BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **24**, No 2
Sofia • 2024 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2024-0014

An Improved Parallel Biobjective Hybrid Real-Coded Genetic Algorithm with Clustering-Based Selection

Andranik S. Akopov[1,2,3]
[1]*Central Economics and Mathematics Institute of Russian Academy of Sciences, 47, Nachimovski Prosp., 117418 Moscow, Russian Federation*
[2]*MIREA – Russian Technological University, 78, Prospekt Vernadskogo, 119454 Moscow, Russian Federation*
[3]*Moscow Institute of Physics and Technology, 9, Institutsky lane, 141700 Dolgoprudny, Moscow Region, Russian Federation*
E-mail: andranik.s.akopov@ieee.org

***Abstract*:** *This work presents an improved parallel biobjective hybrid real-coded genetic algorithm* (*MORCGA-MOPSO-II*). *The approach is based on the combined use of the parallel Multi-Objective Real-Coded Genetic Algorithm* (*MORCGA*) *and the Multi-Objective Particle Swarm Optimization* (*MOPSO*). *At the same time, clustering-based selection techniques are used to form subpopulations of parent individuals. Using well-known clustering algorithms* (*e.g., k-Means, hierarchical clustering, c-means, and DBSCAN*) *in combination with the proposed clustering-based mutation* (*the CL-mutation*) *directed toward the obtained cluster centers allows for improving the quality of the Pareto fronts' approximations. The results of the MORCGA-MOPSO-II were compared with other well-known multi-objective evolutionary algorithms* (*e.g., SPEA2, NSGA-II, FCGA, MOSPO, etc.*). *Moreover, the MORCGA-MOPSO-II was integrated with the previously developed agent-based model of a goods exchange through the objective functions. As a result, the Pareto fronts have been obtained for the agent-based model of a goods exchange in different configurations of the initial distribution of agents.*

***Keywords*:** *Clustering algorithms, Multi-objective optimization, Real-coded genetic algorithms, Particle swarm optimization, Multiagent socioeconomic systems, Agent-based modeling.*

1. Introduction

The search for optimal solutions in large-scale Multi-Agent Systems (MASs) is a challenging computational problem. The values of objective functions in MASs are computed as a result of simulation modeling. Therefore, well-known gradient-based techniques cannot be applied in simulation-based optimization. Instead of this, derivative-free optimization methods such as evolutionary algorithms can be used to find optimal solutions, e.g., standard (binary-coded) Genetic Algorithms (GA) [1],

Real-Coded Genetic Algorithms (RCGAs) [2], Particle Swarm Optimization (PSO) algorithms [3], Parallel Hybrid Genetic Algorithms (RCGA-PSO) [4, 5], etc.

A line of research is devoted to evolutionary optimization algorithms (e.g., [6-9]). At the same time, as shown in works [4, 5], hybrid algorithms of the RCGA-PSO type have a higher time efficiency while maintaining the required level of accuracy in the obtained solutions.

There is a line of research devoted to developing effective clustering methods. In particular, such clustering techniques as the k-Means [10], the hierarchical clustering [11], the c-Means [12] and the DBSCAN [13] are well known and have widespread applications. There is possible the combined use of clustering and genetic algorithms as it is shown in [14]. This paper presents a modified parallel bi-objective hybrid real-coded genetic algorithm (MORCGA-MOPSO-II) in which clustering techniques are used to improve the selection procedure of the best potential decisions for usage as parent individuals within the evolutionary search.

This paper presents an approach in clustering techniques that can be used to form subpopulations of potential decisions (i.e., individuals) with subsequent mutations in the hybrid algorithms of the RCGA-PSO type. RCGAs first proposed in [2], use real-coded heuristic operators such as the Simulated-Binary Crossover [15], the Laplace Crossover (LX) [16], the Power Mutation [17], and similar approaches. The important advantage of both standard GAs and RCGAs is the possibility of parallelizing evolutionary searching procedures related to selection, crossover, and mutation, as well as providing the exchange of the best potential decisions generated by agent processes interacting through a global population.

The main purpose of this study is to develop the MORCGA-MOPSO-II to improve the performance of evolutionary search due to the clustering of potential decisions at the stage of selection of parent individuals and forming new offspring individuals. At the same time, the MORCGA-MOPSO-II is applied for seeking optimal solutions and approximating the Pareto fronts for the bi-objective optimization problem formulated for the previously developed stochastic agent-based model of goods exchange [4]. This model considers paired interactions between multiple agents (sellers and buyers) at random moments, provided that their mutual interests coincide (i.e., there is a supply and demand for a particular product). The objective functions of the studied multiagent socio-economic system are the average utility of future consumption for an ensemble of agents and the total number of agents with money that need to be maximized.

2. Parallel Biobjective Hybrid Real-Coded Genetic Algorithm

2.1. Problem statement

In general terms, the following objective simulation-based optimization problem is considered:

(1) $$\min F(\mathbf{x}) = \left(f_1(x),\ f_2(x) \right),$$

s.t.

$$\mathbf{x} = (x_1,\ x_2,\ ...,\ x_n)^{\mathrm{T}} \in \Omega,$$

where **x** is a decision variable vector with dimension n, $\Omega = \prod_{i=1}^{n}[a_i,\ b_i]$ is the feasible region of the search space ($i = 1,\ 2,\ ...,\ n$ is the index of decision variables), and are the *m*-ths objective functions computed in the result of the simulation modeling. A similar problem statement is suggested in works [4, 5, 8]. In particular, the parallel single-objective hybrid genetic algorithm RCGA-PSO [4] is used to optimize the characteristics of the environment and strategies for making individual decisions by agents involved in barter and monetary interaction. In [5] the first version of the MORCGA-MOPSO has been applied to seek optimal behavior strategies within the simulation model of a multi-agent socio-economic system. In [8], a parallel hybrid objective Real-Coded Genetic Algorithm (FCGA) is proposed that allows for the search for the best solutions within the developed Intelligent Transportation System (ITS). The main difference between the genetic algorithm presented here and others is the use of clustering-based selection techniques to form subpopulations of parent individuals, in combination with the directed (guided) mutation in the vicinity of cluster centers.

