Серия «Математика»
2022. Т. 41. С. 40—56

Онлайн-доступ к журналу:
http://mathizv.isu.ru

И З В Е С Т И Я
*Иркутского
государственного
университета*

Research article

УДК 519.853.4
MSC 90C26
DOI https://doi.org/10.26516/1997-7670.2022.41.40

Hybrid Global Search Algorithm with Genetic Blocks for Solving Hexamatrix Games

Andrei V. Orlov[1]✉

[1] Matrosov Institute for System Dynamics and Control Theory SB RAS, Irkutsk, Russian Federation
✉ anor@icc.ru

Abstract. This work addresses the development of a hybrid approach to solving three-person polymatrix games (hexamatrix games). On the one hand, this approach is based on the reduction of the game to a nonconvex optimization problem and the Global Search Theory proposed by A.S. Strekalovsky for solving nonconvex optimization problems with (d.c.) functions representable as a difference of two convex functions. On the other hand, to increase the efficiency of one of the key stages of the global search — constructing an approximation of the level surface of a convex function that generates the basic nonconvexity in the problem under study — operators of genetic algorithms are used. The results of the first computational experiment are presented.

Keywords: polymatrix games of three players, hexamatrix games, Nash equilibrium, Global Search Theory, local search, level surface approximation, genetic algorithm

Acknowledgements: The research was funded by the Ministry of Education and Science of the Russian Federation within the framework of the project ”Theoretical foundations, methods and high-performance algorithms for continuous and discrete optimization to support interdisciplinary research” (No. of state registration: 121041300065-9, code FWEW-2021-0003).

For citation: Orlov A. V. Hybrid Global Search Algorithm with Genetic Blocks for Solving Hexamatrix Games. *The Bulletin of Irkutsk State University. Series Mathematics*, 2022, vol. 41, pp. 40–56.
https://doi.org/10.26516/1997-7670.2022.41.40

Научная статья

Гибридный алгоритм глобального поиска с генетическими блоками для решения гексаматричных игр

А. В. Орлов[1✉]

[1] Институт динамики систем и теории управления им. В. М. Матросова СО РАН, Иркутск, Российская Федерация
✉ anor@icc.ru

Аннотация. Статья посвящена разработке гибридного подхода к решению полиматричных игр трех лиц (гексаматричных игр). С одной стороны, этот подход базируется на редукции игры к задаче невыпуклой оптимизации и Теории глобального поиска, созданной А. С. Стрекаловским для решения невыпуклых оптимизационных задач с (d.c.) функциями, представимыми в виде разности двух выпуклых функций. С другой стороны, для повышения эффективности одного из ключевых этапов глобального поиска — конструирования аппроксимации поверхности уровня выпуклой функции, задающей базовую невыпуклость в исследуемой задаче — используются операторы генетического алгоритма. Приведены результаты первого вычислительного эксперимента.

Ключевые слова: полиматричные игры трех лиц, гексаматричные игры, равновесие Нэша, теория глобального поиска, локальный поиск, аппроксимация поверхности уровня, генетический алгоритм

Благодарности: Работа выполнена в рамках базового проекта фундаментальных исследований Минобрнауки РФ «Теоретические основы, методы и высокопроизводительные алгоритмы непрерывной и дискретной оптимизации для поддержки междисциплинарных научных исследований» (Номер гос. регистрации: 121041300065-9, код проекта FWEW-2021-0003).

Ссылка для цитирования: Orlov A. V. Hybrid Global Search Algorithm with Genetic Blocks for Solving Hexamatrix Games // Известия Иркутского государственного университета. Серия Математика. 2022. Т. 41. С. 40–56.
https://doi.org/10.26516/1997-7670.2022.41.40

1. Introduction

It is well known that the problem of a numerical finding of equilibria in the Game Theory [7; 18] is one of the urgent issues for the contemporary mathematical optimization theory and methods [19] (when an equilibrium problem can be transformed into the optimization problem).

A classical matrix game can be reduced to two dual linear programming (LP) problems [7; 18], so it has a Convex Structure, and there are no fundamental difficulties with its solution. The first extension of a matrix game is a bimatrix game that already represents Nonconvex (bilinear) Structure [7;15;18;24;28]. We can say the same about polymatrix games [1;7;23].

For example, the search for a Nash equilibrium in a polymatrix game of three players turns out to be equivalent to a special nonconvex optimization problem with a triple bilinear structure [12; 17; 23].

Our group developed the nonconvex optimization approach to the numerical finding of a Nash equilibrium in these games based on the abovementioned equivalence between the games in question and the special mathematical optimization problems with bilinear structures in the objective function [15; 23]. The latter are solved by the Global Search Theory (GST) developed by A.S. Strekalovsky for nonconvex problems with d.c. functions, representable as a difference of two convex functions [20–22].

In contrast to the commonly accepted global optimization methods such as branch-and-bound based techniques, approximation approaches, diagonal methods, etc. [6; 27], the GST employs a reduction of the nonconvex problem to a family of more straightforward problems (usually convex). The latter can be solved by classic convex optimization methods [3;9]. Also, the GST includes some other elements such as one-dimensional search, level surface approximation, and so on [20–22].

When solving nonconvex mathematical optimization problems using the Global Search Theory [20–22], one of the crucial steps is to construct an approximation of the level surface of the convex function that defines the basic nonconvexity in the problem under study (see also [24; 25]).

For example, in the course of solving the following problem of d.c. maximization:

$$\Phi(x) \overset{\triangle}{=} h(x) - g(x) \uparrow \max, \qquad x \in D, \tag{$\mathcal{DC}$}$$

where $g(\cdot)$, $h(\cdot)$ are convex functions, D is a convex set, one should build the finite approximation

$$\mathcal{A}(\zeta, \xi) = \left\{ v^1, ..., v^N \mid h\left(v^i\right) = \xi + \zeta,\ i = 1, \ldots, N \right\}, \tag{1.1}$$

where $\inf(g, D) \leq \xi \leq \sup(g, D)$ is fixed, $\zeta \overset{\triangle}{=} \Phi(z)$ is the value of the objective function of the problem $(\mathcal{DC})$ at the current stationary (critical) point z. The approximation must be representative enough to be able to define whether the current point z is a global solution. It means, in particular, that if we are not in a global solution then the approximation must allow us to "jump out" the critical point where we are. See [20–22] for more details.

Currently, there are no general methods for constructing representative approximations for the problem $(\mathcal{DC})$. When solving specific nonconvex problems, the building of approximations is based on the previous experience [10; 12; 13; 17; 20; 21; 24–26]. So the development of new approaches to constructing approximations is an up-to-date problem in nonconvex optimization.

As for the numerical efficiency of the GST approach for solving equilibrium problems, it turned out to be very effective for large-scale bimatrix games (up to 1000 strategies per player) [28]. However, for hexamatrix games, the results leave much to be desired [12]. The latest Global Search Algorithm (GSA) needs a lot of different techniques for building level surface approximations and shows satisfactory results only for games with sparse matrices [12].

