline{S}_k = 2$, Step 5 gives $\underline{c}_k = \underline{a}_k$ and $\underline{d}_k = \underline{b}_k$. Therefore $\underline{S}_k = \{\underline{a}_k, \underline{b}_k\}$. Three cases may appear: $\underline{I}_k^- \neq \emptyset$; $\underline{I}_k^- = \emptyset$ and, $\underline{I}_k^0 \neq \emptyset$; $\underline{I}_k^- = \emptyset$, and $\underline{I}_k^0 = \emptyset$.

If $\underline{I}_k^- \neq \emptyset$, in view of Theorem 5 we have $\{\underline{u}, \underline{v}\} \subseteq [\underline{a}_k, \underline{d}_k[\cap \underline{S}_k = \{\underline{a}_k\}$. Then, we get $\underline{u} = \underline{v} = \underline{a}_k$. On the other hand, Step 7 gives $\underline{u}_k = \underline{v}_k = \underline{c}_k$. As $\underline{c}_k = \underline{a}_k$, equality (8) holds.

In the second case, as $\underline{I}_k^0 \neq \emptyset$, there is $i \in I$ such that $u_i = c_k = a_k$ and $v_i = d_k = b_k$. Therefore $\underline{u} = a_k$. For all $i \in I \setminus \underline{I}_k^0$ we have $u_i = v_i = d_k = b_k$. Hence, $v_i = d_k = b_k$ for all $i \in I$. Therefore $\underline{v} = b_k$. On the other hand, Step 8 gives $\underline{u}_k = \underline{c}_k = \underline{a}_k$ and $\underline{v}_k = \underline{c}_k = \underline{b}_k$. Again, (8) holds.

If $I^{-}_{\underline{c}_k,\underline{d}_k} = \emptyset$ and $I^{0}_{\underline{c}_k,\underline{d}_k} = \emptyset$, Theorem 5 gives $\{\underline{u}, \underline{v}\} \subseteq]\underline{c}_k, \underline{b}_k] \cap S = \{\underline{b}_k\}$. On the other hand, Step 9 gives $\underline{u}_k = \underline{v}_k = \underline{d}_k = \underline{b}_k$. Hence (8) holds, too. □

In the same manner we can prove:

THEOREM 7. *In the hypotheses* H1–H3*, if h is a natural number and the numbers* $\overline{a}_1, ..., \overline{a}_h$, $\overline{b}_1, ..., \overline{b}_h$, $\overline{u}_1, ..., \overline{u}_h$, $\overline{v}_1, ..., \overline{v}_h$ *are the points given by the UMA Algorithm, then* $\overline{u}, \overline{v} \in [\overline{a}_j, \overline{b}_j]$, *for all* $j \in \{1, ..., h\}$. *If, in addition,* card $\overline{S}_h \leqq 2$, *then* $\overline{u}_h = \overline{u}$, $\overline{v}_h = \overline{v}$.

The following results come easily:

REMARK 8. If $\mu(S) > 0$ and $\varepsilon \leqq \mu(S)$, then the UMA Algorithm stops after a finite number of iterations, $\text{Eff}(S; f) = \text{EF}$ and $\text{WEff}(S; f) = WEF$. □

REMARK 9. If card $\underline{S}_k > 2$, for all natural number k, and there is a convergent sequence $(\underline{\delta}_k)_{k\in\mathbb{N}^*}$ such that

$$\lim_{k\to+\infty} \underline{\delta}_k = 0 \quad \text{and} \quad \underline{b}_k - \underline{a}_k \leqq \underline{\delta}_k, \ \text{ for all } k \in \mathbb{N}^*,$$

then the sequences $(\underline{u}_k)_{k\in\mathbb{N}^*}$ and $(\underline{v}_k)_{k\in\mathbb{N}^*}$ are convergent and $\lim\limits_{k\to+\infty} \underline{u}_k = \underline{u}$, $\lim\limits_{k\to+\infty} \underline{v}_k = \underline{v}$.