The stochastic model of goods exchange proposed in [4] considers a multi-agent trade system. At each moment in time, a barter or monetary transaction can take place between each pair of agents within the visibility range. Agents make individual decisions based on the formation of optimal states (readiness) to conclude trade transactions at each moment. These states are determined outside the model using a developed hybrid genetic algorithm (RCGA-PSO) [4]. The previous problem statement considered the average utility of future consumption as the main objective for an ensemble of agents. At the same time, the total number of agents with money at the end of the model could be considered as a second (alternative) objective to be maximized, along with the utility of future consumption. This objective reflects the level of monetary equality among agents and their ability to participate in non-barter transactions.

The average utility of future consumption for an ensemble of agents at the final moment of the model $t_{|T|},\ t_{|T|} \in T$, equals to

$$U = \frac{1}{|I|}\sum_{t_k=0}^{|T|}\sum_{i=1}^{|I|}\Delta_i(t_k). \quad (2)$$

Here,

- $T = \{t_0,\ t_1\ ...,\ t_{|T|}\}$ is the set of time moments (by days), $|T|$ is the total number of time moments; $t_{|T|} \in T$ are the initial and final moments of the model;
- $I = \{i_1,\ i_2,\ ...,\ i_{|I|}\}$ is the set of indices of agents, where $|I|$ is the total number of agents, $\tilde{i} \in I$ is the sellers' indices, $\hat{i} \in I$ is the buyers' indices;
- $\Delta_i(t_k),\ t_k \in T$, is the value of the utility function of the *i*-th agent, which is computed as a result of the trade interaction between the seller and the buyer considering the costs of the distance between the target and the purchased product.

The total number of agents with money at the final moment of the model $t_{|T|},\ t_{|T|} \in T,$ equals to

$$M = \sum_{i=1}^{|I|} \tilde{m}_i(t_{|T|}), \tag{3}$$

where

$$\tilde{m}_i(t_{|T|}) = \begin{cases} 1 \text{ if } s_i(t_{|T|}) > 0, \\ 0 \text{ if } s_i(t_{|T|}) = 0. \end{cases} \tag{4}$$

Here,

- $s_i(t_k),\ t_k \in T,$ is the value of the money savings of the *i*-th agent, which is defined as the amount of agent's money considering those received and spent within the trade interactions.

Therefore, the problem of finding optimal strategies for agents' behavior and improving environmental characteristics can be stated as follows.

Problem A*. *The need to maximize the average utility of future consumption for an ensemble of agents and the total number of agents with money over the sets of control parameters that determine the states of agents* $\{\mu_b,\ \sigma_b^2,\ \mu_m,\ \sigma_m^2\}$ *and environmental characteristics* $\{c,\ \alpha,\ r,\ h\}$:

$$\begin{cases} \max\limits_{\{\mu_b,\ \sigma_b^2,\ \mu_m,\ \sigma_m^2\},\ \{c,\ \alpha,\ r,\ h\}} U, \\ \max\limits_{\{\mu_b,\ \sigma_b^2,\ \mu_m,\ \sigma_m^2\},\ \{c,\ \alpha,\ r,\ h\}} M, \end{cases} \tag{5}$$

s.t.

$$\mu_b, \mu_m \in [-1,\ 1],\ \sigma_b^2, \sigma_m^2 \in (0,\ 1],\ c \in \{1,\ 2,\ 3,\ 4,\ 5,\ 6\},\ \alpha \in [0,\ 1],\ r \in [1,\ \bar{r}],$$
$$h \in [0,\ 1].$$

Here:

- $\sigma_b^2, \sigma_m^2 \in (0,\ 1]$ are the parameters of log-normal distributions that determine the readiness of agents to conclude barter and monetary transactions;
- $c \in \{1,\ 2,\ 3,\ 4,\ 5,\ 6\}$ is the configuration of the initial distribution of agents in a discrete space (e.g., "the uniform distribution", etc.);
- $\alpha \in [0,\ 1]$ is the coefficient of contractility of the product of the agent-seller with the interests of the agent-buyer, which determines the probability of a deal;
- $r \in [1,\ \bar{r}]$ is the radius of trading interaction;
- $h \in [0,\ 1]$ is the probability of moving agents in the discrete space.

2.2. Algorithm description

The developed algorithm (MORCGA-MOPSO-II) as the RCGA-PSO suggested earlier [4, 5] is based on the combined use of the RCGA and the PSO. At the same time, the clustering-based selection is used to form subpopulations of individuals for choosing the most adapted parent solutions. In particular, the k-Means, the

hierarchical clustering, the c-means, and the DBSCAN, along with a tournament selection [18] can be used to form clusters of potential parent solutions.

Each individual in the RCGA contains the set of decision variables' values united with the set of objective functions' values (particular solutions) calculated over these decisions and the fitness function which estimates the level of dominance of this solution to others. At the same time, clustering executes in a criteria space using computed values of objective functions belonging to individuals.