Therefore, in this paper, for a nonconvex problem with bilinear structures, which arises when searching for Nash equilibrium points in a hexamatrix game, a new Hybrid Global Search Algorithm (HGSA) based on GST is developed. On the one hand, the building of approximations of the level surface in this algorithm uses operators of Genetic Algorithms (GAs): crossover, mutation [4; 8]. On the other hand, it includes blocks of GSA, in particular, specialized methods of local search using the bilinear structure of the problem in question [10; 12; 17; 24; 25] As known, local methods are the main "building blocks" for the GSA based on GST [20; 21; 24; 25]. Previously, such a combination of Genetic Algorithm's elements and a local search proved to be effective in solving the simplest bilevel problems [11]. Note, we will use the term "Genetic Algorithms" for simplicity since the key used blocks appeared there. At the same time, some ideas used in this work (such as the representation of individuals as real vectors, and deterministic selection) are closer to more general evolutionary algorithms [4].

The structure of the paper is the following. Section 2 deals with the main elements of the Global Search Theory, then in Section 3, the selected basic stages of Genetic Algorithms are outlined. Section 4 addresses the optimization formulation of a hexamatrix game and the corresponding Basic GSA for its solving developed earlier. In Section 5, the new Hybrid Global Search Algorithm for hexamatrix games is presented. Section 6 presents the first computational results. Section 7 contains concluding remarks.

2. Elements of the Global Search Theory

Let us briefly recall the main stages of the Global Search in the problem of d.c. maximization $(\mathcal{DC})$ [20; 21; 24]. The theoretical basis of the Global Search is the so-called Global Optimality Conditions (GOCs) which for the case of the problem $(\mathcal{DC})$ takes the following form.

Theorem 1. *[20; 21; 24] If a feasible point $z \in D$ is a global solution to the problem $(\mathcal{DC})$ $(z \in Sol(\mathcal{DC}))$, then*

$$\forall (y, \xi) \in I\!R^n \times I\!R: \; h(y) - \xi = \zeta \overset{\triangle}{=} \Phi(z), \tag{2.1}$$

$$g(y) \leq \xi \leq \sup(g, D), \tag{2.2}$$

$$g(x) - \xi \geq \langle \nabla h(y), x - y \rangle \quad \forall x \in D. \tag{2.3}$$

These GOCs possess the so-called algorithmic (constructive) property [20; 21; 24]. It means that if one was successful in finding the pair $(\bar{y}, \bar{\xi})$ from (2.1)–(2.2), and the point $\bar{x} \in D$ such that the inequality (2.3) is violated, then we obtain the point which is better than the current point z (even z is a critical or stationary point). This constructive property forms the basis for building Global Search Schemes (GSS). One of the variants of the GSS can be briefly presented in the following way.

Let there be given some approximate critical (stationary) point z^k in the problem $(\mathcal{DC})$ with the value of the objective function $\zeta_k \overset{\triangle}{=} \Phi(z^k)$, constructed using some local search method. Then, one has to perform the following chain of operations.

1) Choose the number $\xi \in [\xi_-, \xi_+]$, where $\xi_- \overset{\triangle}{=} \inf(g, D)$, $\xi_+ \overset{\triangle}{=} \sup(g, D)$ and construct some finite approximation $\mathcal{A}_k = \mathcal{A}(\zeta_k, \xi)$ (see (1.1)) of the level surface $\mathcal{U}(\zeta_k, \xi) = \{y \mid h(y) = \xi + \zeta_k\}$ of the convex function $h(\cdot)$, that generates the basic nonconvexity in the problem $(\mathcal{DC})$.

2) For each point of the approximation $\mathcal{A}_k$, verify the inequality $g(v^i) \leq \xi$, $i = 1, 2, ..., N$, following from the GOCs (see (2.2)).

3) Using the point v^i of the approximation $\mathcal{A}_k$, selected at the second stage, find an approximate solution u^i of the convex linearized problem:

$$g(x) - \langle \nabla h(v^i), x \rangle \downarrow \min_x, \quad x \in D. \qquad (\mathcal{PL}(v^i))$$

4) Proceeding from the points $u^i \in D$, find new critical points $\widehat{x}^i$, $i \in \{1, ..., N\}$, in the problem $(\mathcal{DC})$, by means of some local search method.

5) Compare the value of the objective function at each point $\widehat{x}^i$ with the value of the objective function at the current critical point z. If one of the points $\widehat{x}^i$ is better than the current one, the latter is updated.

Global Search Algorithms are built based on GSS stages 1)-5), and they use the features of the specific $(\mathcal{DC})$-type problem under study.

The critical stage of the scheme is 1), where one needs to construct an approximation of the level surface of a convex function h (see also [20;21;24]). The successful construction of the approximation allows to "jump" out of the stationary (critical) point obtained by a local search, which, as is well known, is one of the main goals of the global search.

As mentioned above, there are no general methods for constructing a representative approximation for the problem $(\mathcal{DC})$. At present, this problem is solving by using the previous experience of constructing the approximations and the data from the formulation of the problem in question [10; 12; 13; 17; 20; 21; 24–26]. For example, if the feasible set of the problem has a polyhedron structure, it is obligatory to use basic Euclidean vectors as one of the elements for building approximations in combination with a current critical point and so on [10; 12; 13; 17; 20; 24–26]. But only for a few simplest nonconvex problems (for example, the problem of maximizing the squared norm on a box), it has been possible to construct approximations that theoretically guarantee to obtain a global solution [20].

The main drawback of the existing approaches to the construction of approximations is that the latter are all static and do not change in the process of solving the problem. In this paper, the problem of approximations is proposed to be solved using elements of the Genetic Algorithm. In this case, at each iteration of the algorithm, the approximations change dynamically and consider the current information about the solution process.

3. Principal Stages of Genetic Algorithms

Let's remind the principal stages and operators of Genetic Algorithms [4; 8], for simplicity, presenting these stages for the most general optimization problem:

$$F(x) \uparrow \max, \qquad x \in S. \tag{$\mathcal{P}_0$}$$

There are a lot of variants of implementation genetic operators [4; 8]. The number of publications on this topic for various types of problems is vast (see, for example, the references in [8]). Here we present the most appropriate scheme for our purpose. First, in some way (for example, randomly), we need to generate a set of the feasible points in the problem $(\mathcal{P}_0)$. This set is called a population of individuals: $Pop = \{x^i \mid x^i \in S, \quad i = 1, ..., N\}$. Then one has to select a so-called *fitness function* that can help to evaluate each point x^i to give some measure for them. Most often, the objective function $F(\cdot)$ of the problem $(\mathcal{P}_0)$ is used for this purpose.

