Journal of Global Optimization **13:** 61–74, 1998.

Testing the $\Re$- strategy for a Reverse Convex Problem

ALEXANDER S. STREKALOVSKY and IDER TSEVENDORJ
Irkutsk State University, Gagarin Avenue 20, 664003, Irkutsk-3, Russia
E-mail: strekal@math.isu.runnet.ru

(Received 13 June 1995; accepted 13 November 1997)

Abstract. This paper is devoted to solving a reverse-convex problem. The approach presented here is based on Global Optimality Conditions. We propose a general conception of a Global Search Algorithm and develop each part of it. The results of numerical experiments with the dimension up to 400 are also given.

Key words: Global optimization, Global optimality conditions, Global search algorithm

1. Introduction

It is well known that a wide class of applications gives rise to so-called reverse convex problems, i.e. the problems of the form:

$$f(x) \to \min, \quad x \in S, \qquad g(x) \geqslant 0, \tag{P}$$

where g is a convex function, $S \subset R^n$ and f may be even linear [1-5].

Right the constraint $g(x) \geqslant 0$ generates the principal non-convexity, whose immediate consequence is that there exist local solutions which are not global minimizers in (P). Moreover, there is no method which allows to find a global solution in reverse convex problems of large dimension [1-5]. In particular, conspicuous limitation of conventional local optimization methods is their ability of being trapped at a local minimum (or even a stationary point [6-7]).

Therefore, 'the core of a global optimization method is to deal with the question of how to transcend stationarity' [4] or how to escape from a stationary point.

In our opinion, may be, the most promising way in the field is given by the global optimality conditions (GOC) [6-9]. Only GOC allow to recognize that a given stationary point is actually a global minimizer, and if it is not, it allows to proceed to a better feasible point.

On the other hand, for a practice it would be convenient to possess an algorithm, which can find an ε-global solution for large dimension problems. And for realizing how important the role of problems size is, it suffices to see for example [5], where using a modern version of the cut algorithm [1][2], the authors have not obtained satisfactory results for a test reverse convex problem of dimensions less than 10.

Our paper is specially devoted to these two objectives: (1) to demonstrate how to use a mathematical theory for solving an application problem and (2) to study the influence of problem dimension on the solving time.

The paper consists of 4 sections. In Section 2 we will present the GOC disclosing it's algorithmic features. In Section 3 we will introduce a general conception of the algorithm based on GOC, and similar to a so-called $\Re$ - algorithm [10]. Finally, in Section 4 we will present the results of numerical solving a reverse convex problem (similar to the problem from [5]) for a dimension till 400.

Henceforth we'll use ordinary notations from optimization and convex analysis [7]. For example $< \cdot, \cdot >$ stands for the scalar product in R^n, and $f'(x) \in R^n$ is the gradient of a differentiable function $f : R^n \to R$.

We would like to express the gratitude to our respectable referee for the pertinent remarks which allow us to ameliorate the presentation of the paper.

2. Global optimality conditions

Now consider the problem (P), where f is a continuous function on R^n, $S \subset R^n$ and g is a differentiable convex function, such that

$$S \subset (\text{intdom}\, g)\,. \tag{1}$$

Let in addition

$$\left.\begin{array}{l} D = \{x \in R^n / x \in S, g(x) \geqslant 0\}\,, \\ f_* = \inf_x \{f(x)/x \in D\} > -\infty. \end{array}\right\} \tag{2}$$

We will also use the following assumption:

$$\text{There is no global solution } x_* \in D \text{ such that } g\,(x_*) > 0. \tag{G}$$

Then the following result takes place.

THEOREM 1 [9]. *Suppose, the assumption* (G) *holds. If the point z is a global minimizer of* (P) ($z \in Arg \min$ (P)), *then*

$$\left.\begin{array}{l} \forall y : g\,(y) = 0, \quad \forall x \in S : \quad f(x) \leqslant f(z), \\ < g'\,(y)\,, x - y > \leqslant 0. \end{array}\right\} \tag{E1}$$

REMARKS

1) If f is differentiable and S is convex, then it follows from (E1) $(y = z)$

$$\langle \mu g'\,(z) - \lambda f^{'}\,(z)\,, x - z\rangle \leqslant 0 \quad \forall x \in S. \tag{EC}$$

Condition (EC) is the classical local optimality condition in problem (P) [6–7].