The k-Means Algorithm [10] provides a partition in the number of observations (i.e., all individuals of the RCGA) into K clusters (i.e., potential parent solutions of the RCGA) in which each observation belongs to the cluster with the nearest mean. That assumes solving the following optimization problem:

$$\arg\min_{C} \sum_{j=1}^{|C|} \sum_{x_i \in c_j} \left\| x_i - \eta_j \right\|^2 . \tag{6}$$

Here,

- $I = (i_1, i_2, ..., i_{|I|})$ is the set of individuals' indices, where $|I|$ is the total number of individuals;
- $X = (x_{i_1}, x_{i_2}, ..., x_{i_{|I|}})$ is the set of individuals with their particular solutions (i.e., points that are to be clustered);
- $C = \{c_1, c_2, ..., c_{|C|}\}$ is the set of clusters, where $|C|$ is the total number of clusters;
- η_j is the mean (i.e., the centroid) of points (individuals) in c_j, where $j = 1, 2, ..., |C|$ are indexes of clusters.

The Hierarchical Clustering Algorithm [11] is based on constructing a hierarchy of clusters. For this purpose, the iterative procedure of combining and splitting clusters based on a measure of dissimilarity between sets of observations is exercised (e.g., the Euclidean distance).

For instance, the following distance update formula is used within the median linkage clustering [11]:

$$d(A \cup B, G) = \sqrt{\frac{d(A, G)^2}{2} + \frac{d(B, G)^2}{2} - \frac{d(A, G)^2}{4}}. \tag{7}$$

Here,

- A, B are two clusters joined into a new cluster;
- G is any other cluster that does not contain points (i.e., individuals) belonging to A or B.

The c-Means Algorithm (also referred to as fuzzy clustering) [12] assumes data points can potentially belong to multiple clusters. Therefore, the membership matrix is used to fuzzy graduate such points (e.g., points on the edge of a cluster can belong to the cluster with a lesser degree than points located in the center of the cluster):

(8) $$W = \begin{pmatrix} w_{11} & \cdots & w_{1N} \\ \vdots & \ddots & \vdots \\ w_{c1} & \cdots & w_{cN} \end{pmatrix}.$$

Here,

- $w_{ij} \in \{0, 1\}$, $i \in I$, $j \in C$, is the degree to which element x_i belongs to the cluster c_j, where *N* is the total number of points joined into the *c*-th cluster.

To execute the fuzzy clustering algorithm, it is necessary to minimize the following objective function:

(9) $$\arg\min_{C} \sum_{i=1}^{|I|} \sum_{j=1}^{|C|} w_{ij}^{m} \left\| x_i - \tilde{\eta}_j \right\|^2,$$

where

- $m \in (1, \infty)$ is the fuzzifier (i.e., the parameter defines the degree of fuzziness, which m is commonly set to 2);
- $\tilde{\eta}_j$ is the centroid of points c_j that is computed with considering values of $w_{ij} \in \{0, 1\}$, $i \in I$, $j \in C$.

The DBSCAN is one of the most popular density-based clustering non-parametric algorithms. The algorithm has been proposed in [13]. As with other clustering techniques, it operates with a set of points in some space to be clustered, groping that a closely located together based on the estimation of their density. Those points that lie alone in low-density regions are marked as outliers and are not to be clustered.

The DBCSAN requires the following rule to be fulfilled for each $x_i \in X$, $i \in I$:

(10) $$\tilde{C}(x_i) = \{ y_i \in X : \tilde{d} \in (x_i, y_i) \le \varepsilon \},$$

where $\tilde{C}$ is a subset of individuals belonging to the set of located with a distance of no more than from x_i, $\tilde{d}$ is the Euclidian distance between x_i and y_i.

Thus, the k-Means, the hierarchical clustering, the c-Means, and the DBSAN are used in the MORCGA-MOPSO-II to provide the selection of the most adapted potential solutions. Moreover, to improve the quality of the Pareto fronts' approximations the directed (guided) mutation in the vicinity of cluster centers is proposed.

The MORCGA-MOPSO-II uses the concept of Pareto dominance within the evolutionary search. At each moment $t_k \in T$, $k = 1, 2, \ldots, K$, the archive of nondominated solutions is updated if a new (offspring) solution $\{ f_{\tilde{i}1}(\tilde{\mathbf{x}}_{\tilde{i}}),\ f_{\tilde{i}2}(\tilde{\mathbf{x}}_{\tilde{i}}) \}_{\tilde{i} \in \tilde{I}}$ is better than the existing one, i.e., if the following condition is fulfilled:

(11) $$\begin{aligned} &f_{i1}\left(\mathbf{x}_i(t_k)\right) \succ f_{\tilde{i}1}\left(\tilde{\mathbf{x}}_{\tilde{i}}(t_k)\right) \text{ and } f_{i2}\left(\mathbf{x}_i(t_k)\right) \succeq f_{\tilde{i}2}\left(\tilde{\mathbf{x}}_{\tilde{i}}(t_k)\right) \text{ or} \\ &f_{i2}\left(\mathbf{x}_i(t_k)\right) \succ f_{\tilde{i}2}\left(\tilde{\mathbf{x}}_{\tilde{i}}(t_k)\right) \text{ and } f_{i1}\left(\mathbf{x}_i(t_k)\right) \succeq f_{\tilde{i}1}\left(\tilde{\mathbf{x}}_{\tilde{i}}(t_k)\right) \text{ for } \forall\ i \in I, \end{aligned}$$

where:

- $\tilde{I} = \{\tilde{i}_1, \tilde{i}_2, ..., \tilde{i}_{|I|}\}$ are the indices of new individuals (e.g., offspring-individuals), where $|\tilde{I}|$ is the total number of new individuals;
- $\mathbf{x}_i(t_k)$, $\tilde{\mathbf{x}}_{\tilde{i}}(t_k)$ are the decision variable vectors of existing *i*-ths individuals $(i \in I)$ and the new $\tilde{i}$-th individual $(\tilde{i} \in \tilde{I})$ at moment $t_k \in T$, respectively, where n is the total number of decision variables;
- $\{f_{i1}(\mathbf{x}_i), f_{i2}(\mathbf{x}_i)\}$, $i \in I$, are the values of objective functions computed with the decision variable $\mathbf{x}_i$.