Second, we have to build in a certain way two new individuals (offspring) by two (randomly) selected points (parents) from the population: $x^{i_1}, x^{i_2} \in Pop \quad \Rightarrow \quad y^1 \in S, \; y^2 \in S$. This procedure is called a crossover. Existing types of crossovers are very diverse. The most popular types are single-point, two-point, and uniform crossover [8]. The principal difficulty here is the necessity to preserve the feasibility of the constructed offsprings in the original problem.

The next stage is a random mutation of some components in the constructed offsprings, carried out with a certain probability: $y^1 \Rightarrow w^1 \in S$, $y^2 \Rightarrow w^2 \in S$. It is also necessary to monitor the feasibility of the resulting individuals here [8].

After that, one needs to compare the two resulting individuals with respect to the fitness function. So, the worst individual from the population is updated: $w := \arg\max\{F(w^1), F(w^2)\}$, Let $j : F(x^j) \le F(x^i)$ $\forall x^i \in Pop$. If $F(w) > F(x^j)$, then $x^j := w$.

The process usually finishes when a certain predetermined number of iterations (generations) of the described procedure has been produced [8].

4. Basic Algorithm for Solving Hexamatrix Games

Recall the formulation of a hexamatrix game with mixed strategies [1; 7; 12; 17; 23]:

$$\left.\begin{array}{ll} F_1(x,y,z) \overset{\triangle}{=} \langle x, A_1 y + A_2 z\rangle \uparrow \max\limits_x, & x \in S_m, \\ F_2(x,y,z) \overset{\triangle}{=} \langle y, B_1 x + B_2 z\rangle \uparrow \max\limits_y, & y \in S_n, \\ F_3(x,y,z) \overset{\triangle}{=} \langle z, C_1 x + C_2 y\rangle \uparrow \max\limits_z, & z \in S_l, \end{array}\right\}$$

where $S_p = \{u = (u_1, \ldots, u_p)^T \in I\!R^p \mid u_i \geq 0, \ \sum\limits_{i=1}^{p} u_i = 1\}, \quad p = m, n, l.$

The goal is to find a Nash equlibrium (approximately) [7; 17; 23] in the game $\Gamma_3 = \Gamma(A, B, C)$ $(A = (A_1, A_2),\ B = (B_1, B_2),\ C = (C_1, C_2))$. As known, in such an equilibrium none of the players are profitable to change its optimal strategy. Due to Nash's Theorem [7; 23] there exists a Nash equilibrium in the game $\Gamma_3 = \Gamma(A, B, C)$ with mixed strategies.

Let us consider the following optimization problem $(\sigma \overset{\triangle}{=} (x, y, z, \alpha, \beta, \gamma))$:

$$\left.\begin{array}{c} \Phi(\sigma) \overset{\triangle}{=} \langle x, A_1 y + A_2 z\rangle + \langle y, B_1 x + B_2 z\rangle + \langle z, C_1 x + C_2 y\rangle - \alpha - \beta - \gamma \uparrow \max\limits_{\sigma}, \\ \sigma \in D \overset{\triangle}{=} \{(x, y, z, \alpha, \beta, \gamma) \in I\!R^{m+n+l+3} \mid x \in S_m,\ \ y \in S_n,\ \ z \in S_l, \\ A_1 y + A_2 z \leq \alpha e_m, \quad B_1 x + B_2 z \leq \beta e_n, \quad C_1 x + C_2 y \leq \gamma e_l\}, \end{array}\right\} \quad (\mathcal{P})$$

where α, β, γ are auxuliary scalar variables, $e_p = (1, 1, ..., 1) \in I\!R^p$, $p = m, n, l$.

Theorem 2. [23] *The point (x^*, y^*, z^*) is a Nash equilibrium point in the hexamatrix game $\Gamma(A, B, C) = \Gamma_3$ $((x^*, y^*, z^*) \in NE(\Gamma_3))$ if and only if it is a part of a global solution $\sigma^* \overset{\triangle}{=} (x^*, y^*, z^*, \alpha_*, \beta_*, \gamma_*) \in I\!R^{m+n+l+3}$ to the problem $(\mathcal{P})$. At the same time, the numbers α_*, β_*, and γ_* are the payoffs of the first, the second, and the third players, respectively, in the game Γ_3. In addition, an optimal value $\mathcal{V}(\mathcal{P})$ of the problem $(\mathcal{P})$ is equal to zero:*

$$\mathcal{V}(\mathcal{P}) = \Phi(x^*, y^*, z^*, \alpha_*, \beta_*, \gamma_*) = 0. \tag{4.1}$$

Corollary 1. [23] *Let (x^*, y^*, z^*) is a Nash equilibrium in the game $\Gamma(A, B, C)$ with the payoffs α_*, β_*, and γ_*. Then*

$$\alpha_* = \max_{1 \leq i \leq m} (A_1 y^* + A_2 z^*)_i, \beta_* = \max_{1 \leq j \leq n} (B_1 x^* + B_2 z^*)_j, \gamma_* = \max_{1 \leq t \leq l} (C_1 x^* + C_2 y^*)_t.$$

Theorem 2 allows us to find a Nash equilibrium in the game Γ_3 by solving the problem $(\mathcal{P})$. One can also prove that if an approximate solution to the problem $(\mathcal{P})$ is obtained, then we have an approximate Nash equilibrium

$(NE(\Gamma_3,\varepsilon))$ [23]. In order to find an approximate global solution to the nonconvex problem $(\mathcal{P})$, the approach based on GST was developed [12;17].

The first stage of this approach is the building an explicit d.c. representation for the objective function Φ:

$$\Phi(x,y,z,\alpha,\beta,\gamma) = h(x,y,z) - g(x,y,z,\alpha,\beta,\gamma), \quad \text{where} \tag{4.2}$$

$$h(x,y,z) = \frac{1}{4}\Big(\|x+A_1y\|^2 + \|x+A_2z\|^2 + \|B_1x+y\|^2 + \|y+B_2z\|^2 + \\ +\|C_1x+z\|^2 + \|C_2y+z\|^2\Big), \quad g(\sigma) = \frac{1}{4}\Big(\|x-A_1y\|^2 + \|x-A_2z\|^2 + \\ +\|B_1x-y\|^2 + \|y-B_2z\|^2 + \|C_1x-z\|^2 + \|C_2y-z\|^2\Big) + \alpha+\beta+\gamma. \tag{4.3}$$

It is easy to see that these functions are convex on (x,y,z) and σ, respectively. Thus, the problem $(\mathcal{P})$ is a d.c. maximization problem. Using this decomposition, we presented below the GSA for finding a Nash equilibrium in the game Γ_3 [12; 17], based on the corresponding GOCs (see Section 2).

According to Corollary 1, let us denote: $\alpha(y,z) \overset{\triangle}{=} \max\limits_i (A_1y + A_2z)_i$, $\beta(x,z) \overset{\triangle}{=} \max\limits_j (B_1x + B_2z)_j$, $\gamma(x,y) \overset{\triangle}{=} \max\limits_t (C_1x + C_2y)_t$.