2) In order to verify (E1), we have to solve $\forall y : g(y) = 0$ the linearized problem:

$$\langle g'\,(y)\,, x\rangle \to \max, \quad x \in S,\, f\,(x) \leqslant f\,(z)\,; \tag{PL}$$

(whose solution, if one exists, we will denote by $x(y)$), and, after that, we have to check the inequality :

$$\langle g'(y), x(y) - y\rangle \leqslant 0 \quad \forall y : g(y) = 0.$$

If there exists some $\overline{y}$, $g(\overline{y}) = 0$, such that $\langle g'(\overline{y}), x(\overline{y}) - \overline{y}\rangle > 0$, due to convexity of $g(\cdot)$, we obtain $0 < \langle g'(\overline{y}), x(\overline{y}) - \overline{y}\rangle \leqslant g(x(\overline{y})) - g(\overline{y})$. Therefore, $g(x(\overline{y})) > g(\overline{y}) = 0$, $x(\overline{y}) \in S$, $f(x(\overline{y})) \leqslant f(z)$.

Consequently, in virtue of assumption (G) we have a possibility to decrease the function f, beginning at the point $x(\overline{y})$. On the other hand, if z is a stationary point, but not a global solution, there always exist [9] points $\widetilde{y}$, $g(\widetilde{y}) = 0$, and $\widetilde{x} \in S$, $f(\widetilde{x}) \leqslant f(z)$, such that $\langle g'(\widetilde{y}), \widetilde{x} - \widetilde{y}\rangle > 0$.
That allows to ameliorate (as shown above) the value of $f(z)$, i.e. to transcend stationarity, and to escape from a stationary point.

3) We have to note a relation between Theorem 1 and the following result. It has been proved in [3] that if z solves Problem (P), then z is a solution to the problem

$$g(x) \uparrow \max, \quad x \in S, \quad f(x) \leqslant f(z),$$

and $g(z) = 0$. The latter immediatly implies, under assumption (G), the results of Theorem 1. This can be proved, for instance, using linearization machinery for Convex Maximization from [10].

So, one can say that Theorem 1 is a linearization form of the result from [3] mentioned above. According to certain opinions, the first appearance of the type of results for reverse convex problem is contained in [12].

Now we will display the assumptions under which the condition (E1) becomes sufficient for a feasible point z to be a global solution in problem (P).

THEOREM 2 [9]. *Suppose condition* (G) *does not certainly hold, but instead in addition to conditions* (1) *and* (2) *we have*

$$-\infty \leqslant \inf\left(g, R^n\right) < g(z) = 0; \tag{3}$$

$$\left.\begin{array}{l} \forall y \in S,\ g(y) = 0, \quad \exists v = v(y) \in clcoS : \\ \left\langle g'(y), v - y\right\rangle > 0. \end{array}\right\} \tag{4}$$

Then, stronger form of condition (E1)

$$\left.\begin{array}{l} \forall y : g(y) = 0, \quad \forall x \in clcoS, \quad f(x) \leqslant f(z), \\ \left\langle g'(y), x - y\right\rangle \leqslant 0. \end{array}\right\} \tag{E2}$$

is also sufficient for z to be a global solution in problem (P).

Further, if we introduce the function

$$\varphi(z) = \sup_{x,y} \left\{ \left\langle g'(y), x - y \right\rangle / \ g(y) = 0, \quad x \in clcoS, \quad f(x) \leqslant f(z) \right\}, \quad (5)$$

then condition (E2) can be transformed to the following one:

$$\varphi(z) \leqslant 0.$$

Taking into account that $\forall z \in S, \ g(z) = 0,$ the following is obvious $0 = \left\langle g'(z), z - z \right\rangle \leqslant \varphi(z)$, we finally obtain

$$\varphi(z) = 0. \tag{E}$$

We can prove that under certain assumptions similar to (3) and (4) the condition

$$\lim_{k \to \infty} \varphi\left(z^k\right) = 0 \tag{E$'$}$$

is necessary and sufficient for a sequence $\{z^k\}$ to be minimizing in problem (P).

