УДК 519.853.4

НОВЫЙ ПОДХОД К НЕВЫПУКЛОЙ ОПТИМИЗАЦИИ

А. С. Стрекаловский[1], А. В. Орлов[1]

В работе предлагается новый подход к решению непрерывных невыпуклых задач оптимизации, основанный на условиях глобальной оптимальности. Детально представлена методика решения трех задач: задачи о полиэдральной отделимости, систем нелинейных уравнений и отыскания ситуации равновесия по Нэшу в биматричных играх посредством вариационного подхода с использованием методологии глобального поиска. Работа выполнена при поддержке РФФИ (проект № 05–01–00110) и гранта Президента РФ (проект № МК–6580.2006.1).

Ключевые слова: невыпуклая оптимизация, условия глобальной оптимальности, локальный поиск, глобальный поиск, вычислительный эксперимент.

1. Introduction. Many optimization problems arising in different application areas are really nonconvex [1 – 6]. Most of them deal with the functions that can be represented as a difference of two convex functions (the so-called d.c. functions).

Nowadays, the situation in the continuous nonconvex optimization may be characterized as the dominance of methods borrowed from other sciences [1 – 4], such as discrete optimization (the branch-and-bound method, cut methods, outside and inside approximations, vertex enumeration, and so on), physics, chemistry (simulated annealing methods), biology (genetic and ant colony algorithms), etc.

On the other hand, the classical methods of convex optimization [6 – 8] have been discarded because of their inefficiency [1 – 4]. It is well known that the serious limitation of convex optimization methods applied to nonconvex problems consists in their ability of being trapped at a local extremum or even at a critical point, depending on a starting point [1 – 4]. Thus, the classical apparatus proves itself inapplicable for new problems arising in practice.

In such a situation it seems quite improbable to create an approach for finding a global solution to nonconvex problems in such a way that this approach be connected with the convex optimization theory and be based on using the methods of convex optimization.

Nevertheless, we risked to propose such an approach [9 – 20] and even to advance the following principles of nonconvex optimization.

1. Linearization of the basic (generic) nonconvexity of the problem under study and, consequently, a reduction of the problem to a family of (partially) linearized problems.

2. Application of convex optimization methods for solving linearized problems and, as a consequence, within special local search methods.

3. Construction of "good" (pertinent) approximations (resolving sets) of level surfaces and epigraph boundaries of convex functions.

Obviously, the first and second principles are well known. The depth and efficiency of the third one can be observed in [9 – 20].

Having developed these principles, we come to a methodology for solving the nonconvex problems that can be represented as follows.

1. Exact classification of the problem under study.

2. Application of special local search methods.

3. Application of special conceptual global search methods (strategies).

4. Construction of pertinent approximations of level surfaces of corresponding convex functions with the aid of the experience obtained by solving similar nonconvex problems.

5. Application of convex optimization methods for solving linearized problems and within the frames of special local search methods.

It can be readily seen that this approach lifts the classical convex optimization up to a new altitude, where the efficiency and the time-optimality of the methods become of prime importance not only for convex but for nonconvex optimization as well.

[1] Институт динамики систем и теории управления СО РАН (ИДСТУ СО РАН), ул. Лермонтова, д. 134, 664033, г. Иркутск; e-mail: strekal@icc.ru, anor@icc.ru

Our computational experience gives the evidence that, if you follow the above methodology, you have a more reliable chance to reach the global solution to a nonconvex problem of large size ($\geqslant 1000$) than with the use of the branch-and-bound or cut methods.

At present, together with our colleagues we are investigating the following problems on the basis of the approach described above.

1. Solving a system of nonlinear equations by reducing the problem to an optimization problem (the variational approach).

2. The problem of polyhedral separability when you have to separate two finite sets of points in $\mathbb{R}^n$ by means of a minimum number of hyperplanes.

3. Finding the Nash equilibrium points in bimatrix games with the aid of a special bilinear programming problem.

For solving these problems one needs some background of the d.c. programming theory. Therefore, in this paper we first recall the general schemes of local and global searches in d.c. problems. In the next sections of the paper, some particular problems of d.c. minimization are considered in detail and the efficiency of the approach is illustrated by numerical experiments.

2. Theoretical foundations of d.c. minimization. Consider the problem

$$F(x) = G(x) - H(x) \downarrow \min, \quad x \in \mathbb{R}^n, \tag{2.1}$$

where the functions $G(x)$ and $H(x)$ are convex. Assume that at any point $x \in \mathbb{R}^n$ it is possible to find some subgradient $x^* \in \partial H(x)$.

As mentioned above, the development of special local search methods for each class of nonconvex problems is one of the main stages in the global search. In this case the definition of a critical point depends on properties of the problem and on the corresponding local search method.

Similarly to the situation in the differentiable case, the classical convex optimization methods cannot be directly applied in a nonconvex case even for a local search. Some simple examples illustrate this fact [7−9].

In order to solve problem (2.1), we apply the special local search method described below. This method is a generalization of the local search method for the differentiable constrained d.c. minimization [9, 15, 17, 18].

Let $x^0 \in \mathbb{R}^n$ be a starting point. If a point $x^s \in \mathbb{R}^n$ is known, then we seek the next point x^{s+1} as a solution to the following problem linearized at the point x^s:

$$J_s(x) = G(x) - \langle x_s^*, x \rangle \downarrow \min, \quad x \in \mathbb{R}^n, \quad x_s^* \in \partial H(x^s). \tag{2.2}$$

More precisely, this means that the point x^{s+1} satisfies the inequality

$$G(x^{s+1}) - \langle x_s^*, x^{s+1} \rangle \leqslant \inf_x \bigl\{ G(x) - \langle x_s^*, x \rangle \bigr\} + \delta_s, \tag{2.3}$$

where the conditions

$$\delta_s \geqslant 0, \quad s = 1, 2, \ldots, \quad \sum_{s=0}^{\infty} \delta_s < \infty \tag{2.4}$$

hold for the sequence $\{\delta_s\}$.

Note that problem (2.2) is convex and, consequently, is solvable by contemporary packages and libraries of linear and convex optimization methods, e.g., ILOG CPLEX and/or MINOS.

The convergence theorem for the corresponding algorithm is formulated as follows.

Theorem 1. *Let the goal function $F = G - H$ in problem* (2.1) *be lower bounded and at any point x one can find a subgradient $x^* \in \partial H(x)$.*

(*i*) *Then the sequence $\{x^s\}$ generated by rule* (2.3) *satisfies the condition*

$$\lim_{s\to\infty} \Bigl\{ \inf_x \bigl[G(x) - G(x^{s+1}) - \langle x_s^*, x - x^{s+1} \rangle \bigr] \Bigr\} = 0, \quad x_s^* \in \partial H(x^s). \tag{2.5}$$

(*ii*) *Further, if for a limit point z of the sequence $\{x^s\}$ the corresponding sequence of subgradients is also convergent, i.e.*

$$x_s^* \to z^* \in \partial H(z), \tag{2.6}$$

then the following condition holds:

$$\inf_x \bigl\{ G(x) - G(z) - \langle z^*, x - z \rangle \bigr\} = 0, \quad z^* \in \partial H(z). \tag{2.7}$$

This means that the point z is a solution to the convex problem

$$G(x) - \langle z^*, x\rangle \downarrow \min, \quad x \in \mathbb{R}^n. \tag{2.8}$$

In the sequel, such a point z is said to be critical for problem (2.1).

(*iii*) *In the differentiable case, from* (2.7) *the classical optimality condition follows:*

$$\nabla G(z) - \nabla H(z) = 0. \tag{2.9}$$

The proof of this theorem is similar to the differentiable constrained case [9, 15, 17, 18].

In practical applications of the above method, due to the structure of the problem, the key role is played by the selection of the solving method for the linearized problem (2.2) from the libraries of standard convex optimization methods.

Now, let us formulate some global optimality conditions for problem (2.1) (see also [9, 12, 15, 17, 18]).

Theorem 2. *If a point $z \in \mathbb{R}^n$ is a global solution to problem* (2.1), *then*

$$\begin{aligned} &\forall (y,\gamma) : \gamma - H(y) = \zeta \triangleq F(z), \quad G(y) \leqslant \gamma \leqslant \sup(G), \quad \forall y^* \in \partial H(y) : \\ &G(x) - \langle y^*, x\rangle \geqslant \gamma - \langle y^*, y\rangle \quad \forall x \in \mathbb{R}^n. \end{aligned} \tag{2.10}$$

If, in addition, the condition $\exists v \in \mathbb{R}^n : F(v) > F(z) \triangleq \zeta$ holds, then condition (2.10) *becomes sufficient for the point $z \in \mathbb{R}^n$ to be a global solution to problem* (2.1).

Properties and features of these conditions are discussed in [9, 12, 15, 17, 18]. Note that one of the principal properties can be formulated as follows. If a point z is not a global solution, we can construct a new point being better than z. In the works mentioned above, we generalized these conditions for the minimizing sequences.

In order to decide whether the point under consideration is a global solution to problem (2.1), Theorem 2 suggests the consideration of the following auxiliary problem:

$$\begin{aligned} &\Psi(x, y, y^*, \gamma) \triangleq \langle y^*, x - y\rangle + \gamma - G(x) \uparrow \max_{x,y,y^*,\gamma}, \\ &y^* \in \partial H(y), \quad x \in D, \quad (y,\gamma) \in \mathbb{R}^{n+1} : \gamma - H(y) = \zeta \triangleq F(z), \quad G(y) \leqslant \gamma \leqslant \sup(G, D). \end{aligned} \tag{2.11}$$

Moreover, if we need to examine only the necessary condition, then the last inequalities in (2.11) cannot be taken into account.