Let there be given a starting point $\sigma^0 \overset{\triangle}{=} (x^0,y^0,z^0,\alpha_0,\beta_0,\gamma_0) \in D$, numerical sequences $\{\tau_k\},\{\delta_k\}$ $(\tau_k,\delta_k > 0,\ k=1,2,...;\ \tau_k \downarrow 0,\ \delta_k \downarrow 0\ (k\to\infty))$, a set $Dir = \{(\bar u^1,\bar v^1,\bar w^1),...,(\bar u^N,\bar v^N,\bar w^N) \in I\!R^{m+n+l} | (\bar u^r,\bar v^r,\bar w^r) \neq 0,$ $r=1,...,N\}$, the numbers $\xi_- \overset{\triangle}{=} \inf(g,D)$ and $\xi_+ \overset{\triangle}{=} \sup(g,D)$, parameters M and ν, and ε is a prescribed tolerance for the problem's solution.

Basic Global Search Algorithm

Step 0. Set $k:=1$, $\bar\sigma^k \overset{\triangle}{=} (\bar x^k,\bar y^k,\bar z^k,\bar\alpha_k,\bar\beta_k,\bar\gamma_k) := \sigma^0$, $r:=1$, $\xi := \xi_-$, $\triangle\xi = (\xi_+ - \xi_-)/M$.

Step 1. Start a local search method from the point $(\bar x^k,\bar y^k,\bar z^k,\bar\alpha_k,\bar\beta_k,\bar\gamma_k)$ and construct a τ_k-critical point $\sigma^k \overset{\triangle}{=} (x^k,y^k,z^k,\alpha_k,\beta_k,\gamma_k) \in D$ to the problem $(\mathcal{P})$. Set $\zeta_k := \Phi(\sigma^k)$.

Step 2. If $\zeta_k \geq -\varepsilon$, then stop; in this case, $(x^k,y^k,z^k) \in NE(\Gamma_3,\varepsilon)$.

Step 3. Using $(\bar u^r,\bar v^r,\bar w^r) \in Dir$, construct a point (u^r,v^r,w^r) of the approximation $\mathcal{A}_k = \{(u^1,v^1,w^1),...,(u^N,v^N,w^N) | h(u^r,v^r,w^r) = \xi + \zeta_k,$ $r=1,...,N\}$ of the level surface $\mathcal{U}(\zeta_k,\xi) = \{(x,y,z) \mid h(x,y,z) = \xi+\zeta_k\}$ of the function $h(x,y,z)$. $\alpha_r := \alpha(v^r,w^r)$, $\beta_r := \beta(u^r,w^r)$, $\gamma_r := \gamma(u^r,v^r)$.

Step 4. If

$$g(u^r,v^r,w^r,\alpha_r,\beta_r,\gamma_r) > \xi + \nu\xi, \tag{4.4}$$

$r < N$ and $\xi < \xi_+$, then set $r := r+1$ and go to Step 3.

Step 5. If the inequality (4.4) takes place, but $r = N$ and $\xi < \xi_+$, then set $r := 1$, $\xi := \xi + \Delta\xi$ and go to Step 3.

Step 6. If the inequality (4.4) holds, but $r = N$ and $\xi = \xi_+$, then stop; σ^k is the obtained solution to the problem $(\mathcal{P})$.

Step 7. Find a δ_k-solution $\bar{\sigma}^r \overset{\triangle}{=} (\bar{x}^r, \bar{y}^r, \bar{z}^r, \bar{\alpha}_r, \bar{\beta}_r, \bar{\gamma}_r)$ of the following linearized problem $(\mathcal{PL}_r) = (\mathcal{PL}(u^r, v^r, w^r))$:

$$g(\sigma) - \langle \nabla h(u^r, v^r, w^r), (x, y, z) \rangle \downarrow \min_{\sigma}, \quad \sigma \in D. \qquad (\mathcal{PL}_r)$$

Step 8. Proceeding from the point $\bar{\sigma}^r$, build a τ_k-critical point $\widehat{\sigma}^r := (\widehat{x}^r, \widehat{y}^r, \widehat{z}^r, \widehat{\alpha}_r, \widehat{\beta}_r, \widehat{\gamma}_r) \in D$ to the problem $(\mathcal{P})$.

Step 9. If $\Phi(\widehat{\sigma}^r) \geq -\varepsilon$, then stop; $(\widehat{x}^r, \widehat{y}^r, \widehat{z}^r) \in NE(\Gamma_3, \varepsilon)$.

Step 10. If $\Phi(\widehat{\sigma}^r) \leq \Phi(\sigma^k) + \varepsilon,\ r < N$, then set $r := r + 1$ and return to Step 2.

Step 11. If $\Phi(\widehat{\sigma}^r) \leq \Phi(\sigma^k) + \varepsilon,\ r = N$ and $\xi < \xi_+$, then set $\xi := \xi + \triangle\xi$, $r := 1$ and go to Step 2.

Step 12. If $\Phi(\widehat{\sigma}^r) > \Phi(\sigma^k) + \varepsilon$, then set $\xi := \xi_-$, $(\bar{x}^{k+1}, \bar{y}^{k+1}, \bar{z}^{k+1}, \bar{\alpha}_{k+1}, \bar{\beta}_{k+1}, \bar{\gamma}_{k+1}) := \widehat{\sigma}^r$, $k := k + 1$, $r := 1$ and return to Step 1.

Step 13. If $\Phi(\widehat{\sigma}^r) \leq \Phi(\sigma^k) + \varepsilon,\ r = N$ and $\xi = \xi_+$, then stop. The point σ^k is the obtained solution to the problem $(\mathcal{P})$. #

The GSA is not an algorithm in the usual sense because some of its steps are not specified. For example, we do not know how to construct a feasible starting point and the set Dir, how to compute the points from the level surface approximation by the given set Dir, how to implement a local search, how to solve the problem $(\mathcal{PL}_r)$, etc. We will consider these issues below.

First, note that a feasible starting point can be constructed by using the barycenters of standard simplexes:

$$x_i^0 = \frac{1}{m}, \ \ i = 1, ..., m; \qquad y_j^0 = \frac{1}{n}, \ \ j = 1, ..., n; \qquad z_t^0 = \frac{1}{l}, \ \ t = 1, ..., l \ ;$$

$$\alpha_0 = \alpha(y^0, z^0); \qquad \beta_0 = \beta(x^0, z^0); \qquad \gamma_0 = \gamma(x^0, y^0).$$