When using the MORCGA, the fitness function is calculated for each new individual in each evolutionary process with a following update to the archive of nondominated solutions if condition (10) is fulfilled for these. The fitness function is computed based on the sum of the Pareto strengths of all solutions dominated by the new $\tilde{i}$-th individual $\tilde{i} \in \tilde{I}$:

$$\tilde{f}_{\tilde{i}}\left(\tilde{\mathbf{x}}_{\tilde{i}}(t_k)\right) = \frac{\tilde{h}_{\tilde{i}}\left(\tilde{\mathbf{x}}_{\tilde{i}}(t_k)\right)}{1 + \tilde{w}_{\tilde{i}}\left(\tilde{\mathbf{x}}_{\tilde{i}}(t_k)\right) + \dfrac{1}{2 + \tilde{d}_{\tilde{i}}\left(\tilde{\mathbf{x}}_{\tilde{i}}(t_k)\right)}}, \tag{12}$$

where

- $\tilde{h}_{\tilde{i}}$ is the contribution of the $\tilde{i}$-th individual $\tilde{i} \in \tilde{I}$ to the Logarithmic HyperVolume (LHV) [19] computed using particular solutions;
- $\tilde{w}_{\tilde{i}}$ is the Pareto weakness of the $\tilde{i}$-th individual $\tilde{i} \in \tilde{I}$, computed at the t_k iteration $(t_k \in T)$, The value of $\tilde{w}_{\tilde{i}}$ is the sum of the Pareto strengths of all individuals that dominate the $\tilde{i}$-th individual $\tilde{i} \in \tilde{I}$;
- $\tilde{d}_{\tilde{i}}$ is the Euclidean Distance (ED) from the $\tilde{i}$-th individual to the nearest individual in the criteria space.

The values of decision variables $\tilde{\mathbf{x}}_{\tilde{i}}(t_k)$ of the new $\tilde{i}$-th individual (i.e., the offspring) $(\tilde{i} \in \tilde{I})$ are forming with well-known crossover (e.g., SBX, LX, etc., [15, 16]) and mutation operators (e.g., PM, SUM, etc., [6, 8, 17]). At the same time, it is suggested to use the following new clustering-based mutation operator (CL-mutation):

$$\tilde{\mathbf{x}}_{\tilde{i}}(t_k) = \begin{cases} \mathbf{L}_{\tilde{i}c}(t_k), \text{ if } \mathbf{L}_{\tilde{i}c}(t_k) \in [\underline{\mathbf{x}}, \overline{\mathbf{x}}], \\ \underline{\mathbf{x}} + u(0, 1)(\overline{\mathbf{x}} - \underline{\mathbf{x}}), \text{ if } \mathbf{L}_{\tilde{i}c}(t_k) \notin [\underline{\mathbf{x}}, \overline{\mathbf{x}}], \end{cases} \tag{13}$$

where

$$\mathbf{L}_{\tilde{i}c}(t_k) = \begin{cases} \text{Lambert } W\left(\ln \tilde{N}\left(\mu_{ic}\left(\mathbf{x}_{ic}(t_{k-1})\right), \sigma_{ic}^2\left(\mathbf{x}_{ic}(t_{k-1})\right)\right)\right) \text{ if } p_m \le \xi, \\ \tilde{M}\left(\mathbf{x}_i(t_{k-1})\right) \text{ if } p_m > \xi, \end{cases} \tag{14}$$

where

- $\underline{\mathbf{x}}$, $\overline{\mathbf{x}}$ are the lower and upper boundary values of the decision variables;
- $u(0, 1)$ is the random value with a uniform distribution in the range of $(0, 1)$;

- Lambert $W\left(\ln \tilde{N}\left(\mu_{ic}\left(\mathbf{x}_{ic}(t_{k-1})\right),\ \sigma_{ic}^2\left(\mathbf{x}_{ic}(t_{k-1})\right)\right)\right),\ i \in I,\ c \in C$ is the random value with the heavy-tailed distribution and generated through the combination of the Lambert W function [20] and the log-normal distribution $\ln N\left(\mu_{ic},\ \sigma_{ic}^2\right)$ (to transform it to a heavy-tailed version), where $\mu_{ic}\left(\mathbf{x}_{ic}(t_{k-1})\right)$, $\sigma_{ic}^2\left(\mathbf{x}_{ic}(t_{k-1})\right)$ are the mean value and the standard deviation calculated over decision variables $\mathbf{x}_{ic}(t_{k-1})$ belonging to *i*-ths individuals of *c*-ths clusters;
- $\tilde{M}\left(\mathbf{x}_i(t_{k-1})\right)$ is the random value generated with one of the well-known mutation operators, such as the Uniform Mutation (UM), the Power Mutation (PM), the Scalable-Uniform Mutation (SUM), etc.;
- p_m is the probability of the CL mutation,
- ξ is the given threshold value.