Similarly results can be given for the sequences $(\overline{u}_h)_{h\in\mathbb{N}^*}$, and $(\overline{v}_h)_{h\in\mathbb{N}^*}$. □

If A and B are two real intervals, we set:

$\text{lng}(A \setminus B) = 0$, if $A \setminus B = \emptyset$;

$\text{lng}\,(A \setminus B) = w_2 - w_1$, if $A \setminus B = [w_1, w_2]$;

$\text{lng}(A \setminus B) = w_2 - w_1 + w_4 - w_3$, if $A \setminus B = [w_1, w_2] \cup [w_3, w_4]$.

REMARK 10. In the hypotheses H1–H3, if $\varepsilon > 0$ and the sets EF and WEF are built with the UMA algorithm, then $\text{lng}(\text{WEF} \setminus \text{WEff}(S; f)) \leqq \varepsilon$, $\text{lng}(\text{WEff}(S; f) \setminus \text{WEF}) \leqq \varepsilon$, $\text{lng}\,(\text{EF} \setminus \text{Eff}(S; f)) \leqq \varepsilon$, and $\text{lng}\,(\text{Eff}(S; f) \setminus \text{EF}) \leqq \varepsilon$. □

4. PARTICULAR CASES

In what follows, we show how, from the UMA algorithm, one can obtain the methods given in [3], [8], [9] and [10].

4.1. S is a compact interval. In view of [9], Remark 1.1, when S is a compact interval and the function $f : [a, b] \to \mathbb{R}^m$ is lower unimodal on S, then $\underline{u}_i = \underline{v}_i = x_i$, for all $i \in \{1, ..., m\}$, where $\{x_i\} = \text{Arg}\min\limits_{x\in S} f_i(x)$. Therefore $\underline{u} = \underline{v}$ and $\overline{u} = \overline{v}$. Also, it is known that, if $S = [a, b]$, then, for each natural numbers $k \geqq 1$, we have card $\underline{S}_k > 3$. Therefore, one can choose the points $\underline{c}_k, \underline{d}_k \in \underline{S}_k$ satisfying the conditions: $\underline{a}_k < \underline{c}_k < \underline{d}_k < \underline{b}_k$ and

$\underline{c}_k - \underline{a}_k = \underline{b}_k - \underline{d}_k$. For this, one takes as $\underline{t}_k$ any real numbers satisfying the conditions $0 < \underline{t}_k < 1/2$ and put

$$\underline{c}_k = \underline{a}_k + \underline{t}_k(\underline{b}_k - \underline{a}_k), \quad \underline{d}_k = \underline{b}_k - \underline{t}_k(\underline{b}_k - \underline{a}_k). \tag{10}$$

Then

$$\underline{b}_{k+1} - \underline{a}_{k+1} = (1 - \underline{t}_k)(\underline{b}_k - \underline{a}_k). \tag{11}$$

In view of the UMA algorithm, we have $\underline{u} = \underline{v} \in [\underline{a}_k, \underline{b}_k]$, and $\underline{u}_k, \underline{v}_k \in [\underline{a}_k, \underline{b}_k]$. Therefore, if $(\underline{t}_k)_{k=1}^m$ is a finite sequence of real numbers with $0 < \underline{t}_k < 1/2$, for each $k \in \{1, ..., m\}$, and if the sequence of sets $(\underline{S}_k)_{k=1}^m$ is constructed by the UMA algorithm, where we take $\underline{c}_k$ and $\underline{d}_k$ using (10), then

$$\begin{aligned} |\underline{u} - \underline{u}_k| &\leqq (1 - \underline{t}_{k-1})(\underline{b}_{k-1} - \underline{a}_{k-1}) = \\ &= (1 - \underline{t}_{k-1})(1 - \underline{t}_{k-2})(\underline{b}_{k-2} - \underline{a}_{k-2}) \\ &= ... = \prod_{j=1}^{k-1}(1 - \underline{t}_j) \cdot (b - a) \end{aligned}$$

and

$$\begin{aligned} |\underline{v} - \underline{v}_k| &\leqq (1 - \underline{t}_{k-1})(\underline{b}_{k-1} - \underline{a}_{k-1}) = \\ &= (1 - \underline{t}_{k-1})(1 - \underline{t}_{k-2})(\underline{b}_{k-2} - \underline{a}_{k-2}) \\ &= ... = \prod_{j=1}^{k-1}(1 - \underline{t}_j) \cdot (b - a). \end{aligned}$$