In [9, 15, 17, 18] it had been proved that, on each iteration of the global search, problem (2.11) may be solved approximately and partially. This result can be generalized to the nondifferentiable case.

Nevertheless, problem (2.11) seems to be as difficult as the original one (2.1). Therefore, by analogy with [9, 15, 17, 18], it is suggested to decompose the problem into several simpler ones.

a) Suppose we have a number γ such that $\gamma_- \triangleq \inf(G, D) \leqslant \gamma \leqslant \gamma_+ \triangleq \sup(G, D)$. Then for the level surface $\big(\zeta \triangleq F(z)\big)$ $Y(\zeta, \gamma) = \big\{y \in \mathbb{R}^n \,|\, H(y) = \gamma - \zeta\big\}$ of the function $H(\cdot)$ one has to construct an approximation $\mathcal{A}(\zeta,\gamma) = \big\{y^1, \dots, y^N \,|\, H(y^i) = \gamma - \zeta,\ G(y^i) \leqslant \gamma,\ i = 1, \dots, N\big\}$.

b) For each $y^i \in \mathcal{A}(\zeta,\gamma)$ and some $y_i^* \in \partial H(y^i)$, we solve the linearized problem

$$G(x) - \langle y_i^*, x\rangle \downarrow \min, \quad x \in D. \tag{2.12}$$

Let u^i be an approximate global solution to (2.12).

c) Finally, for each $i = 1, \dots, N$ we solve the so-called level problem

$$\big\langle v^*, u^i - v\big\rangle \uparrow \max_{v,v^*}, \quad H(v) = \gamma - \zeta, \quad v^* \in \partial H(v). \tag{2.13}$$

Let w^i, w_i^* be an approximate global solution to (2.13).

d) As a result, by solving three basic subproblems one obtains the number $\eta(\zeta,\gamma) := \gamma + \eta_0(\zeta,\gamma)$, where

$$\eta_0(\zeta,\gamma) = \langle w_j^*, u^j - w^j\rangle - g(u^j) \triangleq \max_{1\leqslant i\leqslant N} \big\{\langle w_i^*, u^i - w^i\rangle - G(u^i)\big\}; \quad w_j^* \in \partial H(w^j), \quad w_i^* \in \partial H(w^i).$$

If $\eta(\zeta,\gamma) > 0$, then the point u^j turns out to be better than z, since $w_j^* \in \partial H(w^j)$, and due to convexity of $H(\cdot)$ one has $0 < \gamma + \big\langle w_j^*, u^j - w^j\big\rangle - g(u^j) \leqslant H(u^j) - H(w^j) + \gamma - G(u^j) = F(z) - F(u^j)$.

The latter is equivalent to $F(u^j) < F(z)$, so that one may pass to the next iteration by putting $x^{k+1} := u^j$. If $\eta(\zeta,\gamma) \leqslant 0$, then a new evaluation $\overline{\gamma} := \gamma + \Delta\gamma$ is selected with the aid of a one-dimensional optimization method for the function $\Phi(\gamma) := \eta(\zeta,\gamma)$ considered on the segment $[\gamma_-,\gamma_+]$ [7, 8]. Henceforth, we use the passive search [7] everywhere. If $\eta(\zeta,\gamma) \leqslant 0$ $\forall\gamma \in [\gamma_-,\gamma_+]$, then the one-dimensional search for γ is terminated. In this case one can stop, since the current critical point is the sought-for solution to the problem under consideration.

Sometimes, instead of solving problem (2.13) and computing the value $\eta(\zeta,\gamma)$, one directly compares the values $F(u^i)$ of the goal function with the value $\zeta \triangleq F(z)$. It can easily be seen that $\eta(\zeta,\gamma) > 0$ if and only if the point u^j is better than the point z.

Now, one can describe the global search strategy for the nondifferentiable d.c. minimization problem (2.1) [9, 15, 17, 18]. As known, the strategy consists of the stages a) – d) and some additional steps arising by virtue of the features of the problem under consideration.

Note that, in contrast to the global search algorithm for smooth d.c. minimization problems, we have to use an element of the subdifferential of the function $H(\cdot)$ at a current point instead of the gradient. It can be readily seen that the subproblems of stages b) and c) are also nondifferentiable and therefore need some special methods to be solved.

In the following sections of this paper we present some variants of the global search strategy in greater detail, while taking into account specific properties of the problems under scrutiny.

3. Nonlinear equations. This section of the paper is devoted to one of the classical branches of numerical mathematics, which has various applications in many areas of contemporary natural sciences, from economics to ecology. This implies finding a solution to the systems of nonlinear equations [21, 22].

Despite of a wide spectrum of methods developed in this area, the problem of numerical solution of such systems remains rather actual, first of all, because of the fact (when applying, say, the Newton–Kantorovich method) that there appear some difficulties caused by choosing a suitable starting point ensuring the convergence to a solution [21, 22]. On the other hand, the complexity of searching for the starting point increases with the system's size.

When applying variational methods, one deals with nonconvex optimization problems; hence, as mentioned above, the classical methods of convex optimization are inapplicable for solving such problems [7, 8].

In this section the process of solving systems of nonlinear equations with d.c. functions is considered. For the purpose of finding a solution for this class of systems, it is suitable to use the apparatus of d.c. minimization [9] based on global optimality conditions and outlined in the previous section. In addition we note that the problem of finding a suitable starting point disappears. Moreover, the global search algorithm turns out to be able to carry out an auxiliary operation, e.g., to find suitable initial points for a further application, say, of the Newton–Kantorovich method.

3.1. Problem statement and the d.c. decomposition of a goal function. The following system of nonlinear equations is considered:

$$f_i(x) = 0, \quad i = 1,\dots,m, \tag{3.1}$$

where $x \in \mathbb{R}^n$ and $f_i(x)$ are d.c. functions, i.e.

$$f_i(x) = g_i(x) - h_i(x), \tag{3.2}$$

where $g_i(x)$, $h_i(x)$ are convex functions on $\mathbb{R}^n$, $i = 1,\dots,m$. It is well known [22] that this system of equations can be represented as the optimization problem

$$\Phi(x) = F\bigl(f_1(x),\dots,f_m(x)\bigr) \downarrow \min, \quad x \in \mathbb{R}^n, \tag{3.3}$$

where the goal function can be taken, e.g., in the form

$$F(x) = \sum_{i=1}^{m} \bigl|f_i(x)\bigr|. \tag{3.4}$$

It can easily be seen that problem (3.3), (3.4) turns out to be generally nonconvex, and the question of how to solve it remains open.

Nevertheless, using the known properties of d.c. functions [4, 9], one can show that for function (3.4) the following representation takes place:

$$F(x) = G(x) - H(x), \tag{3.5}$$

where $G(x) = 2\sum_{i=1}^{m} \max\{g_i(x), h_i(x)\}$ and $H(x) = \sum_{i=1}^{m}\big(g_i(x) + h_i(x)\big)$ are obviously convex functions.

Since the function $F(x)$ given by formula (3.4) is a d.c. function, the apparatus of d.c. minimization can be applied for solving problem (3.3), (3.4).

3.2. Global search algorithm. Consider the principal stages of global search. Our first task is to obtain a critical point for problem (3.3), (3.4). For this purpose we use the special local search method described above.

Let us formulate the linearized problem. Since the d.c. decomposition of the goal function is given by (3.5), the second term $H(x) = \sum_{i=1}^{m}\big(g_i(x) + h_i(x)\big)$, $x \in \mathbb{R}^n$, turns out to be a smooth function. Then problem (2.2) can be represented as follows:

$$J_s(x) = G(x) - \big\langle \nabla H(x^s), x\big\rangle \downarrow \min, \quad x \in \mathbb{R}^n. \tag{3.6}$$

However, problem (3.6) is nondifferentiable [23], because the function $G(x) = 2\sum\limits_{i=1}^{n} \max\{g_i(x), h_i(x)\}$ is obviously nonsmooth.

Therefore, for solving problem (3.6) one needs to apply nondifferentiable optimization methods. To accomplish this, we apply the r-algorithm of N. Shor. Recall that Shor's r-algorithm uses the operation of space stretching among the directions of the difference of two consecutive subgradients [24, 25]. According to [25], this method is one of the best among similar methods from the practical point of view.

Let us proceed to finding a solution to system (3.1). Here we present a global search algorithm for problem (3.3), (3.4). Principally, this algorithm consists of the stages a) – d) described in the previous section.

Let $x_0 \in \mathbb{R}^n$ be a starting point, $\{\tau_k\}$ and $\{\delta_k\}$ be numerical sequences, $\tau_k, \delta_k > 0$, $k = 0, 1, 2, \ldots$, $\tau_k \downarrow 0$, $\delta_k \downarrow 0$ $(k \to \infty)$. In addition, assume that we are given a number γ_+.

Step 0. Set $k := 0$, $x^k := x_0$.

Step 1. Starting with $x^k \in \mathbb{R}^n$, obtain the τ_k-critical point z^k by the special local search method (see the previous subsection). Set $\zeta_k := F(z^k) \leqslant F(x^k)$.