3. General conception of global optimization algorithm for (P)

For the sake of simplicity, in the sequel we will assume S to be convex and closed. According to (5), in order to verify of the GOC (E) we have to maximize the function

$$\Psi(x, y) = \left\langle g'(y), x - y \right\rangle$$

with respect to two variables x and y, such that :

$$x \in S, \quad f(x) \leqslant f(z), \quad g(y) = 0.$$

In order to simplify the problem, we propose to decompose it into two problems, the first of which is (PL) $\forall y : g(y) = 0$. And the second is so-called level problem: $(u \in S, \quad f(u) \leqslant f(z))$

$$h_u(v) \stackrel{\text{def}}{=} \left\langle g'(v), u - v \right\rangle \to \max_v, \quad g(v) = 0. \tag{6}$$

Assume that the problem (6) is solvable. For example, in the case of quadratic function $g(\cdot)$, we can solve the level problem analytically.

LEMMA 1. *If* $g(x) = 1/2 \langle Cx, x \rangle - \gamma$, *where* $\gamma > 0$ *and* C *is a symmetric* $(C = C^\top)$*, definite positive* $(C > 0\,(n \times n))$*-matrix, the level problem solution* $w = w(u)$ *is as follows*

$$w = \mu u, \quad \mu = (\langle Cu, u \rangle / 2\gamma)^{\frac{1}{2}} \tag{7}$$

Suppose $\{\varepsilon_k\}$, $\{\delta_k\}$ are sequences, such that $\varepsilon_k \downarrow 0,\quad \delta_k \downarrow 0 \quad (k \to \infty)$, $\varepsilon_k > 0, \quad \delta_k > 0, \quad k = 0, 1, 2...,$ and we have a local search method (LSM) capable to construct $\forall \varepsilon > 0$ an $\varepsilon-$stationary point to problem (P) [6,7,11].

Let us describe step by step the Global Search Algorithm based on the condition (E) and similar to this one from [10]. Let $k := 0,\ x^0 \in S,\ g(x^0) \geqslant 0$.

Step 1. Beginning from the initial point $x^k \in S,\ g(x^k) \geqslant 0$, and using a LSM, get an ε_k-stationary point

$$z^k \in S, \quad g(z^k) = 0.$$

Step 2. Construct an approximation

$$R_k = \{y^1, y^2, \dots, y^N / \ g(y^i) = 0, \quad i = 1, \dots, N;\ N = N(k)\}$$

of the level surface $g(y) = 0$.

Step 3. $\forall i = 1, \dots, N$ solve the linearized problem with the tolerance δ_k

$$\langle g'(y^i), x\rangle \to \max, \quad x \in S, \quad f(x) \leqslant f(z^k). \qquad (\mathrm{PL}_i)$$

Let u^i be a δ_k-solution of (PL_i).

Step 4. $\forall i = 1, \dots, N$, solve the level problem:

$$h_i(v) = \left\langle g'(v), u^i - v\right\rangle \to \max_v, \quad g(v) = 0. \qquad (6_i)$$

Let w^i be a δ_k-solution of (6_i).

Step 5. Set $\eta_k := \left\langle g'(w^j), u^j - w^j\right\rangle = \max_{1 \leqslant i \leqslant N} \left\langle g'\left(w^i\right), u^i - w^i\right\rangle$.

Step 6. If $\eta_k > 0$, then set $x^{k+1} := u^j,\quad k := k + 1$, and loop to step 1.

Step 7. If $\eta_k \leqslant 0$ and $\varepsilon_k \leqslant \varepsilon_*,\quad \delta_k \leqslant \delta_*$ where $\varepsilon_*,\ \delta_*$ are given tolerances, stop. If $\varepsilon_k > \varepsilon_*$ or $\delta_k > \delta_*$ set $k := k + 1$ and loop to step 1.

REMARKS

4) In order to be theoretically based, we assume that on step 1 it is possible to find a stationary point by a local search method.

   On step 2 we can construct a pertinent approximation (resolving set) of the level surface $g(x) = 0$ (at each iteration).

   On step 3 one can find a global solution of the linearized problem (which may be non-convex!)

   On step 4 one can globally solve the level problem.

   Under these assumptions the description of the $\Re-$ algorithm becomes completely substantiated.