Analogously, if $(\overline{t}_h)_{h=1}^m$ is a finite sequence of real numbers with $0 < \overline{t}_h < \frac{1}{2}$, for each $h \in \{1, ..., m\}$, and if the sequence of sets $(\overline{S}_h)_{h=1}^m$ is constructed by the UMA algorithm, where

$$\overline{c}_h = \overline{a}_h + \overline{t}_h(\overline{b}_h - \overline{a}_h), \quad \overline{d}_h = \overline{b}_h - \overline{t}_h(\overline{b}_h - \overline{a}_h), \tag{12}$$

then we have

$$|\overline{u} - \overline{u}_h| \leqq \prod_{j=1}^{h-1}(1 - \overline{t}_j) \cdot (b - a) \quad \text{and} \quad |\overline{v} - \overline{v}_h| \leqq \prod_{j=1}^{h-1}(1 - \overline{t}_j) \cdot (b - a).$$

Hence, if we put $\mathrm{WEF} = [\underline{u}_k, \overline{v}_h] \cap S$, then we have $\mathrm{lng}(\mathrm{WEF} \setminus \mathrm{WEff(S; f)}) \leqq (\Pi_{j=1}^{k-1}(1 - \underline{t}_j) + \Pi_{j=1}^{h-1}(1 - \overline{t}_j) \cdot (\max S - \min S)$ and $\mathrm{lng}(\mathrm{WEff(S; f)} \setminus \mathrm{WEF}) \leqq (\Pi_{j=1}^{k-1}(1 - \underline{t}_j) + \Pi_{j=1}^{h-1}(1 - \overline{t}_j) \cdot (\max S - \min S)$. A similar result can be obtained for the sets EF and $\mathrm{Eff}(S; f)$. In order to decrease the number of computations for the values of f, one may choose $\underline{t}_k$ such that $\underline{d}_{k+1} = \underline{c}_k$, if $(\underline{I}_k^- \bigcup \underline{I}_k^0) \neq \emptyset$, and $\underline{c}_{k+1} = \underline{d}_k$, if $(\underline{I}_k^- \cup \underline{I}_k^0) = \emptyset$. Therefore, we have either

$$\underline{b}_{k+1} - \underline{t}_{k+1}(\underline{b}_{k+1} - \underline{a}_{k+1}) = \underline{c}_k, \text{ or } \underline{a}_{k+1} + \underline{t}_{k+1}(\underline{b}_{k+1} - \underline{a}_{k+1}) = \underline{d}_k.$$

Then the numbers $\underline{t}_k$, $k \in \mathbb{N}$, have to satisfy the condition

$$(1 - \underline{t}_{k+1})(1 - \underline{t}_k) = \underline{t}_k. \tag{13}$$

Analogously, we may choose the sequence $(\overline{t}_h)_{h \in \mathbb{N}^*}$ such that

$$(1 - \overline{t}_{h+1})(1 - \overline{t}_h) = \overline{t}_h. \tag{14}$$

By choosing particular values for the sequence $(\underline{t}_k)_{k\in\mathbb{N}^*}$, such that (13) is satisfied, and particular values for the sequence $(\overline{t}_h)_{h\in\mathbb{N}^*}$ such that (14) is satisfied, we obtain a particular type of methods, which, by analogy with the real case, we call the methods of successive section. Two important sub-cases are given further on.

Case I. If $\underline{t}_k = t$, for each $k \in \mathbb{N}^*$, and $\overline{t}_h = t$, for each $h \in \mathbb{N}^*$, then (13) and (14) imply $(1-t)^2 = t$, i.e. $t^2 - 3t + 1 = 0$. The above equation has two solutions. If we choose

$$\underline{t}_k = \tfrac{3-\sqrt{5}}{2}, \ \ \overline{t}^h = \tfrac{3-\sqrt{5}}{2}, \ \text{ for each } \ \ h \in \mathbb{N}^*,$$

then we call the resulting method, the method of the "gold section".