Step 2. Stop if $F(z^k) < \varepsilon$, where ε is a given accuracy and z^k is an approximate solution to system (3.1).

Step 3. Select some $\gamma \in [0, \gamma_+]$.

Step 4. Construct the approximation

$$\mathcal{A}_k(\gamma) = \big\{v^1, \ldots, v^{N_k} \,\big|\, H(v^i) = \gamma - \zeta_k,\ i = 1, \ldots, N_k,\ N_k = N_k(\gamma)\big\}. \tag{3.7}$$

Step 5. For each $i = 1, \ldots, N_k$, find $u^i \in \mathbb{R}^n$ such that

$$G(u^i) - \big\langle \nabla H(v^i), u^i\big\rangle - \delta_k \leqslant \inf_x \Big\{G(x) - \big\langle \nabla H(v^i), x\big\rangle \,\big|\, x \in \mathbb{R}^n\Big\}. \tag{3.8}$$

Step 6. For each $i = 1, \ldots, N_k$, find w^i:

$$\big\langle \nabla H(w^i), u^i - w^i\big\rangle + \delta_k \geqslant \sup_v \Big\{\big\langle \nabla H(v), u^i - v\big\rangle \,\big|\, H(v) = \gamma - \zeta_k\Big\}. \tag{3.9}$$

Step 7. Set $\eta_k(\gamma) := \eta_k^0(\gamma) + \gamma$, where

$$\eta_k^0(\gamma) := \big\langle \nabla H(w^j), u^j - w^j\big\rangle - G(u^j) \triangleq \max_{1 \leqslant i \leqslant N} \Big\{\big\langle \nabla H(w^i), u^i - w^i\big\rangle - G(u^i)\Big\}. \tag{3.10}$$

Step 8. If $\eta_k(\gamma) > 0$, then set $x^{k+1} := u^j$ and go to Step 1.

Step 9. If $\eta_k(\gamma) \leqslant 0$, then set $\gamma := \gamma + \triangle\gamma \in [0, \gamma_+]$ and go to Step 3.

Step 10. If $\eta_k(\gamma) \leqslant 0 \quad \forall \gamma \in [0, \gamma_+]$ (i.e., the one-dimensional search for γ is completed), then stop. The iterative process is terminated.

It can be readily seen that the new stopping criterion is added to the global search scheme due to the statement of problem (3.3), (3.4), since it is clear that, if the value of the goal function is close to zero, we obtain an approximate solution to system (3.1).

The peculiarity of the above-proposed method consists in the implementation of Steps 4, 5 and 6. Actually, at Step 4 we have to construct an approximation of the level surface; at Steps 5 and 6 we have to solve the linearized problem

$$G(x) - \big\langle \nabla H(v^i), x\big\rangle \downarrow \min_x \tag{3.11}$$

and the level problem

$$\big\langle \nabla H(v), u^i - v \big\rangle \uparrow \max_v, \quad H(v) = \gamma - \zeta_k. \tag{3.12}$$

Note that the linearized problem (3.11) coincides with problem (2.2) solved at each step of the special local search method described above.

3.3. Implementation of the global search algorithm. The first test of the global search method for solving the above systems of equations was carried out with the following quadratic systems:

$$f_i(x) \triangleq \frac{1}{2}\,\langle C_i x, x\rangle + \langle b_i, x\rangle + d_i = 0, \quad i = 1,\dots,n, \tag{3.13}$$

where $x \in \mathbb{R}^n$ and C_i are symmetric $(n \times n)$ matrices, not necessarily positive definite. As is known [21], such a matrix can be represented in the form of the difference of two symmetric positive definite matrices: $C_i = A_i - B_i$. This fact leads to the following d.c. representation of the quadratic functions:

$$f_i(x) \triangleq g_i(x) - h_i(x), \quad i = 1,\dots,n, \tag{3.14}$$

where $g_i(x) = \dfrac{1}{2}\,\langle A_i x, x\rangle$ and $h(x) = \dfrac{1}{2}\,\langle B_i x, x\rangle - \langle b_i, x\rangle - d_i$ are convex functions, $A_i, B_i > 0$, $A_i = A_i^T$, $B_i = B_i^T$.

In the case of the quadratic systems of equations, the analytical solution can be given for the level problem (3.12) [9]. For the one-dimensional search of γ, we split the segment $[0, \gamma_+]$ into a finite number of parts according to the dimension of the systems. For constructing the level surface approximation, we use the previous experience of solving various nonconvex problems [9 – 20].

For example, one can choose the points of approximation of the form $y^i = z^k - \lambda_i v^i$, $i = 1,\dots,N_k$, where v^i is a vector constructed, generally speaking, with the aid of the information about the problem under consideration; λ_i is a scalar that verifies the conditions $H(y^i) = \dfrac{1}{2}\,\langle S y^i, y^i\rangle - \langle q, y^i\rangle - \alpha = \gamma - \zeta$.

Taking into account the properties of the problem of interest, the following approximations can finally be applied: $\mathcal{R}_1 = \big\{ y^i = z^k - \lambda_i e^i \,\big|\, i = 1,\dots,n \big\}$, where $e^i = (0,\dots,1,\dots,0)$ is the vector of the standard Euclidean basis; $\mathcal{R}_2 = \big\{ y^i = z^k - \lambda_i p^i \,\big|\, i = 1,\dots,n \big\}$, where $p^i = s^i + \dfrac{1}{2}\,S^{-1} q$ and s^i are the columns of matrix S. Such a definition of the vector p^i is due to the properties of the function $H(x)$. The next approximation has been constructed with the aid of numerical experiments: $\mathcal{R}_3 = \big\{ y^i = z^k - \lambda_i c^i \,\big|\, i = 1,\dots,n \big\}$, where $c_i = (\underbrace{-1,\dots,-1}_{i}, 1, 1, \dots, 1)$.

During our numerical testing, the approximations proposed have not shown a sufficient efficiency. Therefore, we suggested to apply the integrated approximation: $\mathcal{R} = \mathcal{R}_1 \bigcup \mathcal{R}_2 \bigcup \mathcal{R}_3$.

All the software programs were written with the use of Visual C++ 6.0 by Elena Petrova, one of our colleagues. The prescribed accuracy for solving the problem has been chosen equal to 0.001. We select the points as starting ones in such a way to provide for the worst conditions for the algorithm. The test systems with known solutions were taken from [26]:

1) $f_i = (3 - 2x_i)x_i + 1 - x_{i-1} - 2x_{i+1}, \quad i = 1,2,\dots,n, \quad x_0 = x_{n+1} = 0;$

2) $f_i = 3x_i - x_{i-1} - 2x_{i+1} - kx_i^2 + 1, \quad i = 1,2,\dots,n, \quad x_0 = x_{n+1} = 0;$

3) $f_1 = 1 - x_1, \quad f_i = 10(i-1)(x_i - x_{i-1})^2, \quad i = 2,3,\dots,n;$

4) Let n be an even number, then $f_i = 1 - x_i$, $i = 1,3,5,\dots,n-1$, $\quad f_i = 10(x_i - x_{i-1}^2), \quad i = 2,4,\dots,n;$

5) $f_i = 3x_i(x_{i+1} - 2x_i + x_{i-1}) + \dfrac{(x_{i+1} - x_{i-1})^2}{4}, \quad x_0 = 0; \quad x_{n+1} = 20;$

6) $f_i = x_i - 0,1x_{i+1}^2, \quad i = 1,2,\dots,n-1, \quad f_n = x_n - 0,1x_1^2.$

The results of numerical solution of these systems are given in Table 1 with the following notation: No is the number of an example; x_0 is a starting point; n is the system's dimension; F_0 is the initial value of the objective function $F(x)$; F_* is the resulting value of the goal function; St is the number of iterations of the global search algorithm; PL is the number of linearized problems solved; T is the algorithm's operating time (minutes:seconds.parts of seconds on AMD Athlon-1700, 256 Mb RAM).

Note that the global solution to the system in 6) was reached for all the starting points under consideration only with the aid of the special local search method (this is not presented in Table 1).

Table 1

No	x_0	n	F_0	F^*	St	PL	T
1	$(1,-1,\ldots,1,-1)$	10	57	0	2	29	00:00.14
		20	117	0	2	49	00:07.78
		30	177	0	2	247	01:21.22
		40	237	0	2	328	03:11.13
2	$(1,\ldots,1)$	10	12	0	6	91	00:00.43
		20	21	0	8	748	00:03.98
3	$(2,-2,\ldots,2,-2)$	10	7201	0	10	341	00:01.25
		20	30400	0	18	719	00:20.57
		30	69600	0	20	1403	07:12.66
		40	124800	0	22	1598	15:40.68
4	$(0,\ldots,0)$	10	5	0	42	5755	00:10.81
		20	10	0	61	9437	00:45.32
5	$(10,\ldots,10)$	10	600	0	8	673	00:01.34
		20	600	0	17	1715	02:17.35

In addition to the quadratic test systems, during the numerical experiments the following systems of nonlinear equations were solved:

7) $f_i = 2x_i - x_{i-1} - x_{i+1} + \dfrac{\nu^2}{2}\,(x_i + t_i + 1)^3, \quad i = 1,\ldots,n, \quad \nu = \dfrac{1}{n+1}\,, \quad t_i = i\nu, \quad x_0 = x_{n+1} = 0,$

8) $f_i = x_{i-1} - 2x_i + x_{i+1} - \nu^2 \exp x_i, \quad i = 1,\ldots,n, \quad \nu = \dfrac{1}{n+1}\,, \quad x_0 = x_{n+1} = 0,$

9) $f_i = x_i - \dfrac{1}{2n}\left(\displaystyle\sum_{j=1}^{n} x_j^3 + i\right), \quad i = 1,\ldots,n.$

As to the d.c. representation of these functions, it can be readily seen that $x^3 = g_1(x) - h_1(x)$, where $g_1(x) = \left(x + \dfrac{1}{4}\right)^4 + \dfrac{1}{256}$ and $h_1(x) = x^4 + \dfrac{x^2}{4} + \dfrac{1}{8}\left(x + \dfrac{1}{4}\right)^2$ are convex functions. In these cases, moreover, instead of solving the level problem and computing the value of $\eta(\zeta,\gamma)$ we compared the values of the goal function at the current and obtained critical points (see the end of Section 2). Other stages of the global search for the problems in 7) – 9) coincide with the stages for the problems in 1) – 6).