5) It can be easily seen that the sequence $\{z^k\}$ generated by the algorithm above is a sequence of ε_k-stationary points due to describing of step 1.

6) On steps 1 and 3 one is supposed to use standard algorithms of local optimization, if the function $f(\cdot)$ is convex. If it is not, it is necessary to study non-convexity generated by $f(\cdot)$ and to use convenient algorithms for finding a global maximizer of the problem (PL$_i$). This property of the algorithm above can be viewed as advantageous, because it allows to use standard software libraries.

7) When $\eta_k > 0$ (step 6), due to convexity of $g(\cdot)$, we have

$$g(x^{k+1}) - g(z^k) = g(u^j) - g(w^j) \geqslant \langle g'(w^j), u^j - w^j \rangle = \eta_k > 0$$

whence $g(x^{k+1}) > g(z^k) + \eta_k = \eta_k > g(z^k) = 0$.

Therefore, on account of (PL$_i$) it follows

$$f(x^{k+1}) \leqslant f(z^k), \quad x^{k+1} \in S, \quad g(x^{k+1}) > 0.$$

Consequently beginning new local search at x^{k+1} under the assumption (G), or, how they say in [1, 4], when the constraint $g(\cdot)$ is essential, we will obtain z^{k+1} with the property $f(z^{k+1}) < f(z^k)$.

Thus, the algorithm above becomes relaxing, i.e. decreasing the value of $f(\cdot)$ at every iteration, when $\eta_k > 0$.

8a) Clearly, choosing a method for solving the problems (PL$_i$) or (6_i), as well as the local search on Step 1 of the algorithm above is rather difficult, but in some sense standard, ‘already seen’ [6,7,11].

On the other hand, the constructing of the approximation R_k on Step 2 is of the paramount importance from the view point of real global search. Secondly, it is unprecedented, ‘never seen before’ and consequently it gives rise to unforeseen difficulties and novel problems.

So, the crucial question here is on what principle the choice of approximation R_k must be based. We propose to look at the situation from the viewpoint of Global Optimality Conditions.

Suppose that problems (PL$_i$) and (6_i) are solvable and u^i and w^i are their δ-solutions respectively.

DEFINITION 1. *An approximation*

$$R_k = \left\{ y^1, y^2, \dots, y^N / g(y^i) = 0,\ i = 1, \dots, N;\ N = N(k) \right\}$$

is said to be $(z^k, \varepsilon, \delta) - resolving$, *if from the fact that* z^k *is not* ε*-solution to* (P), *i.e.*

$$f\left(z^k\right) > \inf_x \{f(x) / x \in S, g(x) \geqslant 0\} + \varepsilon$$

it follows

$$\max_{1\leqslant i\leqslant N} \left\langle g'(w^i), u^i - w^i \right\rangle \stackrel{\triangle}{=} \eta_k > 0.$$

In other words, if R_k is a $(z^k, \varepsilon, \delta)-$resolving set, and z^k is not an $\varepsilon-$global solution in (P), this allows us to transcend stationarity, i.e. to escape from the ε-stationary point z^k with an obligatory improvement of the value of the objective function $f(\cdot)$ according to Remark 3.

8b) It can be readily seen that on step 2 we don't precise the way to construct a pertinent approximation of the level surface.

In fact, the choice of the way requires the deep acquaintance with the nature and the structure of the problem under study, as well as the deep understanding of GOC (Theorems 1 and 2).

For instance, the existence of R_k follows from the fact that if z^k is not a ε_k-global solution, then there exists [9] some y^k :

$$g(y^k) = 0, \quad \left\langle g'(y^k), x(y^k) - y^k \right\rangle > 0$$

(where $x(y^k)$ is a solution of the linearized problem (PL), with $y = y^k$).

But how should we find or construct R_k for a concrete problem? We have to decide on this question during the solution, this is to be precised in the case. For instance, it will be shown computationally in Section 4 for a concrete problem.

In the same manner, *on step 1*, you are free to choose any local search method, taking into account, first, the nature and the structure of the problem. In addition, the local search must be fast, since we have to repeat it at each iteration of $\Re$-algorithm.(k=0,1,2,...)