As for a local search (see Steps 1 and 8), it can be based on the consecutive solution of the LP problems derived from the problem $(\mathcal{P})$ [12; 17]:

$$\left.\begin{array}{c} f_1^{(v,w)}(x, \beta) \overset{\triangle}{=} \langle x, (A_1 + B_1^T)v + (A_2 + C_1^T)w \rangle - \beta \uparrow \max\limits_{(x,\beta)}, \\ (x, \beta) \in X(v, w, \bar{\gamma}) \overset{\triangle}{=} \{(x, \beta) \mid x \in S_m, \\ B_1 x - \beta e_n \leq -B_2 w, C_1 x \leq \bar{\gamma} e_l - C_2 v\}; \end{array}\right\} \qquad (\mathcal{LP}_x(v, w, \bar{\gamma}))$$

$$\left.\begin{array}{c} f_2^{(u,w)}(y, \gamma) \overset{\triangle}{=} \langle y, (B_1 + A_1^T)u + (B_2 + C_2^T)w \rangle - \gamma \uparrow \max\limits_{(y,\gamma)}, \\ (y, \gamma) \in Y(u, w, \bar{\alpha}) \overset{\triangle}{=} \{(y, \gamma) \mid y \in S_n, \\ A_1 y \leq \bar{\alpha} e_m - A_2 w, C_2 y - \gamma e_l \leq -C_1 u\}; \end{array}\right\} \qquad (\mathcal{LP}_y(u, w, \bar{\alpha}))$$

$$\left.\begin{array}{c} f_3^{(u,v)}(z,\alpha) \overset{\triangle}{=} \langle z, (C_1 + A_2^T)u + (C_2 + B_2^T)v\rangle - \alpha \uparrow \max\limits_{(z,\alpha)}, \\ (z,\alpha) \in Z(u,v,\bar\beta) \overset{\triangle}{=} \{(z,\alpha) \mid z \in S_l, \\ A_2 z - \alpha e_m \le -A_1 v, B_2 z \le \bar\beta e_n - B_1 u\}. \end{array}\right\} \qquad (\mathcal{LP}_z(u,v,\bar\beta))$$

where $(u, v, w, \bar\alpha, \bar\beta, \bar\gamma) \in D$ is a feasible point in the problem $(\mathcal{P})$. This type of the local search is efficient for problems with a bilinear structure [10; 12; 13; 15–17; 21; 24–26].

We implemented other stages and parameters of the above GSA according to our previous experience [10;13;15;21;24–26], except the key moment: the construction of a level surface approximation. This stage is realized at Step 3. For the problem $(\mathcal{P})$ the approximation $\mathcal{A}_k = \mathcal{A}(\zeta_k, \xi)$ is constructed with the help of special sets of directions [10;12;13;15;17;21;24–26]. The triples $(u^r, v^r, w^r) \in \mathcal{A}_k$ by the given set Dir, are produced analytically through solving the quadratic equation (see, [17] for more details).

The sets Dir are chosen experimentally and should, first of all, contain as much information as possible from the problem formulation and information obtained in the solution process. Usually, when solving problems with a bilinear structure on polyhedral sets, for a positive outcome of the global search, it was enough to use the Euclidean basis vectors, vectors of ones $(1, 1, ..., 1)$, rows and columns of matrices involved in the objective function, and the components of the current critical point [10; 13; 15; 21; 24–26]. However, the numerical solution of hexamatrix games turned out to be a much more difficult problem and required the additional use of various kinds of pairwise conjugate vectors, eigenvectors of matrices, and the implementation in spaces of different dimensions [12; 17].

Moreover, for some sets, which contain a lot of points when the dimension of the problems grows (in some cases, the number of the points is equal to mnl), we had to use special techniques for reducing the number of points in them (see [10; 12; 15; 24; 25]).

Nevertheless, the numerical results concerning randomly generated hexamatrix games leave much to be desired [12]. The principal disadvantage of the existing approach to constructing approximations is that they do not change from iteration to iteration of the algorithm and contain a sufficiently large number of points (even after the procedure of reducing). The latter fact vastly affects the efficiency of the Global Search Algorithm when the problem dimension increases.

The Hybrid Global Search Algorithm, described in the next section, does not have these shortcomings.

5. Hybrid Global Search Algorithm

First of all, let us describe the chosen variants of implementing genetic operators in the hybrid algorithm for solving the problem $(\mathcal{P})$. This is based on our previous experience of using them in the GSA for solving the simplest bilevel optimization problems [11].

As a population at each iteration of the global search we select not the set of the feasible points but the points from the current level surface approximation: $Pop_k = \{(u^r, v^r, w^r) \mid (u^r, v^r, w^r) \in \mathcal{A}_k, r = 1, ..., N\}$.

The fitness function, which evaluates the approximation points, has the name $PLoc(\cdot)$ and is constructed in the following way. To calculate this function, first, we need to solve the linearized problem $(\mathcal{PL}_r)$ (which are linearized at the r-th point of the approximation) and obtain a feasible point $\bar{\sigma}^r$ in the problem $(\mathcal{P})$ (see Step 7 of the GSA). Then the value of the function $PLoc(\cdot)$ is the value of the objective function $\Phi(\cdot)$ at the approximate critical point $\widehat{\sigma}^r$obtained by the local search beginning from the point $\bar{\sigma}^r$ (see Step 8): $PLoc(u^r, v^r, w^r) \overset{\triangle}{=} \Phi(\widehat{\sigma}^r)$. Let the denotation $ArgPLoc(\cdot)$ means that we obtain a critical point provided by the function $PLoc(\cdot)$. So $\widehat{\sigma}^r = ArgPLoc(u^r, v^r, w^r)$.

Note that the approximation points may be infeasible. Nevertheless, due to the properties of the original problem $(\mathcal{P})$, the linearized problem $(\mathcal{PL}_r)$ will have a solution in any case [17;23]. And the local search method produces feasible approximate critical points.

To carry out the crossover operator, the so-called uniform crossover [8] was implemented. First, choose two arbitrary indices $r_1, r_2 \in \{1, ...N\}$ $(r_1 \neq r_2)$ and set $q := Rand[0, 1]$, where "Rand" is some subroutine for generating pseudorandom numbers. Next, $\forall j = 1, ..., m + n + l$, if $q < 0.5$ then $(\check{u}, \check{v}, \check{w})_j := (u^{r_1}, v^{r_1}, w^{r_1})_j$, $(\widetilde{u}, \widetilde{v}, \widetilde{w})_j := (u^{r_2}, v^{r_2}, w^{r_2})_j$, else $(\widetilde{u}, \widetilde{v}, \widetilde{w})_j := (u^{r_1}, v^{r_1}, w^{r_1})_j$, $(\check{u}, \check{v}, \check{w})_j := (u^{r_2}, v^{r_2}, w^{r_2})_j$ (each component of one of the offsprings is a component of one of the parents with a probability of 1/2).

As for a mutation operator, we use the simple random procedure [8]. Let P_m be a probability of employing the mutation, K is a positive constant. Set $q_1 := Rand[0, 1]$. If $q_1 < P_m$ then $(\check{u}, \check{v}, \check{w})_j := Rand[0, K]$ $\forall j = 1, ..., m + n + l$. Set $q_2 := Rand[0, 1]$. If $q_2 < P_m$ then $(\widetilde{u}, \widetilde{v}, \widetilde{w})_j := Rand[0, K]$ $\forall j = 1, ..., m+n+l$.