The results of computing these systems are given in Table 2.

Table 2

No	x_0	n	F_0	F^*	St	PL	T
7	$(1,-1,\ldots,1,-1)$	10	38.3	0.01229	2	167	$>$ 10 min
		20	78.2	0.07898	2	214	$>$ 10 min
8	$(1,\ldots,1)$	10	38.01	0	1	9	00:01.31
		20	78.1	0.04537	1	173	$>$ 10 min
9	$(2,-2,\ldots,2,-2)$	10	4.6	0	2	12	00:00.43
		20	9	0	3	21	00:02.64
		30	13.37	0	3	24	00:18.62
		40	17.75	0	2	28	00:20.84
		50	22.15	0	3	42	00:16.18
		60	26.5	0	3	47	00:53.78
		70	30.87	0	3	58	01:56.35

The solution time has been limited down to 10 minutes. It turned out that in some cases during this time (systems in 7) and 8)) we failed to find a solution with the desired accuracy. However, it is obvious that the value of the goal function has been decreased quite substantially. This fact allows one to use the obtained point as a good starting point for the standard numerical methods intended for solving systems of equations.

It is clear that to the end of improving the obtained results, in particular in order to increase the dimension of the systems under consideration, one needs some further improvement of the level surface approximation. For example, it can be readily seen that one needs to reduce the number of approximation points.

Nevertheless, one can conclude from the tables that for a reasonable solution time the algorithm is able to solve the systems of sufficiently large dimension. Thus, this fact confirms the possibility of solving the nonlinear systems of equations by the new global search algorithms.

4. Polyhedral separability problem. This section is devoted to one of the problems of the so-called general separability. As known [7], the practical demands are not limited by the linear separability only, and one often needs a more general concept of separability, e.g., the polyhedral separability [27, 28]. The practical applications concerning the cancer diagnosis and prognosis can be found, e.g., in [29].

4.1. Problem statement. First of all, recall the statement of the linear separability problem [7]. Consider two nonempty finite sets $\mathcal{A}$ and $\mathcal{B}$ from $\mathbb{R}^n$. These sets include M and N points, respectively, so that $\mathcal{A}$ and $\mathcal{B}$ can be represented by matrices $A \in \mathbb{R}^{M\times n}$ and $B \in \mathbb{R}^{N\times n}$. The columns of these matrices are the coordinates of points from $\mathcal{A}$ and $\mathcal{B}$, respectively. Let us denote these points by $a_i \in \mathcal{A}$, $i = 1, \dots, M$, and $b_j \in \mathcal{B}$, $j = 1, \dots, N$.

With this notation, we say that a hyperplane $H = \big\{x \in \mathbb{R}^n \big| \langle \omega, x\rangle = \xi\big\}$ strictly separates the sets $\mathcal{A}$ and $\mathcal{B}$ (see [7]) if

$$\min_{1\leqslant i\leqslant M} \langle \omega, a_i\rangle > \xi > \max_{1\leqslant j\leqslant N} \langle \omega, b_j\rangle. \tag{4.1}$$

Using the standard procedures, one can pass to the normalized form of linear separability (see [27, 28]):

$$\min_{1\leqslant i\leqslant M} \langle v, a_i\rangle \geqslant \mu + 1 > \mu - 1 \geqslant \max_{1\leqslant j\leqslant N} \langle v, b_j\rangle. \tag{4.2}$$

The next result follows from the separation theorems [7].

Proposition 1. *Sets $\mathcal{A}$ and $\mathcal{B}$ are linearly separable if and only if the intersection of their convex envelopes is empty:* $\operatorname{conv}(\mathcal{A}) \bigcap \operatorname{conv}(\mathcal{B}) = \emptyset$.

In the case when the sets are not linearly separable, one can introduce the notion of polyhedral separability. As a consequence, the problem of polyhedral separability consists in the separation of two finite sets of space $\mathbb{R}^n$ by a minimum number of hyperplanes.

Recall [27, 28] that sets $\mathcal{A}$ and $\mathcal{B}$ are said to be polyhedrally separable by the family of hyperplanes

$$H_p(\omega^p, \xi_p) = \big\{x \in \mathbb{R}^n \big| \langle \omega^p, x\rangle = \xi_p\big\}, \quad \omega^p \in \mathbb{R}^n, \quad \xi_p \in \mathbb{R}, \quad p = 1, \dots, P,$$

if for each point $a_i \in \mathcal{A}$ and for each hyperplane H_p the inequality $\langle a_i, \omega^p\rangle \leqslant \xi_p$ holds and for each point $b_j \in \mathcal{B}$ and at least for one hyperplane H_p the inequality $\langle b_j, \omega^p\rangle \geqslant \xi_p$ is fulfilled.

It can easily be seen that the sets $\mathcal{A}$ and $\mathcal{B}$ are involved into the above definition in an asymmetric manner. Moreover, this definition can be reformulated in a more formal way [27, 28].

Definition 1. Sets $\mathcal{A}$ and $\mathcal{B}$ are said to be *polyhedrally separable* by the family of hyperplanes

$$H_p = \big\{x \in \mathbb{R}^n \big| \langle v^p, x\rangle = \mu_p\big\}, \quad v^p \in \mathbb{R}^n, \quad \mu_p \in \mathbb{R}, \quad p = 1, \dots, P,$$

if the following inequalities hold: $\forall i = 1, \dots, M$, $\forall p = 1, \dots, P$, $\langle a_i, v^p\rangle \leqslant \mu_p - 1$; $\forall j = 1, \dots, N$, $\exists p \in \{1, \dots, P\}$, $\langle b_j, v^p\rangle \geqslant \mu_p + 1$.

As an analogue of Proposition 1, the following proposition is valid.

Proposition 2 [27, 28]. *Sets $\mathcal{A}$ and $\mathcal{B}$ are polyhedrally separable if and only if the convex hull of the set $\mathcal{A}$ has an empty intersection with the set $\mathcal{B}$:*

$$\operatorname{conv}(\mathcal{A}) \bigcap (\mathcal{B}) = \emptyset. \tag{4.3}$$

Recall also that the number

$$\Delta = \frac{1}{M}\sum_{i=1}^{M} \max\Big\{0, \max_{1\leqslant p\leqslant P} \big(\langle a_i, v^p\rangle - \mu_p + 1\big)\Big\} + \frac{1}{N}\sum_{i=1}^{N} \max\Big\{0; \min_{1\leqslant qp\leqslant P} \big(-\langle b_j, v^p\rangle + \mu_p + 1\big)\Big\} \tag{4.4}$$

is called the general classification error for the sets $\mathcal{A}$ and $\mathcal{B}$ with respect to the family of hyperplanes $\{H_p\}$.

Theorem 3. *Sets $\mathcal{A}$ and $\mathcal{B}$ are separable by the family of the hyperplanes $\{H_p\}$ if and only if*

$$\Delta = 0. \tag{4.5}$$

This theorem can be considered as a foundation of the polyhedral separability algorithm described below.

4.2. Analysis of the error function. Let V be a $(P \times n)$ matrix having the vectors $v^p,\ p = 1, \dots, P$, as its columns and $\mathcal{M} = (\mu_1, \dots, \mu_P)$ be a vector from $\mathbb{R}^P$. Define the error function depending on $P \times (n+1)$ variables as follows:

$$F(V, \mathcal{M}) = F_1(V, \mathcal{M}) + F_2(V, \mathcal{M}), \tag{4.6}$$

where

$$\begin{aligned} F_1(V, \mathcal{M}) &= \frac{1}{M} \sum_{i=1}^{M} \max\Big\{0, \max_{1 \leqslant p \leqslant P} \big(\langle a_i, v^p \rangle - \mu_p + 1\big)\Big\}, \\ F_2(V, \mathcal{M}) &= \frac{1}{N} \sum_{i=1}^{N} \max\Big\{0; \min_{1 \leqslant p \leqslant P} \big(-\langle b_j, v^p \rangle + \mu_p + 1\big)\Big\}. \end{aligned} \tag{4.7}$$

First of all we note that the error function is a d.c. function, since it is defined as a sum of max and min of the affine functions [9]. In order to apply the d.c. minimization theory described in the Section 2, we have to obtain a transparent d.c. representation of the error function to solve the problem of polyhedral separability

It can be shown that the function $F(V, \mathcal{M})$ can be represented as $F = g - f$, where g and f are the convex functions defined as follows:

$$g(V, \mathcal{M}) = \frac{1}{M} \sum_{i=1}^{M} \max\Big\{0, \max_{1 \leqslant p \leqslant P} \big(\langle a_i, v^p \rangle - \mu_p + 1\big)\Big\} + \frac{1}{N} \sum_{j=1}^{N} \max\Big\{0; \max_{1 \leqslant p \leqslant P} \big(\langle b_j, v^p \rangle - \mu_p - 1\big)\Big\};$$

$$f(V, \mathcal{M}) = \frac{1}{N} \sum_{j=1}^{N} \max_{1 \leqslant p \leqslant P} \big(\langle b_j, v^p \rangle - \mu_p - 1\big).$$

Thus, the problem of polyhedral separability can be formulated as a problem of unconstrained minimization of the nondifferentiable nonconvex error function $F(V, \mathcal{M})$:

$$F(V, \mathcal{M}) \downarrow \min, \quad (V, \mathcal{M}) \in \mathbb{R}^{P \times (n+1)}. \tag{4.8}$$

For solving this problem (as in the case of systems of d.c. equations), one can apply the theory of d.c. minimization outlined in Section 2 [9].