Similarly, *on step 3*, for solving the linearized problem you are free to choose a method, which must be, nevertheless, very, very fast, because we have to repeat the solving of the linearized problem several times at each iteration. So, analytical solution (if possible) would be the best!

On step 4, there is also no exact algorithm to solve the level problem (6_i), which must be solved several times at each iteration! You have to solve it by quickest machinery, better analytically! Lemma 1 gives us the good example.

To summarize, one can say, the $\Re$- algorithm is not an algorithm in a sense, but a strategy for solving reverse convex problems of the form (6).

Moreover, we can prove that under the assumptions of Remark 4 the $\Re$- algorithm generates a minimizing sequence $\{z^k\}$ for problem (P).

Since the choice of the resolving set is crucial for the success of global search, it will be reasonable to call the algorithm described above the $\Re$-algorithm, that we will do in the sequel.

By the way, the importance of approximation of the level set $g(x) = 0$ will be demonstrated in section 4, where numerical experiments are presented.

In the following section, we will describe each step of the $\Re$-algorithm for a concrete problem similar to this one from [5].

REMARK

9) It can be easily seen that on steps 3 and 5 one has to solve similar problems (PL_i) or (6_i) for every $i = 1, \dots, N$. Therefore, it would be actually relevant to apply some parallel processing to perform the work.

In such cases, parallel processing can enable us to solve reverse convex problems of substantially larger dimension than those solvable using a serial computer. Some applications require real time solutions. For these applications, it is likely that parallel processing is the only way to obtain acceptable performance. And right the $\Re$- algorithm provides this possibility.

4. Numerical experiments

As noted in the introduction , this section is devoted to numerical solving of the problem [5]:

$$\left.\begin{array}{l} f(x) = \frac{1}{2}\|x - y\|^2 \to \min, \\ -1 \leqslant x_i \leqslant 1, \quad i = 1, \dots, n; \\ g(x) \overset{\triangle}{=} \|x\|^2 - (n - 0.5) \geqslant 0, \end{array}\right\} \qquad \text{(P1)}$$

where $y = (-0.25, 1, \dots, 1)^\top \in R^n$. It can be easily seen that the global solution of (P1) is $x_* = (-\sqrt{0.5}, 1, \dots, 1)^\top$, $f(x_*) = 0.104$, and $x^0 = (1, -1, \dots, -1)^\top \in R^n$ is the worst feasible point.

As displayed in [5], the known ‘cut method’ cannot solve a similar to (P1) problem with a linear objective function beyond the dimension 10.

Here we present, first, the results of numerical solving the problem (P1) for the dimension up to 400 by the $\Re$-algorithm, that gives rise to using the $\Re$-algorithm for solving reverse convex problems of large dimension.

It would be pertinent to do a few remarks on the local search method for the problem (P1) and on solving the corresponding linearized problem $(g(v) = 0, v \neq z, \zeta = f(z))$:

$$\left.\begin{array}{l} \langle g'(v), x\rangle \to \max, \quad f(x) \leqslant \zeta, \\ x \in \Pi = \{x \in R^n \ / \ a_i \leqslant x_i \leqslant b_i, i = 1, \dots, n\} \end{array}\right\} \qquad \text{(PL)}$$

According to remark 8b), which advises to choose a fast and simplest local search method we used a combination of the gradient projection method (when $g(x^s) > 0$) and a variant of linearization method (when $g(x^s) = 0$, see [11]).

Besides, taking into account that the solving of problem (PL) is only a part of the $\Re$- algorithm repeated $N = N(k)$ times at each iteration $(z = z^k)$, we had to solve (PL) rapidly and efficiently. Usage of the analytical solutions of the problems :

$$\left\langle g'(v), x\right\rangle \to \max, \quad x \in \Pi; \tag{PL1}$$

$$\left\langle g'(v), x\right\rangle \to \max, \quad f(x) \leqslant \zeta; \tag{PL2}$$

considerably facilitates the (PL)– solving.