Based on the principal elements of GAs, further, we present the Hybrid Global Search Algorithm, combining these elements and the steps of the Basic Global Search Algorithm from the previous section.

Let there be given a point $\sigma^0 \in D$, numerical sequences $\{\tau_k\}, \{\delta_k\}$ $(\tau_k > 0,\ \delta_k > 0,\ k = 1, 2, ...,\ \ \tau_k \downarrow 0,\ \delta_k \downarrow 0\ (k \to \infty))$, a set of directions $Dir = \{(\bar{u}^1, \bar{v}^1, \bar{w}^1), ..., (\bar{u}^N, \bar{v}^N, \bar{w}^N) \in I\!R^{m+n+l} | (\bar{u}^r, \bar{v}^r, \bar{w}^r) \neq 0,\ r = 1, ..., N\}$, numbers $\xi_- \overset{\triangle}{=} \inf(g, D)$ and $\xi_+ \overset{\triangle}{=} \sup(g, D)$, mutation probability P_m, the maximal number of generations G_{max}, and ε is a prescribed tolerance for the problem's solution.

Hybrid Global Search Algorithm (HGSA)

Step 0. Set $k := 1$, $\bar{\sigma}^k := \sigma^0$, $\xi := \xi_-$, $\triangle\xi = (\xi_+ - \xi_-)/N$.

Step 1. Beginning from the point $\bar{\sigma}^k$ by the local search method find a τ_k-critical point $\sigma^k = (x^k, y^k, z^k, \alpha_k, \beta_k, \gamma_k) \in D$ to the problem $(\mathcal{P})$. Set $\zeta_k := \Phi(\sigma^k)$.

Step 2. If $\zeta_k \geq -\varepsilon$, then stop; in this case, $(x^k, y^k, z^k) \in NE(\Gamma_3, \varepsilon)$.

Step 3. Using the points $(\bar{u}^r, \bar{v}^r, \bar{w}^r) \in Dir$ construct the points (u^r, v^r, w^r) of the approximation of the level surface of the function $h(\cdot)$, $r = 1, ..., N$, such that $h(u^r, v^r, w^r) = \xi + \triangle\xi \cdot (r-1) + \zeta_k$, i.e. construct an initial population of points Pop_k of the level surface approximation. For each population point, calculate the value of the fitness function: $\zeta_r := PLoc(u^r, v^r, w^r)$, $r = 1, ..., N$. Let $j : \zeta_j \leq \zeta_r \ \ \forall r = 1, ..., N$.

Step 4. If for some $\bar{r} \in \{1, ...N\}$ the inequality $\zeta_{\bar{r}} \geq -\varepsilon$ holds, then stop; the approximate ε-Nash equilibrium in the game Γ_3 is found. $(x^*, y^*, z^*, \alpha_*, \beta_*, \gamma_*) := ArgPLoc(u^{\bar{r}}, v^{\bar{r}}, w^{\bar{r}})$ is an approximate global solution to the problem $(\mathcal{P})$.

Step 5. $r_1 := Rand\{1, ..., N\}$, $r_2 := Rand\{1, ..., N\}$. Using the crossover procedure by the points $(u^{r_1}, v^{r_1}, w^{r_1})$ and $(u^{r_2}, v^{r_2}, w^{r_2})$ construct the points $(\widetilde{u}, \widetilde{v}, \widetilde{w})$ and $(\breve{u}, \breve{v}, \breve{w})$.

Step 6. Implement the mutation procedure with probability P_m for the points $(\widetilde{u}, \widetilde{v}, \widetilde{w})$ and $(\breve{u}, \breve{v}, \breve{w})$.

Step 7. Calculate $\widetilde{\alpha} := \alpha(\widetilde{v}, \widetilde{w})$, $\widetilde{\beta} := \beta(\widetilde{u}, \widetilde{w})$, $\widetilde{\gamma} := \gamma(\widetilde{u}, \widetilde{v})$; $\breve{\alpha} := \alpha(\breve{v}, \breve{w})$, $\check{\beta} := \beta(\breve{u}, \breve{w})$, $\breve{\gamma} := \breve{\gamma}(\widetilde{u}, \widetilde{v})$. Set $\widetilde{\sigma} := (\widetilde{u}, \widetilde{v}, \widetilde{w}, \widetilde{\alpha}, \widetilde{\beta}, \widetilde{\gamma})$, $\breve{\sigma} := (\breve{u}, \breve{v}, \breve{w}, \breve{\alpha}, \check{\beta}, \breve{\gamma})$. Compute $\widetilde{\zeta} := \Phi(\widetilde{\sigma})$, $\check{\zeta} := \Phi(\breve{\sigma})$; $\widetilde{\xi} := g(\widetilde{\sigma})$, $\check{\xi} := g(\breve{\sigma})$ and construct analytically two new points lying on the level surfaces: $h(\bar{u}^{r_1}, \bar{v}^{r_1}, \bar{w}^{r_1}) = \widetilde{\xi} + \widetilde{\zeta}$ and $h(\bar{u}^{r_2}, \bar{v}^{r_2}, \bar{w}^{r_2}) = \check{\xi} + \check{\zeta}$.

Step 8. Calculate $PLoc(\bar{u}^{r_1}, \bar{v}^{r_1}, \bar{w}^{r_1})$ and $PLoc(\bar{u}^{r_2}, \bar{v}^{r_2}, \bar{w}^{r_2})$.
Set $(\bar{u}^r, \bar{v}^r, \bar{w}^r) = \arg\max\{PLoc(\bar{u}^{r_1}, \bar{v}^{r_1}, \bar{w}^{r_1}), PLoc(\bar{u}^{r_2}, \bar{v}^{r_2}, \bar{w}^{r_2})\}$.

Step 9. If $PLoc(\bar{u}^r, \bar{v}^r, \bar{w}^r) \geq -\varepsilon$, then stop; the approximate ε-Nash equilibrium in the game Γ_3 is found.
$(x^*, y^*, z^*, \alpha_*, \beta_*, \gamma_*) := ArgPLoc(\bar{u}^r, \bar{v}^r, \bar{w}^r)$ is a solution to the problem $(\mathcal{P})$.

Step 10. If $PLoc(\bar{u}^r, \bar{v}^r, \bar{w}^r) > \zeta_j$, then update the point j of the population: $(u^j, v^j, w^j) := (\bar{u}^r, \bar{v}^r, \bar{w}^r)$.

Step 11. If $k < G_{max}$ then $k := k+1$ and go to Step 3, otherwise stop.
$(u^*, v^*, w^*) := \arg\max\{PLoc(u^r, v^r, w^r) | (u^r, v^r, w^r) \in Pop_k, \ r = 1, ..., N\}$,
$(x^*, y^*, z^*, \alpha_*, \beta_*, \gamma_*) := ArgPLoc(u^*, v^*, w^*)$ is the obtained solution of the problem. #

Note, the tolerances τ_k and δ_k are used in the HGSA inside the function $PLoc(\cdot)$, where we solve the linearized problem and implement the local search method. So, there is no mention of τ_k and δ_k at the steps of the algorithm.