4.3. Global search. Now, let us describe the process of global search for problem (4.8), which includes the same stages a) – d) from Section 2 and reflects some peculiarities of the polyhedral separability problem.

First, the local search for problem (4.8) was performed by the special local search method described in Section 2. Suppose $\big(\widetilde{V}^k, \widetilde{\mathcal{M}}^k\big)$ is an obtained point. As in the case of solving the systems of d.c. equations, the linearized subproblems have been solved by Shor's r-algorithm [24]. Instead of solving only one linearized problem (stage b)) we have executed local search, i.e., a finite sequence of linearized problems has been solved. Let $\big(U^s, \Upsilon^s\big)$ be the point obtained by this procedure.

Next, instead of solving the level problem and computing the value of $\eta(\zeta, \gamma)$, one directly compares the value of the goal function $F\big(U^s, \Upsilon^s\big)$ with the value of $F\big(\widetilde{V}^k, \widetilde{\mathcal{M}}^k\big)$. As mentioned in Section 2, $\eta(\zeta, \gamma) > 0$ if and only if the point $\big(U^s, \Upsilon^s\big)$ is better than the vector $\big(\widetilde{V}^k, \widetilde{\mathcal{M}}^k\big)$.

As to the one-dimensional search, during the process of solving the polyhedral separability problem an approach similar to that used in the previous section was applied. We split the segment $[0, \gamma_+]$ into a finite number of subsegments. Let $\Delta\gamma$ be the length of each subsegment. The number $\Delta\gamma$ is selected from the set $\{\Delta\gamma_1, \dots, \Delta\gamma_I\}$.

Note that the polyhedral separability problem is rather difficult and that for constructing approximations of the level surface, as a whole, twelve variants of the approximation construction scheme based on different ideas have been proposed.

Further, since the global minimum of the error function is known: $F_* = 0$, we added two extra stopping criteria to the global search scheme.

Hence, the global search algorithm for problem (4.8) assumes the following form.

Suppose we have a starting point $(V, \mathcal{M})_0$, a local search accuracy τ_0, and the number γ_+.

Step 0. Set $k := 0$, $i := 0$, $\big(V^k, \mathcal{M}^k\big) := (V, \mathcal{M})_0$, and $\tau_k := \tau_0$.

Step 1. Select $\Delta\gamma_i \in \{\Delta\gamma_1, \dots, \Delta\gamma_I\}$.

Step 2. Starting with the point $\big(V^k, \mathcal{M}^k\big) \in \mathbb{R}^{P(n+1)}$, find a τ_k-critical point $\big(\widetilde{V}^k, \widetilde{\mathcal{M}}^k\big)$ for problem (4.8) by the special local search method. Set $\zeta_k := F\big(\widetilde{V}^k, \widetilde{\mathcal{M}}^k\big) \leqslant F\big(V^k, \mathcal{M}^k\big)$.

Step 3. If $F\big(\widetilde{V}^k, \widetilde{\mathcal{M}}^k\big) \leqslant \varepsilon_1$, then stop: $\big(\widetilde{V}^k, \widetilde{\mathcal{M}}^k\big)$ is an approximate global solution to problem (4.8).

Step 4. Select $\gamma \in [0, \gamma_+]$.

Step 5. Construct an approximation of the level surface

$$\mathcal{A}_k(\gamma) = \Big\{\big(V^s, \mathcal{M}^s\big) \mid f\big(V^s, \mathcal{M}^s\big) = \gamma - \zeta_k,\ s = 1, \dots, S_k\Big\}.$$

Step 6. For each $s \in \{1, \dots, S_k\}$, starting with the point $\big(V^s, \mathcal{M}^s\big)$, with the aid of the special local search method find a τ_k-critical point $\big(U^s, \Upsilon^s\big)$.

Step 7. If for some number $s \in \{1, \dots, S_k\}$ one has $F\big(U^s, \Upsilon^s\big) \leqslant \varepsilon_1$, then stop: $\big(U^s, \Upsilon^s\big)$ is an approximate global solution to problem (4.8).

Step 8. If for a number $s \in \{1, \dots, S_k\}$ one has $F\big(U^s, \Upsilon^s\big) < \zeta_k$, then set $k := k + 1$, $\tau_{k+1} = \tau_k/2$, $\big(V^{k+1}, \mathcal{M}^{k+1}\big) := \big(U^s, \Upsilon^s\big)$, and go to Step 2.

Step 9. If $F\big(U^s, \Upsilon^s\big) \geqslant \zeta_k$ for each $s \in \{1, \dots, S_k\}$, then set $\gamma := \gamma + \Delta\gamma_i \in [0, \gamma_+]$ and go to Step 5.

Step 10. If $F\big(U^s, \Upsilon^s\big) \geqslant \zeta_k$ for each $s \in \{1, \dots, S_k\}$ for all $\gamma \in [0, \gamma_+]$ and $\tau_k \leqslant \varepsilon_2$, then set $i := i + 1$, $\Delta\gamma_i := \Delta\gamma_{i+1}$ and go to Step 1.

Step 11. If $i = I$, i.e., the search for $\Delta\gamma$ is terminated, then stop.

4.4. Numerical experiments. For the numerical testing of the above-proposed algorithm, a set of 50 test examples has been constructed. In these examples the dimension of the original space varies from 2 up to 10, the number of points in the sets A and B changes from 1 up to 100, the number of separating hyperplanes balances between 1 and 5, and the general dimension of problem (4.8) varies from 3 up to 24. For each example a set of separating hyperplanes has been constructed, and for each test problem a global solution has been found with the corresponding value of the goal function: $F_* = 0$.

The accuracy of the stopping criterion has been chosen to be $\varepsilon_1 = 10^{-3}$ and $\varepsilon_2 = 10^{-4}$. The value of $\Delta\gamma_i$ ($I = 5$) has been selected from the set $\{0.5, 0.3, 0.2, 0.1, 0.05\}$. The starting point for all cases has been chosen as follows:

$$(V, \Gamma)_0 = \{v^p = b^p,\ \gamma_p = 0,\ p = 1, \dots, P\}. \tag{4.9}$$

The initial accuracy for the local search τ_0 was equal to 0.05.

The set of examples and the corresponding software implemented with the use of Visual C++ 6.0 have been developed by our colleague Oksana Druzhinina. The results of 12 approximations to the set of testing examples are given in Table 3.

In this table we use the following notation: $\mathcal{R}$ is the number of approximation; S is the number of elements in the approximation; Sol (%) is the percentage of solved examples; $\Delta_{\max}$ is the maximum deviation of obtained solutions from the global solution among all the examples; $PL_{\max}$ is the maximum number of solved linearized problems; No_{PLm} is the number of the example where $PL_{\max}$ is reached; $T_{\max}$ is the maximum operation time of the algorithm (Pentium Celeron, 660 MHz, 256 MB RAM); No_{Tm} is the number of the example where $T_{\max}$ was reached; K is the number of the examples in which an improvement of the goal function cannot be reached with the use of the level surface approximation elements.

Note that the approximations $\mathcal{R}11$ and $\mathcal{R}12$ represent a union of the approximations $\mathcal{R}3$, $\mathcal{R}4$, and $\mathcal{R}10$ taken in a different order.

It can be seen from Table 3 that the approximation $\mathcal{R}11$ turns out to be the best among the ones considered for solving the problem of polyhedral separability (94 % of examples were solved and the maximum deviation was 0.469). Moreover, the good results are shown by the approximations $\mathcal{R}12$ (92 % and 0.491) and $\mathcal{R}10$ (90 % and 1.97). The algorithm has given the maximum operation time and the maximum number of the solved linearized problems when applying approximation $\mathcal{R}5$. This may be explained by the fact that this approximation has the largest number of elements. From the viewpoint of problem difficulty, one can mark problems 26 and 30 (see the columns No_{PLm} and No_{Tm}).

Further, note that the approximation $\mathcal{R}10$ seems to be the most attractive for the problem of polyhedral separability. Although it slightly loses with respect to the approximations $\mathcal{R}11$ and $\mathcal{R}12$ as regards the percentage of problems solved, but it wins rather much as regards to the number of the linearized problems solved and the algorithm's operation time. To sum up, one can say that, despite all the difficulties of the nonconvex and nondifferentiable problem of polyhedral separability, the proposed algorithm has demonstrated very promising computational results.