Now let us demonstrate the considerable influence of the choice of a level set approximation on the solution, solving time and the volume of work. Since it is not yet analytically constructed of any resolving set for (P1), during the tests we used the following approximations of the level set

$$U = \{x \,/\, g(x) = 0 = g(z)\} :$$

$$R_1 \stackrel{\triangle}{=} \left\{ \begin{array}{l} v^i = z - \theta_i e^i = (z_1, \dots, z_{i-1}, -z_i, z_{i+1}, \dots z), e^i = (0, \dots, 1_i, \dots, 0), \\ \theta_i : g(v^i(\theta)) = 0, i = 1, \dots, n \end{array} \right\}$$

$$R_2 \stackrel{\triangle}{=} \left\{ v^i = (z_1, \dots, z_{i-1}, -z_i, -z_{i+1}, \dots, z_n), i = 1, \dots, n-1 \right\}$$

$$R_3 \stackrel{\triangle}{=} \left\{ \begin{array}{l} v^1 = (-z_1, -z_2, z_3, \dots, z_n), \quad v^n = (z_1, \dots, z_{n-2}, -z_{n-1}, -z_n), \\ v^i = (z_1, \dots, z_{i-2}, -z_{i-1}, -z_i, -z_{i+1}, z_{i+2}, \dots z_n), i = 2, \dots, n-1, \end{array} \right\}$$

$$R_{12} \stackrel{\triangle}{=} \left\{ \begin{array}{l} v^1 = (-z_1, z_2, \dots, z_n), \quad v^{n+1} = (z_1, \dots, z_{n-1}, -z_n) \\ v^i = (z_1, \dots, z_{i-2}, -z_{i-1}, -z_i, z_{i+1}, \dots, z_n), i = 2, \dots, n, \end{array} \right\}$$

$$R_{21} \stackrel{\triangle}{=} R_2 \cup \left\{ v^n = (z_1, \dots, z_{n-1}, -z_n) \right\}$$

$$R_{22} \stackrel{\triangle}{=} R_2 \cup \left\{ v^n = (-z_1, -z_2, \dots, -z_n) \right\}$$

For solving the problem (PL_i) and for local search in (P1), the corresponding algorithms above were used. The solution of the level problem (6_i) was obtained according to Lemma 1.

Let n be the dimension of the problem,
$f_0 = f\left(x^0\right)$ –- the initial value of the function $f(\cdot)$,
R-the level set approximation, f_m-the best obtained value of $f(\cdot)$,
St-the number of obtained stationary points, from what one managed to escape,
LP-the number of linearized problems (PL_i) obtained during the solution, and finally, T be the time of solving (min:sec).

The tests were performed on a serial PC/AT IBM- 386.

Naturally, we have first tested the problem of small dimension (3–8).

Having analyzed the results, we excluded from the consideration the worst approximations R_3 and R_{22}. Studying Tables 1 and 2 one can easily find out that the approximation R_1 enables us to obtain the global solution for all dimensions,

Table 1.

n	f_0	R	f_m	St	LP	T
3	4.781	R_1	0.104	4	9	00:00.43
		R_2	0.4573	4	6	00:00.44
		R_3	0.4573	4	8	00:00.60
		R_{12}	0.104	7	14	00:01.27
		R_{21}	0.4573	4	7	00:00.61
		R_{22}	0.4573	4	7	00:00.55
4	6.781	R_{12}	0.104	11	25	00:02.08
5	8.781	R_1	0.104	6	20	00:01.15
		R_2	4.241	5	11	00:00.88
		R_3	2.2419	4	10	00:00.82
		R_{12}	0.104	12	32	00:03.46
		R_{21}	2.2419	6	17	00:01.32
		R_{22}	2.2419	7	18	00:01.32
6	10.781	R_2	3.7482	4	13	00:01.26
7	12.781	R_1	0.104	8	35	00:02.52
		R_2	1.740	5	21	00:01.75
		R_3	4.2641	5	17	00:01.21
		R_{12}	1.740	7	29	00:02.14
		R_{21}	1.740	5	22	00:01.75
		R_{22}	1.740	5	22	00:01.81
8	14.781	R_2	3.7490	5	22	00:02.08
		R_{12}	1.7476	8	39	00:03.41

while other approximations have not solved the problem (P2) for some dimensions. Moreover, some approximation may enable us to find the solution for some dimension but does not provide it for close dimensions. To get convinced of this, it suffices to analyze Table 3.

The conclusion is obvious: having obtained first promising numeric results, a mathematician must be careful concerning different choices of resolving sets. The same thing can be demonstrated by Table 4.