Also, pay attention that the procedure for constructing points lying on the level surface at Step 7 of the algorithm can be implemented for any point in the Euclidian space (except for 0), so there are no problems with the points feasibility obtained as a result of applying operators of the Genetic Algorithm.

6. Numerical Experiment

In order to demonstrate the workability and efficiency of the presented hybrid algorithm, we use the following several hexamatrix games from existing publications.

"Test problem 1" has the dimension $(3 \times 3 \times 3)$ and is taken from [1], multiplied by 10 for convenience:

$$A_1 = \begin{pmatrix} 10 & 10 & -10 \\ 20 & -10 & -10 \\ 30 & -15 & -10 \end{pmatrix}; \; A_2 = \begin{pmatrix} 20 & 30 & 25 \\ -10 & 0 & 20 \\ -10 & 20 & 10 \end{pmatrix}; \; B_1 = \begin{pmatrix} -10 & 20 & 10 \\ 30 & 0 & 35 \\ 30 & 35 & 30 \end{pmatrix};$$

$$B_2 = \begin{pmatrix} -20 & 30 & 10 \\ 10 & -10 & -10 \\ 20 & 10 & -20 \end{pmatrix}; C_1 = \begin{pmatrix} -30 & -10 & 10 \\ 40 & 10 & 40 \\ 10 & 20 & 22 \end{pmatrix}; C_2 = \begin{pmatrix} 10 & 20 & -30 \\ 20 & 10 & 20 \\ 30 & 20 & 40 \end{pmatrix}.$$

"Test problem 2" has the dimension $(4 \times 3 \times 2)$ and is taken from [2]:

$$A_1 = \begin{pmatrix} 1 & 1 & 3 \\ 1 & 2 & 0 \\ 3 & 4 & 1 \\ 2 & 3 & 2 \end{pmatrix}; \quad A_2 = \begin{pmatrix} 1 & 1 \\ 4 & 1 \\ 3 & 2 \\ 1 & 2 \end{pmatrix}; \quad B_1 = \begin{pmatrix} 5 & 2 & 2 & 3 \\ 2 & 5 & 3 & 4 \\ 1 & 4 & 2 & 1 \end{pmatrix};$$

$$B_2 = \begin{pmatrix} 2 & 5 \\ 1 & 2 \\ 4 & 1 \end{pmatrix}; \quad C_1 = \begin{pmatrix} 1 & 1 & 2 & 1 \\ 2 & 1 & 2 & 1 \end{pmatrix}; \quad C_2 = \begin{pmatrix} 5 & 4 & 3 \\ 2 & 1 & 3 \end{pmatrix}.$$

It also turned out to be interesting to study "Test Problem 2" multiplied by 10 as "Test Problem 2a".

Finally, "Test problem 3" is taken from our recent work [14] where one economic conflict problem was modeled as a hexamatrix game. This test has the dimension $(11 \times 11 \times 11)$ and we do not present it here to save space in the paper. See [14] for the data of matrices.

For the numerical experiment we use a computer with an Intel Core i5-2400 CPU (3.1 GHz), 4 Gb RAM, and MATLAB 7.11.0.584 R2010b programming system. Auxiliary quadratic and linear problems are solved by the standard MATLAB subroutines "quadprog" and "linprog", respectively, with default settings.

In Table 1 you can see the results of solving four formulated games by the Basic GSA with the following parameters of the algorithm: $\tau = 10^{-6}$, $\varepsilon = 10^{-5}$; $\xi_- = \inf(g, D)$, $\xi_+ = \xi_- + 2000$, $\Delta\xi = 1000$; $\nu = 0.02$; $Dir = Dir1 = \{(e^i, e^j, e^t),\ i = 1, ..., m,\ j = 1, ..., n,\ t = 1, ..., l\}$. Here $No.$ is the number of the Test problem; Φ_0 and $\hat{\Phi}$ are the values of the objective function of the problem in the starting point and the first critical point, respectively; GIt is the number of iterations of the GSA; QP and LP means the number of solved auxiliary convex quadratic and linear programming problems inside the algorithm, respectively. Note that $QP = Loc$, where Loc is the number of starts Local Search Algorithm. T is the problem's solution time in seconds. The indicators QP and LP, as well as T, can serve as the measure of the algorithm's efficiency.

Table 1

The results of the Basic Global Search Algorithm

No.	Φ_0	$\hat{\Phi}$	St	$QP = Loc$	LP	T
1	-47.1111	-8.9465	3	8	60	0.52
2	-3.6667	-1.2315	2	7	49	0.42
2a	-36.6667	-12.3148	3	10	946	5.72
3	-19.5545	-16.6828	10	91	584	6.71

The results of the Hibrid GSA are presented in Table 2. We test this algorithm with the different number of individuals in the population: $N = CountPop = 2, 3, 4, 5, 7, 10$; and with the various probability of mutation: $P_m = 0.01, 0.02, 0.035, 0.05$. Other parameters are fixed: $\tau = 10^{-6}$, $\varepsilon = 10^{-5}$; $\xi_- = \inf(g, D), \xi_+ = \xi_- + 2000, \Delta\xi = 1000$; $K = 1$, $G_{max} = 250$. Initial Dir is built by N random vectors from $Dir1$. In all cases global solutions were found but in the table, you can see the best variant for each problem. The notation $CurrGen$ is the number of the generation where the solution was found. If $CurrGen = 0$ then we obtain the solution at the stage of building the initial population (see Step 3 of the HSGA).

Table 2

The results of the Hybrid GSA

No.	N	P_m	$CurrGen$	$QP = Loc$	LP	T
1	3	0.01	0	4	36	0.23
2	2	0.01	0	2	9	0.06
2a	2	0.01	9	20	84	0.67
3	4	0.05	2	9	54	0.71

Comparing the results of the Basic GSA and the HGSA, we can see that values QP, LP, and T for HGSA are several times smaller. This confirms the workability and efficiency of using HGSA for seeking Nash equilibria in hexamatrix games.

7. Concluding Remarks

In the present paper, we developed a new hybrid approach for finding Nash equilibria in hexamatrix games, which combines the use of the Global Search Theory [20–22] and elements of genetic algorithms [8].

We described the original Global Search Algorithm and showed how to incorporate the genetic operators into the GSA. As a result, we build the new Hybrid Global Search Algorithm that takes into account the properties of the problem in question.

The first computational experiment shows the workability and efficiency of the HGSA on several known hexamatrix games. Our further research will be devoted to organizing and realizing a broad numerical experiment concerning comparing both algorithms in the vast field of test hexamatrix games. Based on our previous computational experience of applying a similar hybrid approach to the simplest bilevel problems [11], we hope that the algorithm developed can be efficient for large dimension hexamatrix games and competitive with the up-to-date numerical results concerning the solving of finite games (see, e.g., [5]).