5. Bilinear programming and bimatrix games. Various problems in engineering design, the decision making theory, operations researches, and economy can be described by bilinear programming problems. In this section we consider a bilinear programming problem and the related problem of finding the Nash equilibrium

Table 3

$\mathcal{R}$	S	$Sol\,\%$	$\Delta_{\max}$	$PL_{\max}$	No_{PLm}	$T_{\max}$	No_{tm}	K
1	n	68 %	2.0	13127	19	0 : 01 : 18.94	19	8
2	n	66 %	2.0	18005	26	0 : 01 : 30.79	26	4
3	N	82 %	2.0	36743	26	0 : 03 : 34.53	26	3
4	M	88 %	1.901	43576	26	0 : 22 : 30.86	26	0
5	$M*N$	82 %	2.0	328235	6	1 : 03 : 43.57	6	1
6	1	46 %	1.966	41627	19	0 : 08 : 38.18	30	6
7	n	74 %	1.98	6490	4	0 : 03 : 11.12	30	0
8	n	64 %	2.086	12749	26	0 : 06 : 54.16	30	0
9	n	74 %	2.0	9466	26	0 : 03 : 11.02	30	1
10	1	90 %	1.97	3701	26	0 : 01 : 29.17	24	1
11	$M+2$	94 %	0.469	47018	26	0 : 24 : 05.46	30	0
12	$M+2$	92 %	0.491	78022	26	0 : 31 : 28.14	26	0

point [20, 30, 31]. In spite of seeming simplicity, the bilinear problems turn out to be nonconvex. As mentioned above, in nonconvex problems there exist a lot of local extrema or even stationary (critical) points that can be very far from the global solutions even if the goal function values are concerned.

There are two types of bilinear problems: those with joint constraints and those with disjoint constraints. The former is a harder problem than the latter one. However, even for a problem with disjoint constraints, constructing a fast algorithm is a very complicated problem. Several methods have been proposed in the literature to solve the disjoint bilinear problems [3]. In this section we study the efficiency of our global search theory approach for disjoint bilinear problems.

5.1. Problem statement and the d.c. decomposition of a goal function. Consider the bilinear function

$$F(x,y) = \langle c, x\rangle + \langle x, Qy\rangle + \langle d, y\rangle, \tag{5.1}$$

where $c,\ x \in \mathbb{R}^m$; $d,\ y \in \mathbb{R}^n$; and Q is an $(m \times n)$ matrix.

The disjoint bilinear programming problem can be written down as follows:

$$\begin{aligned} &F(x,y) \uparrow \max_{(x,y)}, \\ &s.t.\quad x \in X \overset{\triangle}{=} \bigl\{x \in \mathbb{R}^m \mid Ax \leqslant a,\ x \geqslant 0\bigr\}, \quad y \in Y \overset{\triangle}{=} \bigl\{y \in \mathbb{R}^n \mid By \leqslant b,\ y \geqslant 0\bigr\}, \end{aligned} \tag{5.2}$$

where A is an $(m_1 \times m)$ matrix, B is an $(n_1 \times n)$ matrix, and $a \in \mathbb{R}^{m_1}$, $b \in \mathbb{R}^{n_1}$. Assume, that X and Y are bounded polytopes.

It can be readily seen that here the goal function can be represented as the difference of two convex functions:

$$F(x,y) = f(x,y) - g(x,y), \tag{5.3}$$

where $f(x,y) = \dfrac{1}{4}\,\|x + Qy\|^2$ and $g(x,y) = \dfrac{1}{4}\,\|x - Qy\|^2 - \langle d, y\rangle - \langle c, x\rangle$.

Therefore, for solving this problem we can apply the approach described in Section 2. However, it is necessary to note that, in contrast to the problems from the previous sections, the function $F(x,y)$ is differentiable and the objective is to maximize the goal function. This leads us to an appropriate modification of the global search algorithm.

5.2. Local search. First we note that if, for the local search in problem (5.2) we apply the method described in Section 2, the specific character of the problem, i.e., its bilinearity, will be lost (see [20] for details). Therefore, we propose two special local search methods for (5.2).

The first one is a modification of the local search algorithm for bilinear problems from [31]. Let (x_0, y_0) be a starting point.

The X-procedure

Step 0. Set $s := 0$ and $x^s := x_0$.
Step 1. Find a $\rho_s/2$-solution y^{s+1} to the linear problem

$$\langle d + x^s Q, y\rangle \uparrow \max_y, \quad y \in Y, \tag{LP_y}$$

so that the following inequality should hold:

$$\langle d + x^s Q, y^{s+1}\rangle + \rho_s/2 \geqslant \sup_y \bigl\{\langle d + x^s Q, y\rangle \bigm| y \in Y\bigr\}. \tag{5.4}$$

Step 2. Find a $\rho_s/2$-solution x^{s+1} to the linear problem

$$\langle c + Q y^{s+1}, x\rangle \uparrow \max_x, \quad x \in X, \tag{LP_x}$$

so that the following inequality should hold:

$$\langle c + Q y^{s+1}, x^{s+1}\rangle + \rho_s/2 \geqslant \sup_x \bigl\{\langle c + Q y^{s+1}, x\rangle \bigm| x \in X\bigr\}. \tag{5.5}$$

Step 3. If

$$F(x^{s+1}, y^{s+1}) - F(x^s, y^{s+1}) \leqslant \tau, \tag{5.6}$$

where τ is a solution accuracy, then stop. Otherwise, set $s := s + 1$ and go to Step 1.

This algorithm implies the approximate solution of the linear programming problems with respect to x and y alternatively.

Now let us formulate the convergence theorem for this method.

Theorem 4. *Let $F(\cdot)$ be an upper-bounded function on $X \times Y$; suppose $\rho_s > 0$, $s = 1, 2, \ldots,$ and $\sum\limits_{s=1}^{\infty} \rho_s < +\infty$. Then the sequence of points (x^s, y^s) from the X-procedure converges to $(\widehat{x}, \widehat{y})$ in such a way that*

$$F(\widehat{x}, \widehat{y}) \geqslant F(\widehat{x}, y) \quad \forall y \in Y, \tag{5.7}$$

$$F(\widehat{x}, \widehat{y}) \geqslant F(x, \widehat{y}) \quad \forall x \in X. \tag{5.8}$$

The pair $(\widehat{x}, \widehat{y})$ is called the critical point for (5.2) if inequalities (5.7) and (5.8) hold. This point is a partial global solution to (5.2) with respect to x and y separately.

When the algorithm terminates (i.e., inequality (5.6) holds), we obtain an approximate critical point for (5.2).

Further, let us emphasize some properties of the algorithm. First, the algorithm initially uses only the component x_0 from the pair (x_0, y_0). Second, the pair (x_0, y_0) may be unfeasible. Regardless the latter fact, the convergence of the X-procedure has been proved.

Below we propose another “symmetric” method for a local search in problem (5.2). In order to start this algorithm, only the component y_0 from the pair (x_0, y_0) is used.

The Y-procedure

Step 0. Set $s := 0$ and $y^s := y_0$.

Step 1. Find a $\rho_s/2$-solution x^{s+1} to the linear problem

$$\langle c + Qy^s, x\rangle \uparrow \max_x, \quad x \in X, \tag{LP_x}$$

so that the following inequality should hold:

$$\langle c + Qy^s, x^{s+1}\rangle + \rho_s/2 \geqslant \sup_x \big\{ \langle c + Qy^s, x\rangle \;\big|\; x \in X \big\}. \tag{5.9}$$

Step 2. Find a $\rho_s/2$-solution x^{s+1} to the linear problem

$$\langle d + x^{s+1}Q, y\rangle \uparrow \max_y, \quad y \in Y; \tag{LP_y}$$

this means that the following inequality should hold:

$$\langle d + x^{s+1}Q, y^{s+1}\rangle + \rho_s/2 \geqslant \sup_y \big\{ \langle d + x^{s+1}Q, y\rangle \;\big|\; y \in Y \big\}. \tag{5.10}$$

Step 3. If

$$F(x^{s+1}, y^{s+1}) - F(x^{s+1}, y^s) \leqslant \tau, \tag{5.11}$$

where τ is a solution accuracy, then stop. Otherwise, put $s := s + 1$ and go to Step 1.

Theorem 5. *Under the conditions of Theorem* 4, *the sequence* (x^s, y^s) *constructed by the Y-procedure converges to* $(\widehat{x}, \widehat{y})$, *which satisfies conditions* (5.7) *and* (5.8).

It should be noted that both the X- and Y-procedures converge to the point with the same properties; this point turns out to be critical for problem (5.2).

5.3. Global search algorithm in bilinear problems. Further, as in Sections 3 and 4, the global search algorithm for the bilinear problem is presented.

Suppose $(x^0, y^0) \in D \stackrel{\triangle}{=} X \times Y$ is a starting point, $\{\tau_k\}$, $\{\delta_k\}$ are two numbers sequences, τ_k, $\delta_k > 0$, $k = 0, 1, 2, \ldots$, $\tau_k \downarrow 0$, $\delta_k \downarrow 0$, $(k \to \infty)$.