When using the MOPSO, the velocity vector for the decision variables is calculated, which determines the position of the *i*-ths particles $(i \in I)$ in the space of potential decisions at the moment t_k $(t_k \in T)$:

$$\mathbf{v}_i(t_k) = \theta \mathbf{v}_i(t_{k-1}) + c_1 q(0,\ 1)(\mathbf{x}_i^*(t_{k-1}) - \mathbf{x}_i(t_{k-1})) + \\ + c_2 e(0,\ 1)(\mathbf{x}^g(t_{k-1}) - \mathbf{x}_i(t_{k-1})). \tag{15}$$

Additionally, the values of decision variables are computed with the following updating the archive of nondominated solutions if condition (10) is fulfilled for these:

$$\mathbf{x}_i(t_k) = \begin{cases} \mathbf{x}_i(t_{k-1}) + \mathbf{v}_i(t_{k-1}) \text{ if } \mathbf{x}_i(t_{k-1}) + \mathbf{v}_i(t_{k-1}) \in [\underline{\mathbf{x}},\ \overline{\mathbf{x}}], \\ \mathbf{x}_i(t_{k-1}) \text{ if } \mathbf{x}_i(t_{k-1}) + \mathbf{v}_i(t_{k-1}) \notin [\underline{\mathbf{x}},\ \overline{\mathbf{x}}]. \end{cases} \tag{16}$$

Here,

- $I = \{i_1,\ i_2,\ ...,\ i_{|I|}\}$ is the set indices of particles of MOPSO, where the total number of agent particles; is the best (i.e., nondominated) potential decisions obtained by particles of MOPSO during the search period and all particles at the moment t_k $(t_{k-1} \in T)$;
- $q(0,\ 1)$, $e(0,\ 1)$ are random values uniformly distributed on the interval $[0,\ 1]$;
- θ, c_1, c_2 are constants, the values of which, as a rule, are set as follows:

$$\theta \in [0.4, 1.4], c_1 \in [1.5, 2], c_2 \in [2, 2.5].$$

The developed MORCGA-MOPSO-II Algorithm is shown in Fig. 1. Here, the parameter sets the frequency of the interaction between the MORCGA and the MOPOSO.

As shown in Fig. 1, the proposed MORCGA-MOPSO-II Algorithm has two important features. The first of these is related to the selection procedure, which is based on the use of one of the clustering techniques (e.g., k-Means, DBSCAN, etc.).

The proposed clustering-based approach aims to form subpopulations of the most similar solutions, providing the possibility of selecting the most dissimilar parent individuals during the selection procedure and overcoming the well-known

inbreeding problem [1]. At the same time, to form new offspring individuals besides crossover operators, it is suggested to use the mutation directed towards the cluster centers calculated as a result of clustering. This process executed within the MORCGA is illustrated in Fig. 2.