References

1. Audet C., Belhaiza S., Hansen P. Enumeration of all the extreme equilibria in game theory: bimatrix and polymatrix games. *J. Optim. Theory Appl.*, 2006, vol. 129, no. 3, pp. 349–372. https://doi.org/10.1007/s10957-006-9070-3
2. Belhaiza S. Computing Perfect Nash Equlibria for Polymatrix Games. *Les Cahiers du GERAD G-2012-24.* Montreal, GERAD,2012.
3. Bonnans J.-F., Gilbert J.C., Lemarechal C., Sagastizabal C.A. *Numerical optimization: theoretical and practical aspects.* Springer, Berlin-Heidelberg, 2006.
4. Eiben A.E., Smith J.E. *Introduction to Evolutionary Computing.* Springer, Berlin-Heidelberg, 2003.
5. Golshteyn E., Malkov U., Sokolov N. The Lemke-Howson Algorithm Solving Finite Non-Cooperative Three-Person Games in a Special Setting. *DEStech Trans. Comput. Sci. Eng. (optim), Supplementary volume*, 2019, pp. 265–272. https://doi.org/10.12783/dtcse/optim2018/27938
6. Horst R., Tuy H. *Global optimization. Deterministic approaches.* Berlin, Springer-Verlag, 1993.
7. Mazalov V. *Mathematical game theory and applications.* New York, John Wiley & Sons, 2014.
8. Michalewicz Z. *Genetic Algorithms + Data Structures = Evolution Programs.* New York, Springer-Verlag, 1994.

9. Nocedal J., Wright S.J. *Numerical optimization.* Springer-Verlag, New York-Berlin-Heidelberg, 2000.
10. Orlov A.V. Numerical solution of bilinear programming problems. *Comput. Math. Math. Phys.*, 2008, vol. 48, pp. 225–241. https://doi.org/10.1134/S0965542508020061
11. Orlov A.V. Hybrid genetic global search algorithm for seeking optimistic solutions in bilevel optimization problems. *Bulletin of Buryat State University. Mathematics. Informatics.*, 2013, vol. 9, pp. 25–32 (in Russian).
12. Orlov A.V. Finding the Nash equilibria in randomly generated hexamatrix games. *Proceedings of the 14th International Symposium on Operational Research (SOR'17), Slovenia, Bled, September 27–29, 2017*, Slovenian Society Informatika, Section for Operational Research, Ljubljana, 2017, pp. 507–512.
13. Orlov A.V. The global search theory approach to the bilevel pricing problem in telecommunication networks. *Kalyagin, V.A. et al. (Eds.) Computational Aspects and Applications in Large Scale Networks*, Springer, Cham, 2018, pp. 57–73. https://doi.org/10.1007/978-3-319-96247-4_5
14. Orlov A.V., Batbileg S. Oligopolistic banking sector of Mongolia and polymatrix games of three players. *The Bulletin of Irkutsk State University. Series Mathematics*, 2015, vol. 11, pp. 80–95. (in Russian)
15. Orlov A.V., Strekalovsky A.S. Numerical search for equilibria in bimatrix games. *Comput. Math. Math. Phys.*, 2005, vol. 45, pp. 947–960.
16. Orlov A.V., Strekalovsky A.S. On a Local Search for Hexamatrix Games. *CEUR Workshop Proceedings. DOOR-SUP 2016.*, 2016, vol. 1623, pp. 477–488.
17. Orlov A.V., Strekalovsky A.S., Batbileg S. On computational search for Nash equilibrium in hexamatrix games. *Optim. Lett.*, 2016, vol. 10, no. 2, pp. 369–381. https://doi.org/10.1007/s11590-014-0833-8
18. Owen G. *Game Theory.* San Diego, Academic Press, 1995.
19. Pang J.-S. Three modeling paradigms in mathematical programming. *Math. Program. Ser.B.*, 2010, vol. 125, no. 2, pp. 297–323. https://doi.org/10.1007/s10107-010-0395-1
20. Strekalovsky A.S. *Elements of nonconvex optimization.* Novosibirsk, Nauka Publ., 2003. (in Russian)
21. Strekalovsky A.S. On Solving Optimization Problems with Hidden Nonconvex Structures. *Rassias, T.M., Floudas, C.A., Butenko, S. (eds.) Optimization in Science and Engineering*, New-York, Springer, 2014, pp. 465–502. https://doi.org/10.1007/978-1-4939-0808-0_23
22. Strekalovsky A.S. On a Global Search in D.C. Optimization Problems. *Jacimovic, M., Khachay, M., Malkova, V., Posypkin, M. (eds.) Optimization and Applications. OPTIMA 2019. Communications in Computer and Information Science*, Springer, Cham, 2020, vol. 1145, pp. 222–236. https://doi.org/10.1007/978-3-030-38603-0_17
23. Strekalovsky A.S., Enkhbat R. Polymatrix games and optimization problems. *Autom. Remote Control*, 2014, vol. 75, no. 4, pp. 632–645. https://doi.org/10.1134/S0005117914040043
24. Strekalovsky A.S., Orlov A.V. *Bimatrix games and bilinear programming.* Moscow, Fizmatlit Publ., 2007. (in Russian)
25. Strekalovsky A.S., Orlov A.V. *Linear and quadratic-linear problems of bilevel optimization.* Novosibirsk, SB RAS Publ., 2019. (in Russian)
26. Strekalovsky A.S., Orlov A.V. Global Search for Bilevel Optimization with Quadratic Data. *In: Dempe, S., Zemkoho, A. (eds.) Bilevel optimization: advances and next challenges, Springer Optimization and Its Applications*, Springer, Cham, 2020, vol. 161, pp. 313–334. https://doi.org/10.1007/978-3-030-52119-6_11

27. Strongin R.G., Sergeyev Ya.D. *Global optimization with non-convex constraints. Sequential and parallel algorithms.* New York, Springer-Verlag, 2000.
28. Vasilyev I.L., Klimentova K.B., Orlov A.V. A parallel search of equilibrium points in bimatrix games. *Numer. Methods Prog.*, 2007, vol. 8, no. 3, pp. 233–243. Available at: https://en.num-meth.ru/index.php/journal/article/view/265. (accessed 2022, June 28). (in Russian)

Об авторах

Орлов Андрей Васильевич, канд. физ.-мат. наук, доц., Институт динамики систем и теории управления им. В. М. Матросова СО РАН, Российская Федерация, 664033, г. Иркутск, anor@icc.ru, https://orcid.org/0000-0003-1593-9347

About the authors

Andrey V. Orlov, Cand. Sci. (Phys.–Math.), Assoc. Prof., Matrosov Institute for System Dynamics and Control Theory SB RAS, Irkutsk, 664033, Russian Federation, anor@icc.ru, https://orcid.org/0000-0003-1593-9347

Поступила в редакцию / Received 29.06.2022
Поступила после рецензирования / Revised 04.08.2022
Принята к публикации / Accepted 11.08.2022