In addition, let $Dir = \big\{ (u^1, v^1), \ldots, (u^N, v^N) \in \mathbb{R}^{m+n} \;\big|\; (u^s, v^s) \neq 0,\ s = 1, \ldots, N \big\}$ be a set of vectors, $\gamma_- \stackrel{\triangle}{=} \inf(g, D)$, $\gamma_+ \stackrel{\triangle}{=} \sup(g, D)$, and ν and q be some scalars.

Step 0. Let $k := 1$, $(\overline{x}^k, \overline{y}^k) := (x^0, y^0)$, $s := 1$, $p := 1$, $\gamma := \gamma_-$, and $\triangle\gamma = (\gamma_+ - \gamma_-)/q$.

Step 1. Starting with $(\overline{x}^k, \overline{y}^k) \in D$, by the X-procedure or the Y-procedure obtain a τ_k-critical point $(x^k, y^k) \in D$ for (5.2), here $F(x^k, y^k) \geqslant F(\overline{x}^k, \overline{y}^k)$. Set $\zeta_k := F(x^k, y^k)$.

Step 2. Compute the point $(\overline{u}^s, \overline{v}^s)$ with the aid of $(u^s, v^s) \in Dir$; here $(\overline{u}^s, \overline{v}^s) = \lambda_s(u^s, v^s)$ and $f(\overline{u}^s, \overline{v}^s) = \gamma + \zeta_k$.

Step 3. If $g(\overline{u}^s, \overline{v}^s) > \gamma + \nu\gamma$, then set $s := s + 1$ and go to Step 2 else go to Step 4.

Step 4. Starting with $(\overline{u}^s, \overline{v}^s)$, by the X-procedure or the Y-procedure obtain a δ_k-critical point for (5.2): $(\widehat{x}^s, \widehat{y}^s) \in D$.

Step 5. Find a point (x_0^s, y_0^s), $f(x_0^s, y_0^s) = \gamma + \zeta_k$, satisfying the following inequality:

$$\begin{aligned} &\big\langle \nabla_x f(x_0^s, y_0^s), \widehat{x}^s - x_0^s \big\rangle + \big\langle \nabla_y f(x_0^s, y_0^s), \widehat{y}^s - y_0^s \big\rangle + \delta_k \geqslant \\ &\qquad \geqslant \sup_{x,y} \Big\{ \big\langle \nabla_x f(x, y), \widehat{x}^s - x \big\rangle + \big\langle \nabla_y f(x, y), \widehat{y}^s - y \big\rangle : f(x, y) = \gamma + \zeta_k \Big\}. \end{aligned} \tag{5.12}$$

Step 6. Compute

$$\eta_k(\gamma) = \gamma - g(\widehat{x}^s, \widehat{y}^s) + \big\langle \nabla_x f(x_0^s, y_0^s), \widehat{x}^s - x_0^s \big\rangle + \big\langle \nabla_y f(x_0^s, y_0^s), \widehat{y}^s - y_0^s \big\rangle. \tag{5.13}$$

Step 7. If $\eta_k(\gamma) \leqslant 0$, $s < N$, then set $s := s + 1$ and go to Step 2.

Step 8. If $\eta_k(\gamma) \leqslant 0$, $s = N$, then set $\gamma := \gamma + \triangle\gamma$, $s := p$ and go to Step 2.

Step 9. If $\eta_k(\gamma) > 0$, then set $(\overline{x}^{k+1}, \overline{y}^{k+1}) := (\widehat{x}^s, \widehat{y}^s)$, $k := k + 1$, $s := s + 1$, $p := s$ and go to Step 1.

Step 10. If $s = N$, $\eta_k(\gamma) \leqslant 0$ $\forall\gamma \in [\gamma_-, \gamma_+]$ then stop.

Let us explain some specific properties of the above algorithm.

At Step 2 we construct the level surface approximation of $f(\cdot)$ with the aid of the set of vectors from Dir. At Step 3 we choose the points of level surface approximation satisfying the inequality from global optimality conditions (see [9]). The scalar ν is the first parameter of the algorithm by which one can change the precision of the inequality at Step 3. At Step 4 we implement an additional local search instead of solving one linearized problem.

At Step 6 we compute the quality estimate of the algorithm's iteration. At Steps 7 – 10 we check the stopping criterion and perform loops in the internal (generated by the points of the level surface approximation and the partition of the segment $[\gamma_-, \gamma_+]$) and the external (generated by the critical points) cycles. The scalar q is the second parameter of the algorithm by which one can change the number of the above-mentioned segment partition.

The above algorithm was tested for a special disjoint bilinear problem. This problem is equivalent to the problem of finding the Nash equilibrium point for bimatrix games [20]. The computational simulations are described in the next subsections.

5.4. Bimatrix games. Recall the definition of the Nash equilibrium point in a bimatrix game $\Gamma(A, B)$.

Definition 2. A point $(x^*, y^*) \in S_m \times S_n$ is called the Nash equilibrium point for the bimatrix game $\Gamma(A, B)$ if the following inequalities hold:

$$\begin{aligned} &\langle x^*, Ay^* \rangle \geqslant \langle x, Ay^* \rangle, \quad \forall x \in S_m \triangleq \Big\{ x \mid x_i \geqslant 0, \sum_{i=1}^{m} x_i = 1 \Big\}, \\ &\langle x^*, By^* \rangle \geqslant \langle x^*, By \rangle, \quad \forall y \in S_n \triangleq \Big\{ y \mid y_j \geqslant 0, \sum_{j=1}^{n} y_j = 1 \Big\}. \end{aligned} \tag{5.14}$$

The set of all such points is denoted by $NE(\Gamma)$.

Theorem 6 [30, 31]. *A point (x^*, y^*) is the Nash equilibrium point for a bimatrix game $\Gamma(A, B)$ if and only if it is a part of the global solution $(x^*, y^*, \alpha_*, \beta_*) \in \mathbb{R}^{m+n+2}$ of the following mathematical programming problem:*

$$F(x, y, \alpha, \beta) \overset{\triangle}{=} \langle x, (A + B)y \rangle - \alpha - \beta \uparrow \max, \quad (x, \beta, y, \alpha) \in X \times Y, \tag{5.15}$$

where $X = \big\{ (x, \beta) \in S_m \times \mathbb{R} \bigm| x^T B \leqslant \beta e_n \big\}$, $Y = \big\{ (y, \alpha) \in S_n \times \mathbb{R} \bigm| Ay \leqslant \alpha e_m \big\}$, $e_p = (1, 1, \dots, 1)^T \in \mathbb{R}^p$, and $p = m, n$.

At the same time, the numbers α_* and β_* are the profits of the first and the second players in the game $\Gamma(A, B)$, respectively: $\langle x^*, Ay^* \rangle = \alpha_*$ and $\langle x^*, By^* \rangle = \beta_*$; the optimum value of the goal function of problem (5.15) *is zero:*

$$F(x^*, y^*, \alpha_*, \beta_*) = 0. \tag{5.16}$$

Note that problem (5.15) is a bilinear one. Therefore, for finding an approximate Nash equilibrium point one can apply the global search algorithm proposed in Subsection 5.3.

In addition, we have to add two extra stopping criteria to the scheme after implementation of local search at Steps 1 and 4 in the same way as was made during the process of solving the polyhedral separation problem.

5.5. Numerical experiments. In Table 4 one can see the results of solution for randomly generated series of bimatrix games. The dimension of games varies from (5×5) up to (200×200).

First of all we would like to note that during the computational simulation we have selected the X-procedure to implement the local search. For solving linear problems by the X-procedure, a realization of the simplex method and its dual variant have been applied and the best results (with respect to the solution time) have been included in this table.

For constructing the level surface approximation at Step 2 of the algorithm and for the level problem at Step 5, analytical solutions can be given [20].

Constructing a level surface approximation was performed with the aid of the three direction sets. The first one is formed by the standard basis vectors: $Dir1 = \big\{ (e^i, e^j),\ i = 1, \dots, m,\ j = 1, \dots, n \big\}$.

To form the second direction set, the vectors e^i and e^j are combined with the vector (x, y), which is included in the current critical point (x, y, α, β) of problem (5.15) obtained by the local search procedure:

$$Dir2 = \big\{ (e^i + x, e^j + y),\ i = 1, \dots, m,\ j = 1, \dots, n \big\}.$$

The third direction set is constructed with the aid of elements of the matrices A and B and the all-one vectors e_p, $p = m, n$: $Dir3 = \big\{ (a^j + e_m, b^i + e_n),\ i = 1, \dots, m,\ j = 1, \dots, n \big\}$. Here $a^j \in \mathbb{R}^m$ are columns of A and $b^i \in \mathbb{R}^n$ are rows of B.