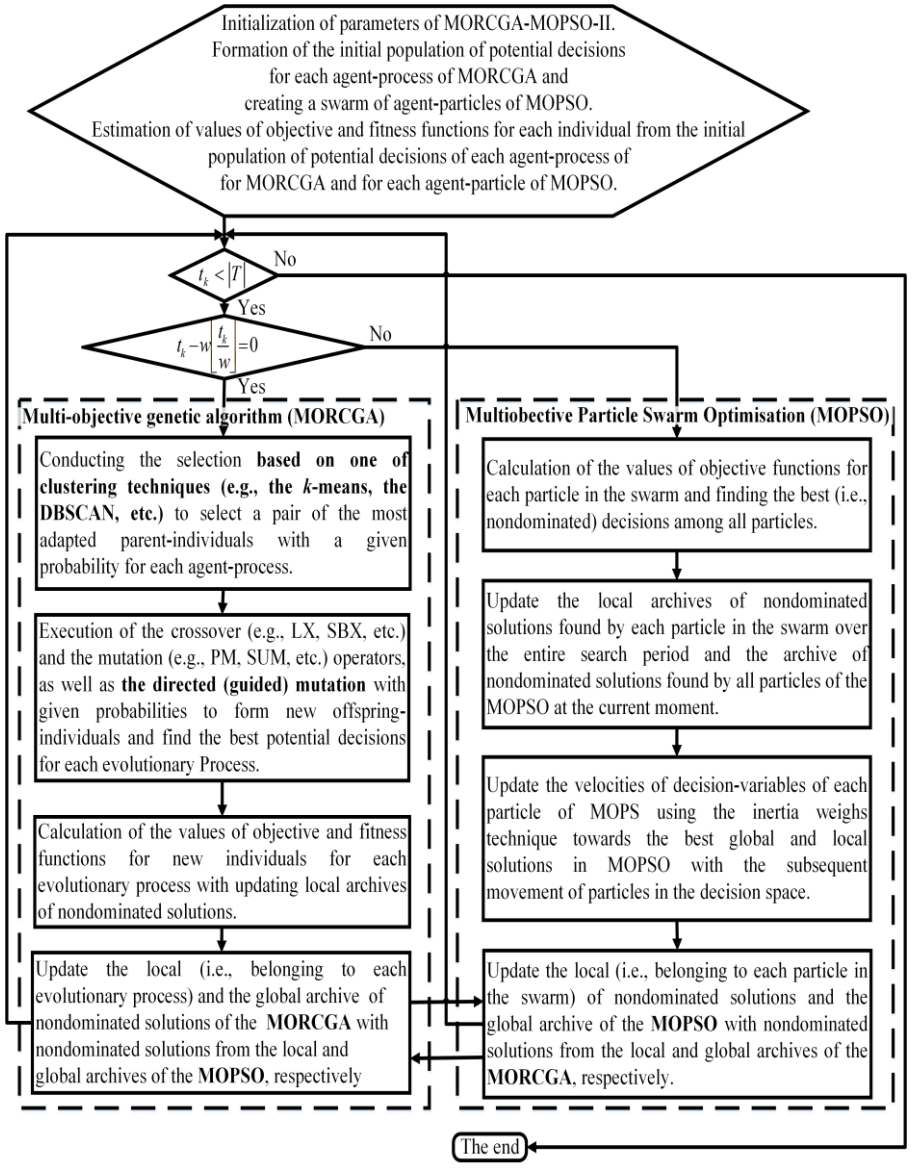

Fig 1. An improved bi-objective hybrid genetic algorithm (MORCGA-MOPSO-II)

As shown in Fig.2 some stages of the developed procedure are highlighted in grey because such elements are related to the MORCGA-MOPSO-II as a whole (see Fig. 1). Here, $g_k = 1, 2, ..., |G|$ is the index of internal iterations (to generate multiple offspring-individuals), $|G|$ is the total number of internal iterations.

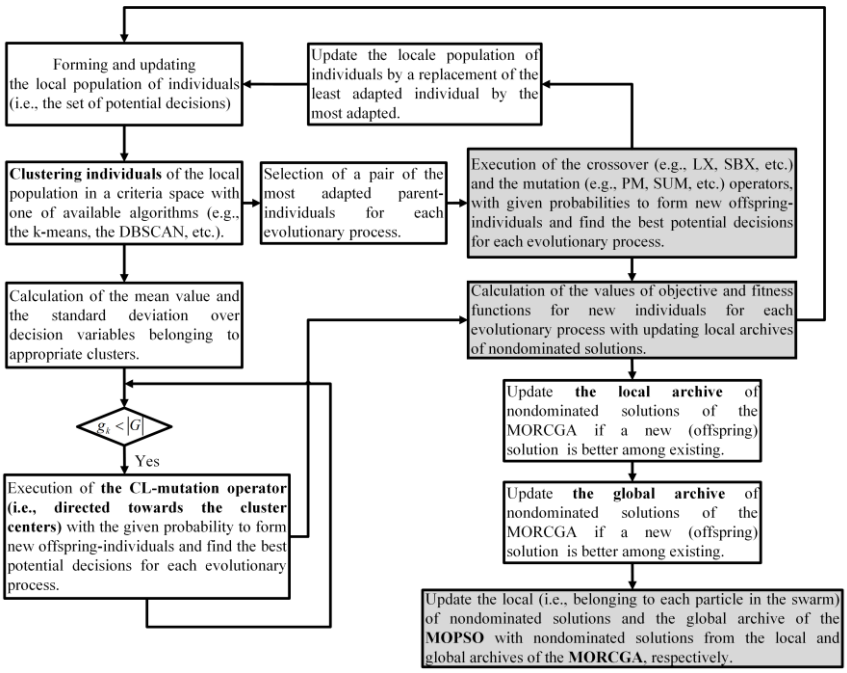

Fig 2. The procedure of forming new offspring-individuals based on clustering selection in MORCGA

2.3. Test results

In Table 1 known test instances [21] were used to test and verify the developed MORCGA-MOPSO-II Algorithm. The number of decision variables is set equal to 120 (i.e., $n = 120$ in Table 1), which can be considered as solving large-scale optimization problems.

At the initial stage, the efficiency of the MORCGA-MOPSO-II uses the k-Means technique for the selection procedure of the parent individuals. The CL-mutation to generate offspring-individual should be compared with other parallel evolutionary algorithms where a tournament selection is used. An assessment of performance metric values was conducted for the following parallel heuristic algorithms:

- SPEA2 is the multiobjective evolutionary algorithm based on the estimation of the Pareto strengths [22];
- NSGA-II is the multiobjective evolutionary algorithm based on the fast non-dominated sorting technique [23];
- FCGA is the multiobjective evolutionary algorithm based on fuzzy clustering and applied in ITS [8];
- MOPSO is the multi-objective particle swarm algorithm in which an interaction between all particles is carried out to find the global one [24].
- MORCGA-MOPSO is the parallel hybrid real-coded genetic algorithm that uses a tournament selection [5];
- MORCGA-MOPSO-II is the hybrid algorithm that combines the Multi-Objective Real-Coded Genetic Algorithm (MORCGA) with the Multi-Objective

Particle Swarm Optimization (MOPSO) and uses the clustering-based selection (equations (6)-(10)) with subsequent CL-mutations (equations (13)-(14)).