Case II. It is known that the Fibonacci numbers F_k, $k \in \mathbb{N}^*$, satisfy the following recurrence formula

$$F_{k+1} = F_k + F_{k-1}, \text{ for each } k \in \mathbb{N}^*,\ k \geqq 3,\ F_1 = F_2 = 1.$$

Let $m \in \mathbb{N}^*$. It is easy to see that, if we choose

$$\underline{t}_k = \overline{t}_k = \tfrac{F_{m-k+1}}{F_{m-k+3}}, \text{ for each } k \in \{1, ..., m\}, \tag{15}$$

then

$$(1-\underline{t}_{k+1})(1-\underline{t}_k) = (1-\overline{t}_{h+1})(1-\overline{t}_h) = \tfrac{F_{m-k+1}}{F_{m-k+3}} = \underline{t}_k,\ \forall\, k \in \{1, ..., m-1\}.$$

As

$$0 < \tfrac{F_{m-k+1}}{F_{m-k+3}} = \tfrac{F_{m-k+1}}{F_{m-k+2}+F_{m-k+1}} < \tfrac{F_{m-k+1}}{2F_{m-k+1}} = \tfrac{1}{2},$$

the numbers $\underline{t}_k$, $\overline{t}_k$, $k \in \{1, ..., m\}$, given by (15), satisfy condition (13) and (14). The method of successive section obtained using (15) is known as the "Fibonacci's method" (see [3]).

4.2. S **is a finite set.** Let be $S = \{x_1, ..., x_n\}$, where $n \in \mathbb{N}$, $n \geqq 2$, and $x_1 < x_2 < ... < x_n$. In this case, in the first step of the UMA algorithm, we have $a = x_1$ and $b = x_n$, and, therefore, $\underline{a}_1 = \overline{a}_1 = x_1$ and $\underline{b}_1 = \overline{b}_1 = x_n$. Then $\underline{S}_1 = \overline{S}_1 = \{x_1, ..., x_n\}$. Therefore card $\underline{S}_1$ = card $\overline{S}_1 = n$, and, in each iterations, card $\underline{S}_k \in \mathbb{N}^*$ and card $\overline{S}_h \in \mathbb{N}^*$. We suppose that, at each iteration, k, we rewrite the elements of the set S_k, such that

$$\underline{S}_k = \{x_1^k, ..., x_{n_k}^k\}, \quad \text{and} \quad x_1^k < x_2^k < ... < x_{n_k}^k,$$

where $n_k =$ card $\underline{S}_k$. Two cases may appear: $n_k = 2$, or $n_k > 3$.

If at the k iteration we have $n_k > 3$, then we may choose $\underline{c}_k = x_m^k$ and $\underline{d}_k = x_{m+1}^k$, where $m = [n_k/2]$, in the step 5 of the UMA algorithm.

If at the k iteration we have $n_k = 2$, then we may choose $d_k = x_2^k$ in the step 5 of the UMA algorithm.

Analogously, in each iteration, we may choose the points $\overline{c}_h$ and $\overline{d}_h$. These specifications being done, the UMA algorithm is the same as the algorithm given in [8].

Table 1. Results of the UMA Algorithm

it	$\underline{a}_k$	$\underline{b}_k$	$\underline{c}_k$	$\underline{d}_k$	$\underline{u}_k$	$\underline{v}_k$	$\underline{sw}$	$\overline{a}_k$	$\overline{b}_k$	$\overline{c}_k$	$\overline{d}_k$	$\overline{u}_k$	$\overline{v}_k$	$\overline{sw}$
1	0	1	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	1	0	1	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1
2	0	$\frac{1}{2}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{8}$	1	$\frac{1}{4}$	1	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	1
3	0	$\frac{1}{4}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	1	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0
4	$\frac{1}{16}$	$\frac{1}{4}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	1							0
5	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{8}$	0							0

4.3. ***S*** **is an infinite set, but not a real interval.** The UMA algorithm can also be used in the case when the set S is infinite but not a real interval, situation which could not be accomplished by the other cited methods.