Table 4

$m = n$	Nm	UnS	$CntLoc$	Loc	Loc_A	T	T_A
5	100000	66	45012	1191424	11.9	5 : 52.87	0 : 00.00
10	10000	4	2272	189484	18.9	2 : 51.90	0 : 00.01
15	10000	0	1370	110628	11.1	6 : 40.21	0 : 00.04
20	10000	0	967	93764	9.4	15 : 25.98	0 : 00.09
25	1000	0	73	11655	11.7	3 : 09.65	0 : 00.18
30	1000	0	51	15875	15.9	6 : 43.68	0 : 00.40
35	1000	0	39	15838	15.8	11 : 45.26	0 : 00.70
40	100	0	0	2522	25.2	2 : 46.67	0 : 01.66
45	100	0	2	3016	30.2	5 : 09.06	0 : 03.09
50	100	0	0	3514	35.1	11 : 33.34	0 : 06.93
55	100	0	2	4710	47.1	20 : 19.67	0 : 12.19
60	50	0	1	3349	67.0	31 : 42.62	0 : 38.05
65	50	0	0	2376	47.5	29 : 34.12	0 : 35.48
70	20	0	0	1595	79.8	43 : 04.34	2 : 09.21
75	20	0	0	1156	57.8	27 : 01.78	1 : 21.08
80	10	0	0	397	39.7	11 : 23.42	1 : 08.34
85	10	0	0	794	79.4	31 : 37.43	3 : 09.74
90	5	0	0	457	91.4	36 : 15.92	7 : 15.18
95	5	0	0	526	105.2	32 : 10.01	6 : 26.00
100	5	0	0	1166	233.2	102 : 30.93	20 : 30.18
110	2	0	0	225	112.5	40 : 12.56	20 : 06.28
120	2	0	0	162	81.0	29 : 51.78	14 : 55.89
130	2	0	0	163	81.5	47 : 42.60	23 : 51.30
140	1	0	0	149	149.0	30 : 33.53	30 : 33.53
150	1	0	0	158	158.0	61 : 02.09	61 : 02.09
175	1	0	0	204	204.0	180 : 56.01	180 : 56.01
200	1	0	0	221	221.0	347 : 30.73	347 : 30.73

In Table we use the following notation: m and n are the numbers of the pure strategies of the players; Nm is the number of problems in the series; UnS is the number of unsolved problems in the series (when a given accuracy has not been reached); $CntLoc$ is the number of problems from the series solved only by the local search procedure at Step 1 of the algorithm; Loc is the number of the performed local searches; Loc_A is the average number of starts of the local search procedure for only one problem; T is the total working time of the software programs (minutes:seconds.parts of second on Pentium-4, 3.2 GHz, 1 Gb RAM); T_A is the average time needed to solve one problem.

It can be readily seen that almost all of the problems generated have been solved with the prescribed accuracy. The accuracy for the problems of dimension up to (100×100) was equal to 10^{-4}, whereas for the problems of dimension larger than (100×100) it was 10^{-3}. A number of problems have been solved only with the aid of the local search method in the case when the problem dimension was not so large.

Furthermore, the average time needed to solve one problem monotonically grows up to the dimension (60 × 60), then it decreases sometimes, but, on the whole, it keeps the tendency to increase. This fact can be explained by a lack of the necessary number of generated problems of large dimension.

The average number of starts of the local search procedure increases with the dimension and rises approximately from 10 for small problems up to 100 – 200 for the problems of large dimension.

As for finding an approximate Nash equilibrium point, one of the encouraging results of computational simulation is the solution of the example of dimension (200 × 200) within a reasonable time, whereas in the publications available the results of dimension only up to (96 × 96) can be found [32].

It should be noted that the algorithm spends much solution time on solving the linear problems, which is considered to be a satisfactory result. Now we hope to apply contemporary linear programming packages (e.g., CPLEX), which will allow us to increase the dimension of the games to be solved.

6. Conclusion. In this paper we have demonstrated the efficiency of the global search theory combined with the variational approach with respect to the problems from various mathematical fields. We proposed the global search algorithms for systems of nonlinear equations, problems of polyhedral separability, and a special problem of bilinear programming related to the bimatrix games.

All algorithms have been tested on the basis of a large number of numerical experiments. The results of our computational simulation confirm the efficiency of the algorithms developed and, being integrated with the previous results, certify some attractive advantages of the approach based on the global optimality conditions discussed in [9 – 20].

СПИСОК ЛИТЕРАТУРЫ

1. *Horst R., Tuy H.* Global optimization. Deterministic approaches. Berlin: Springer-Verlag, 1993.
2. *Horst R., Pardalos P.M., and Thoai V.* Introduction to global optimization. Dordrecht: Kluwer Academic Publishers, 1995.
3. *Floudas C.A., Visweswaran V.* Quadratic optimization // Handbook of Global Optimization / Ed. by Horst R., Pardalos P. Dordrecht: Kluwer Academic Publishers, 1995. 224–228.
4. *Tuy H.* D.C. Optimization: theory, methods and algorithms // Handbook of Global Optimization / Ed. by Horst R., Pardalos P. Dordrecht: Kluwer Academic Publishers, 1995. 149–216.
5. *Hiriart-Urruty J.-B.* Generalized differentiability, duality and optimization for problems dealing with difference of convex functions // Lecture Notes in Economics and Mathematical Systems. Vol. 256. Berlin: Springer-Verlag, 1985. 37–69.
6. *Hiriart-Urruty J.-B., Lemarchal C.* Convex analysis and minimization algorithms. Vols. I and II. Berlin: Springer-Verlag, 1993.
7. *Vasil'ev F.P.* Optimization methods. Moscow: Factorial-Press, 2002 (in Russian).
8. *Bazaraa M.S., Shetty C.M.* Nonlinear programming. Theory and algorithms. New York: Wiley, 1979.
9. *Strekalovsky A.S.* Elements of nonconvex optimization. Novosibirsk: Nauka, 2003 (in Russian).
10. *Strekalovsky A.S.* The search for a global maximum of a convex functional on an admissible set // Computational Mathematics and Mathematical Physics. 1993. **33**, N 3. 315–328.
11. *Strekalovsky A.S.* Extremum problems on complements of convex sets // Cybernetics and System Analysis. 1994. N 1. 88–100.
12. *Strekalovsky A.S.* Global optimality conditions for nonconvex optimization // J. of Global Optimization. 1998. **12**, N 4. 415–434.
13. *Strekalovsky A.S., Tsevendorj I.* Testing the $\mathfrak{R}$-strategy for a reverse convex problem // J. of Global Optimization. 1998. **13**, N 1. 61–74.
14. *Strekalovsky A.S., Kuznetsova A.A., and Tsevendorj I.* An approach to the solution of integer optimization problems // Computational Mathematics and Mathematical Physics. 1999. **39**, N 1. 9–16.
15. *Strekalovsky A.S.* One way to construct a global search algorithm for d.c. minimization problems // Nonlinear Optimization and Related Topics. Vol. 36 / Ed. by Pillo G.Di., Giannessi F. Dordrecht: Kluwer Academic Publishers, 2000. 429–443.
16. *Kuznetsova A.A., Strekalovsky A.S.* On solving the maximum clique problem // J. of Global Optimization. 2001. **21**, N 3. 265–288.
17. *Strekalovsky A.S.* On the minimization of the difference of convex functions on a feasible set // Computational Mathematics and Mathematical Physics. 2003. **43**, N 3. 380–390.
18. *Strekalovsky A.S.* Some remarks on d.c. programming // Optimization and Optimal Control / Ed. by Pardalos P.M., Tseveendorj I., and Enkhbat R. New Jersey–London–Singapore–Hong Kong: World Scientific Publishing Co. 2003. **1**. 347–367.
19. *Strekalovsky A.S., Yakovleva T.V.* On a local and global search involved in nonconvex optimization problems // Automation and Remote Control. 2004. **65**, N 3. 375–387.

20. *Orlov A.V., Strekalovsky A.S.* Numerical search for equilibria in bimatrix games // Computational Mathematics and Mathematical Physics. 2005. **45**, N 6. 947–960.
21. *Faddeev D.K., Faddeeva V.N.* Computational methods of linear algebra. San Francisco: W. H. Freeman and Co., 1963.
22. *Ortega J.M., Reinbolt W.C.* Iterative solution of nonlinear equations in several variable. San Diego: Academic Press, 1970.
23. *Demyanov V.F., Vasiliev L.V.* Nondifferentiable optimization. New York: Springer-Optimization Software, 1985.
24. *Shor N.Z.* Minimization methods of nondifferentiable functions and their applications. Kiev: Naukova Dumka, 1979 (in Russian).
25. *Mikhalevich V.S., Trubin V.A., and Shor N.Z.* Optimization problems of production-transportation planning. Moscow: Nauka, 1986 (in Russian).
26. *Roose A., Kulla V., Lomp M., and Meresoo T.* Test examples of systems of nonlinear equations. Tallin: Estonian Software and Computer Service Co., 1990.
27. *Astorino A., Gaudioso M.* Polyhedral separability through successive LP // Journal of Optimization Theory and Applications. 2002. **112**, N 2. 265–293.
28. *Mangasarian O.L.* Linear and nonlinear separation of patterns by linear programming // Operations Research. 1965. **13**, N 3. 444–452.
29. *Mangasarian O.L., Street W.N., and Wolberg W.H.* Breast cancer diagnosis and prognosis via linear programing // Operations Research. 1995. **43**, N 4. 570–577.
30. *Mills H.* Equilibrium points in finite games // J. of the Society for Industrial and Applied Mathematics. 1960. **8**, N 2. 397–402.
31. *Mukhamediev B.M.* The solution of bilinear problems and finding the equilibrium situations in bimatrix games // Computational Mathematics and Mathematical Physics. 1978. **18**, N 1. 60–66.
32. *McKelvey R.D., McLennan A.* Computation of equilibria in finite games // Handbook of Computational Economics. Vol. 1 / Ed. by Amman H.M., Kendrick D.A., and Rust J. Amsterdam: Elseivier, 1996. 87–142.

Поступила в редакцию
06.04.2007