Now, we present the numeric solving results for Problem (P2) of dimension up to 100 by the $\Re$-algorithm using the approximations R_1, R_2, R_{12} . Here one can see, in particular, the influence of the problem dimension on the solving time.

Regardless of the promising results for R_2 and R_{12}, we must remember Tables 1, 2 and 3 and be always very careful concerning their application to problem (P1).

Table 2.

n	f_0	R	f_m	St	LP	T
9	16.781	R_1	0.104	10	54	00:03.90
		R_{12}	0.104	8	42	00:03.13
		R_{21}	0.104	6	34	00:02.75
		R_2	1.747	5	24	00:02.08
10	18.781	R_1	0.104	11	65	00:04.45
		R_{12}	1.7475	9	53	00:04.45
		R_{21}	0.104	7	44	00:03.68
		R_2	3.7413	6	33	00:01.92
11	20.781	R_1	0.104	12	77	00:05.38
		R_2	1.7475	6	35	00:02.86
		R_{12}	0.104	9	56	00:03.73
		R_{21}	0.104	7	47	00:03.73
12	22.781	R_2	3.7631	7	46	00:02.53
		R_{12}	1.7473	10	69	00:04.67
13	24.721	R_1	0.104	14	104	00:07.30
		R_2	1.7474	7	48	00:03.90
		R_{12}	0.104	10	72	00:05.43
		R_{21}	0.104	8	62	00:05.00
14	26.781	R_2	3.7627	8	61	00:00.35
		R_{12}	1.7474	11	87	00:06.04
15	28.781	R_1	0.104	16	135	00:09.45
		R_2	1.7474	8	63	00:05.05
		R_{12}	0.104	11	90	00:06.87
		R_{21}	0.104	9	79	00:06.42

Table 3.

n	f_0	R	f_m	St	LP	T
14	26.781	R_2	3.7627	8	61	00:03.35
15	28.781	R_2	1.7474	8	63	00:05.05
16	30.781	R_2	0.104	9	79	00:04.89
17	32.781	R_2	1.7474	9	80	00:06.65
18	34.781	R_2	0.104	10	98	00:06.73
19	36.781	R_2	1.7474	10	99	00:08.30

Table 4.

n	f_0	R	f_m	St	LP	T
		R_1	0.104	21	230	00:14.01
20	38.781	R_2	0.104	11	119	00:06.59
		R_{12}	0.104	13	133	00:09.39
		R_1	0.104	31	495	00:33.01
30	58.781	R_2	0.104	16	252	00:15.60
		R_{12}	0.104	18	273	00:20.98
		R_1	0.104	41	860	01:01.36
40	78.781	R_2	0.104	21	439	00:29.33
		R_{12}	0.104	13	463	00:38.12
		R_1	0.104	51	1325	01:44.25
50	98.781	R_2	0.104	26	674	00:49.79
		R_{12}	0.104	28	703	01:03.05
		R_1	0.104	61	1890	02:39.45
60	118.781	R_2	0.104	31	959	01:16.40
		R_{12}	0.104	33	993	01:35.03
		R_1	0.104	71	2555	03:55.96
70	138.781	R_2	0.104	36	1294	01:53.31
		R_{12}	0.104	38	1333	02:18.74
		R_1	0.104	81	3320	05:27.30
80	158.781	R_2	0.104	41	1679	02:37.41
		R_{12}	0.104	43	1723	03:12.07
		R_1	0.104	91	4125	07:21.32
90	178.781	R_2	0.104	46	2114	03:32.70
		R_{12}	0.104	48	2163	04:13.81
		R_1	0.104	101	5150	09:33.97
100	198.781	R_2	0.104	51	2599	04:37.53
		R_{12}	0.104	53	2653	05:30.26

So, very likely, we have now the numerical demonstration of fundamental importance of the resolving set selection for solving a reverse convex problem.

In some sense one can say that solving a reverse convex problem is reduced to constructing a resolving set. This means that using the $\mathfrak{R}$-algorithm for solving a reverse convex problem, we actually apply the new information about the problem in the form of Global Optimality Conditions.