Table 1. Test instances for MORCGA-MOPSO-II

Test instances	Problem statement (objectives to be minimized)	Feasible ranges
FT1 – Zitzler-Deb-Thiele's function No 1	$\begin{cases} f_1 = x_1; \\ f_2 = gh, \end{cases}$ where $\begin{cases} g = 1 + \frac{9}{n-1}\sum_{j=2}^{n} x_j; \\ h = 1 - \sqrt{x_1 / g} \end{cases}$	$0 \le x_j \le 1,$ $1 \le j \le n$
FT2 – Zitzler-Deb-Thiele's function No 2	$\begin{cases} f_1 = x_1; \\ f_2 = gh, \end{cases}$ where $\begin{cases} g = 1 + \frac{9}{n-1}\sum_{j=2}^{n} x_j; \\ h = 1 - (x_1 / g)^2 \end{cases}$	$0 \le x_j \le 1,$ $1 \le j \le n$
FT3 – Zitzler-Deb-Thiele's function No 3	$\begin{cases} f_1 = x_1; \\ f_2 = gh, \end{cases}$ where $\begin{cases} g = 1 + \frac{9}{n-1}\sum_{j=2}^{n} x_j; \\ h = 1 - \sqrt{x_1 / g} - (x_1 / g)\sin(10\pi x_1) \end{cases}$	$0 \le x_j \le 1,$ $1 \le j \le n$
FT4 – Zitzler-Deb-Thiele's function No 4	$\begin{cases} f_1 = x_1; \\ f_2 = gh, \end{cases}$ where $\begin{cases} g = 1 + 10(n-1) + \sum_{j=2}^{n}\left(x_j^2 - 10\cos(4\pi x_j)\right); \\ h = 1 - \sqrt{x_1 / g} \end{cases}$	$0 \le x_1 \le 1,$ $-5 \le x_j \le 5,$ $2 \le j \le n$
FT5 – Zitzler-Deb-Thiele's function No 5	$\begin{cases} f_1 = 1 - \exp(-4x_1)\sin^6(6\pi x_1); \\ f_2 = gh, \end{cases}$ where $\begin{cases} g = 1 + 9\left(\sum_{j=2}^{n} x_j / (n-1)\right)^{0.25}; \\ h = 1 - (x_1 / g)^2 \end{cases}$	$0 \le x_j \le 1,$ $2 \le j \le n$

Afterward, the performance of the MORCGA-MOPSO-II Algorithm is assessed in comparison with other known parallel multicriteria optimization algorithms according to the most important criteria [25]:

- LHV is the metric characterizing the logarithmic hypervolume, which determines the total area of space covered by the Pareto front (it is to be maximized);
- CPF is the Pareto front cardinality characterizing the number of calculated non-dominated solutions belonging to the Pareto front (it is to be maximized);

- PT (in s) is the processing time spent to get the Pareto-optimal solutions.

Optimization experiments were conducted on the portable supercomputer at the Central Economics and Mathematics Institute of the Russian Academy of Sciences (DSWS PRO with 2x Intel Xeon Silver 4114, 1x NVIDIA QUADRO RTX 6000) using 100 evolutionary processes in the MORCGA and particles in MOPSO.

Table 2. Evaluation of performance metrics of MORCGA-MOPSO-II

Performance metrics	MORCGA-MOPSO-II	Other evolutionary algorithms				
		MORCGA-MOPSO	SPEA2	NSGA-II	FCGA	MOPSO
FT1 – Zitzler-Deb-Thiele's function No 1						
LHV	–0.00053	–0.00054	–0.00054	–0.00054	–0.00054	–0.00059
CPF	751	751	749	748	749	744
PT, s	143.3	126.0	76.5	101.0	122.5	96.5
FT2 – Zitzler-Deb-Thiele's function No 2						
LHV	–0.00054	–0.00055	–0.00055	–0.00055	–0.00055	–0.00099
CPF	748	747	747	743	744	741
PT, s	139.5	124.0	75.5	98.0	110.0	85.5
FT3 – Zitzler-Deb-Thiele's function No 3						
LHV	0.20205	0.20151	0.20173	0.20133	0.20210	0.20105
CPF	260	259	261	260	261	257
PT, s	126.1	104.5	38.0	60.0	74.5	81.0
FT4 – Zitzler-Deb-Thiele's function No 4						
LHV	3.20729	3.20409	3.20411	3.20411	3.20411	1.52874
CPF	24	20	74	109	110	9
PT, s	67.6	53.3	51.7	82.0	78.3	88.0
FT5 – Zitzler-Deb-Thiele's function No 5						
LHV	–0.16147	–0.19354	–0.21217	–0.21070	–0.19006	–0.29704
CPF	30	21	23	37	49	29
PT, s	101.2	99.7	28.4	77.3	79.3	81.6

The following values of control parameters are used:

- total number of iterations: $|T|$, is set to 100;
- population size: $|I|$, is set to 100;
- a total number of internal iterations with the CL-mutation: $|G|$, is set to 1;
- total number of clusters in MORCGA-MOPSO-II: $|C|$, is set to 10;
- frequency of the interaction between the MORCGA and the MOPOSO: w, is set to 5.

The results of optimization experiments (i.e., the average values of performance metrics) completed using the MORCGA-MOPSO-II (with the selection based on k-Means clustering) in comparison with other evolutionary algorithms are presented in Table 2.

As evident from Table 2, the MORCGA-MOPSO-II (which uses a clustering-based selection method and CL-mutations) outperforms the MORCGA-MOPSO (which uses a tournament selection) in terms of the LHV and the CPF for the most of test instances. At the same time, it maintains a comparable level of time efficiency (i.e., PT). The main advantage of the proposed MORCGA-MOPSO-II Algorithm is that it can improve the values of the most important performance metrics (such as LHV and CPF), in comparison to other evolutionary algorithms. Although the

MOPSO often demonstrates better values for time efficiency than many genetic algorithms, traditional swarm methods for multi-objective optimization do not produce solutions of the required quality. Therefore, it is justified to develop hybrid algorithms such as MORCGA-MOPSO-II.