EXAMPLE 11. Let be $S = \{\frac{1}{2^n} \,|\, n \in \mathbb{N}\} \cup \{0\}$ and $f = (f_1, f_2) : [0, 1] \to \mathbb{R}^2$ the function given by

$$f_1(x) = |x - 1/2|, \quad f_2(x) = |x - 1/6|, \text{ for all } x \in [0, 1].$$

□

Obviously, f is lower unimodal on S. If we apply the UMA algorithm, taking $\varepsilon = 1/100$, we have to make 5 iterations, in order to compute the sets WEF and EF, as can be seen in Table 1.

Therefore

$$WEF = \{\tfrac{1}{8}, \tfrac{1}{4}, \tfrac{1}{2}\}, \quad EF = \{\tfrac{1}{8}, \tfrac{1}{4}, \tfrac{1}{2}\}.$$

5. THE PARALLEL UMA ALGORITHM

The presented UMA Algorithm is thought for a serial implementation. Due to the fact that its second part (Steps 11–20) contains the same type of instructions as the first part (Steps 2–10), the time of execution can be reduced by using parallel calculus.

There are several way of using more that one processor, but we shall consider a parallel execution of Master-Slave type (see [2]). In the created network, we have a Master processor, with the identification number (ID) equal with 1, and three Slaves processors, with IDs equal to 2, 3 and 4. All the processors memorize the whole program, but each of them will perform exactly the instructions needed, according with its ID number.

A possible parallel algorithm is the following:

If $ID = 1$ then let {Master execution}

$k = 0;\ h = 0;$

$\underline{a}_1 = \min S;\ \overline{a}_1 = \min S;\ \underline{b}_1 = \max S;\ \overline{b}_1 = \max S;$

    $bool = 0$; {for the points of minimum}

Repeat

$k = k + 1;$

$\mu_k = \inf\{|x - y| \mid x, y \in [\underline{a}_k, \underline{b}_k] \cap S\};$

Send Message to Slaves $(\underline{a}_k, \underline{b}_k, bool)$

Receive Message from Slaves (flag);
If $flag = 0$ then $\underline{a}_{k+1} = \underline{a}_k$;
$\underline{b}_{k+1} = \max\{[\underline{a}_k, \underline{d}_k - \mu_k] \cap S\}$
If $falg = 1$ then $\underline{a}_{k+1} = \underline{c}_k$; $\underline{b}_{k+1} = \underline{d}_k$;
$\underline{u}_k = \underline{c}_k$; $\underline{v}_k = \underline{d}_k$;
If $flag = 2$ then $\underline{a}_{k+1} = \min\{[\underline{c}_k + \mu_k, \underline{b}_k] \cap S\}$;
$\underline{b}_{k+1} = \underline{b}_k$; $\underline{u}_k = \underline{d}_k$; $\underline{v}_k = \underline{d}_k$
Until $\underline{b}_k - \underline{a}_k < \dfrac{\varepsilon}{2}$; or $\mathrm{card}[\underline{a}_k, \underline{b}_k] \cap S = 2$;
$bool = 1$; {for the points of maximum}
Repeat
$h = h + 1$;
$\nu_h = \inf\{|x - y| \mid x, y \in [\overline{a}_h, \overline{b}_h] \cap S\}$;
Send Message to Slaves $(\overline{a}_h, \overline{b}_h, bool)$;
Receive Message from Slaves (flag);
If $flag = 0$ then $\overline{a}_{h+1} = \min\{[\overline{c}_j + \nu_h, \overline{b}_h] \cap S\}$;
$\overline{b}_{h+1} = \overline{b}_h$; $\overline{u}_h = \overline{d}_h$; $\overline{v}_h = \overline{d}_h$;
If $flag = 1$ then $\overline{a}_{h+1} = \overline{c}_h$; $\overline{b}_{h+1} = \overline{d}_h$; $\overline{u}_h = \overline{c}_h$;
$\overline{v}_h = \overline{d}_h$;
If $flag = 2$ then $\overline{a}_{h+1} = \overline{a}_h$; $\overline{b}_{h+1} = \max\{[\overline{a}_h, \overline{d}_h - \nu_h] \cap S\}$;
$\overline{u}_h = \overline{d}_h$; $\overline{v}_h = \overline{d}_h$
Until $\overline{b}_h - \overline{a}_h < \dfrac{\varepsilon}{2}$ or $\mathrm{card}[\overline{a}_h, \overline{b}_h] \cap S \leq 2$;
Compute (WEF);
Compute (EF)
else
If $bool = 0$ then
Receive Message from Master $(\underline{a}_k, \underline{b}_k, bool)$ {Slaves execution}
For $ID = 2, 4$ in parallel do
Verify $(\mathrm{card}[\underline{a}_k, \underline{b}_k] \cap S)$;
Take $(\underline{c}_k, \underline{d}_k)$;
Compare $(f(\underline{c}_k), f(\underline{d}_k), flag)$;
End For;
Send Message to Master (flag)
else
if $bool = 1$ then
Receive Message from Master $(\overline{a}_h, \overline{b}_h, bool)$;
For $ID = 2, 4$ in parallel do
Verify $(\mathrm{card}[\overline{a}_h, \overline{b}_h] \cap S)$;
Take $(\overline{c}_h, \overline{d}_h)$;
Compare $(f(\overline{c}_h), f(\overline{d}_h), flag)$;
End For
Send Message to Master (flag);