Table 5.

n	f_0	R	f_m	St	LP	T
5	8.781	R_{20}	0.104	6	17	00:01.76
10	18.781	R_{20}	0.104	7	44	00:03.40
20	38.781	R_{20}	0.104	11	120	00:08.68
30	58.781	R_{20}	0.104	16	255	00:19.88
40	78.781	R_{20}	0.104	21	440	00:37.30
50	98.781	R_{20}	0.104	26	675	01:02.45
60	118.781	R_{20}	0.104	31	960	01:35.89
70	138.781	R_{20}	0.104	36	1295	02:22.75
80	158.781	R_{20}	0.104	41	1680	03:18.23
90	178.781	R_{20}	0.104	46	2115	04:25.13
100	198.781	R_{20}	0.104	51	2600	05:47.46
150	298.781	R_{20}	0.104	76	5775	16:27.12
200	398.781	R_{20}	0.104	101	10200	34:02.52
300	598.781	R_{20}	0.104	151	22800	01hr:44:34.80
400	798.781	R_{20}	0.104	201	25136	03hr:58:18.65

Finally, we display the best obtained results for solving problem (P1), which have been gained by using the following approximation:

$$R_{20} = \left\{ \begin{array}{l} v^i = (z_1, \dots, z_{i-1}, -z_i, -z_{i+1}, z_{i+2}, \dots, z_n)^\top, i = 1.n-1; \\ \\ v^n = z - 2f'(z) \cdot < f'(z), z > / \; \|f'(z)\|^2 \end{array} \right\}$$

This approximation enables us to find the global solution for all the considered dimensions, as well as R_1, but the solving time by R_{20} is smaller than that one by R_1. In other words, one can say that, in this case, we were successful to construct computationally the resolving sets for the problem (P1).

5. Conclusion

In this paper we considered the general deterministic approach for solving a reverse convex problem based on Global Optimality Conditions (GOC).
In particular,

a) we have introduced the global search algorithm based on GOC, so-called $\Re$-algorithm;

b) further, we have studied certain features of $\Re$-algorithms application for a concrete problem;

c) finally, we have demonstrated the possibility of solving the reverse convex problem for large dimensions by using the $\mathfrak{R}$-algorithm and studied the influence of a level set approximation on the solution as well as on the solving time.

References

1. Horst R. and Tuy H. (1990), *Global Optimization (Deterministic Approach)*, Springer Verlag.
2. Pardalos P. M. and Rosen J. B. (1987), *Constrained global optimization: Algorithms and applications*, Lecture Notes in Computer Science 268, Springer Verlag.
3. Tuy H. (1987), Convex programs with an additional reverse convex constraint, *Journal of Optimization Theory and Applicattions* 52: 463–485.
4. Tuy H. (1992), The complementary Convex Structure in Global Optimization, *Journal of Global Optimization* 2: 21–40.
5. Gurlitz T. R. and Jacobsen S. E. (1991), On the use of cuts in reverse convex programs, *Journal of Optimization Theory and Applications* 68: 257–274.
6. Vasil'yev F. P. (1988), *Numerical Methods of Extremum Problems Solving*, Nauka, Moscow.
7. Hiriart-Urruty J.-B. and Lemarchal C. (1993), *Convex Analysis and Minimization Algorithms*, Grundlehrender Mathematishen Wissenschaften, vol. 305 and 306, Springer Verlag.
8. Hiriart-Urruty J.-B (1989), *From convex optimization to non-convex optimization, Part 1: Necessary and sufficient conditions for global optimality, in Nonsmooth Optimization and Related Topics*, Plenum Press, pp. 219–239.
9. Strekalovsky A, (1993), *Extremal problems on complements of convex sets*, Translated from Kibernetika i Sistemnyi Analiz, No. 1, pp 88–100, Plenum Publishing Corporation.
10. Strekalovsky A. (1993), The search for a global maximum of a convex functional on an admissible set, *Comput. Math. and Math Physics* 33(3): 315–328, Pergamon Press.
11. Meyer R., (1970), The validity of a family of optimization methods, *SIAM Journal on Control* 8: 41–54.
12. Bansal P.P. and Jacobsen S.E. (1975), An algorithm for optimizing network flow capacity under economy of Scale, *Journal of Optimization Theory and Applications* 15: 565–586.