Remark 12. The cell named "flag" takes values corresponding with the cases enounced in Theorem 2.3, so

$$flag = \begin{cases} 0; & \text{if } I^-_{c,d} \neq \emptyset \text{ or } I^+_{c,d} \neq \emptyset \\ 1; & \text{if } (I^-_{c,d} = \emptyset \text{ and } I^0_{c,d} \neq \emptyset) \text{ or } (I^+_{c,d} = \emptyset \text{ and } I^0_{c,d} \neq \emptyset) \\ 2; & \text{else} \end{cases} .$$

□

Remark 13. Using this type of parallel execution, the amount of computation, and consequently the execution time, reduces at least three times, compared with the serial algorithm. □

REFERENCES

[1] Björklund, H., Sandberg, S., Vorobyov, S., *An Experimental Study of Algorithms for Completely Unimodal Optimization.* Technical report 2002-030, October 2002. Department of Information Technology. Uppsala University.

[2] Chiorean, I., *Parallel calculus*, Cluj-Napoca, Editura Microinformatica, 2000 (in Romanian).

[3] Chiorean, I., Lupsa, L. and Popovici, N., *A Fibonacci type method for determine the set of efficient points of an unimodal multicriteria optimization*, Creative Math. & Inf., **16**, pp. 114–123, 2007.

[4] Demaine, E.D., Langerman, St., *Optimizing a 2D Function Satisfying Unimodality Properties*, www.eurocg.org/www.us.es/ewcg04/Articulos/demaine.ps

[5] Karmanov, V. G., *Programmation mathématique*, Editions Mir, Moscou, 1977.

[6] Luc, D. T., *Theory of Vector Optimization*, Springer-Verlag, Berlin, 1989.

[7] Lupşa, L. and Blaga, L. R., *Optimum points and integer unimodal functions*, Automation Computers Applied Mathematics, **13**, no. 1, pp. 121–130, 2004.

[8] Lupşa, L., Popovici, N., *A new algorithm for solving multicriteria unimodal optimization problems.* Ann. Tiberiu Popoviciu Sem. Functional Equations, Approximation Convexity, **3**, pp. 123–130, 2005.

[9] Lupşa, L., Popovici, N., *Generalized Unimodal Multicriteria Optimization*, Rev. Anal. Numér. Théor Approx., **35**, pp. 65–70, 2006.

[10] Popovici, N., *Multicriteria optimization with unimodal objective functions, Approximation and Optimization*, Proceedings of the International Conference on Approximation and Optimization, Romania, ICAOR, Cluj-Napoca, July 29 - August 1, Vol. **1**, pp. 341–344, 1996.

Received by the editors: February 12, 2008.