Table 3. Evaluation of performance metrics of MORCGA-MOPSO-II at different clustering techniques

Performance metrics	Method of clustering-based selection			
	k-Means	hierarchical clustering	*c*-Means	DBSCAN
FT1 – Zitzler-Deb-Thiele's function No 1				
LHV	–0.00053	–0.00053	–0.00052	–0.00053
CPF	751	744	752	748
PT, s	143.3	142.0	343.0	141.0
FT2 – Zitzler-Deb-Thiele's function No 2				
LHV	–0.00054	–0.00054	–0.00053	–0.00054
CPF	748	744	749	745
PT, s	139.5	129.9	329.2	128.4
FT3 – Zitzler-Deb-Thiele's function No 3				
LHV	0.20205	0.20219	0.23961	0.20201
CPF	260	260	263	260
PT, s	126.1	122.7	280.6	120.9
FT4 – Zitzler-Deb-Thiele's function No 4				
LHV	3.20729	3.20670	3.20785	3.20551
CPF	24	22	48	19
PT, s	67.6	65.3	241.1	63.6
FT5 – Zitzler-Deb-Thiele's function No 5				
LHV	–0.16147	–0.13686	–0.11601	–0.19833
CPF	30	29	37	27
PT, s	101.2	99.5	171.4	95.1

In the second stage, the effectiveness of the MORCGA-MOPSO-II is assessed when using different clustering techniques (Table 3).

As seen in Table 3, using the k-Means algorithm in the selection procedure of the MORCGA-MOPSO-II is preferable to the other techniques in terms of the values of the main performance metrics. However, the c-Means algorithm that is the most time-expensive provides the best values of the LHV and the CPF for all tests. The sensitivity tests for the LHV and the CPF completed using the MORCGA-MOPSO-II with the selection procedure based on the k-Means technique are shown in Fig. 3.

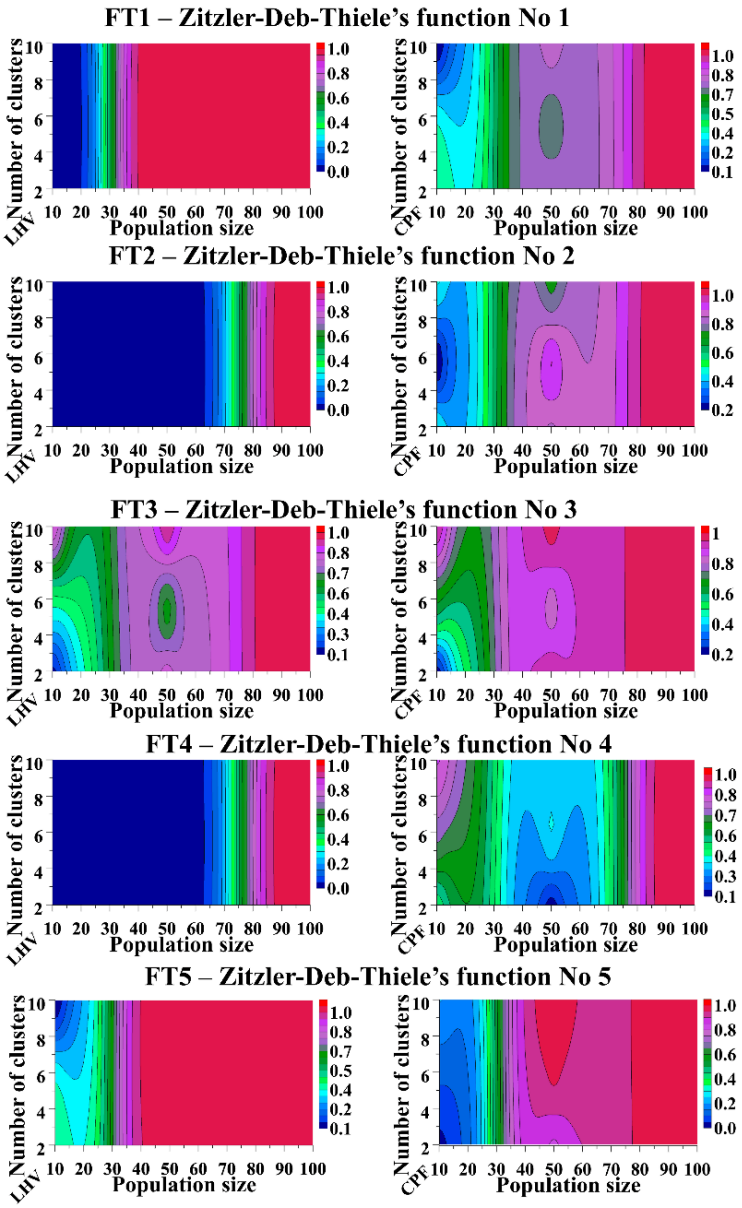

Fig. 3. Sensitivity tests completed with the MORCGA-MOPSO-II: dependencies of the normalized values of the LHV and CPF on the total number of clusters and the population size, where the best values of performance metrics correspond to 1 and the worst values of performance metrics correspond to 0

Fig. 3 shows that, as a rule, the performance metrics for the MORCGA-MOPSO-II Algorithm improve with an increase in the population size. At the same time, the total number of clusters to be set up depends on the solved optimization problem. As evident from Fig. 3, the best values of the total

number of clusters are different for various test instances. This is consistent with the known principles concerning the need for a choice of the number of clusters optimal in compactness criteria.

3. Results of simulation and optimization experiments

The results of optimization experiments aiming at solving the **Problem A*** completed with the proposed genetic algorithm (MORCGA-MOPSO-II) are presented in Fig. 4. These experiments were carried out under different configurations of the initial distribution of agents (i.e., sellers and buyers) in a discrete space of the stochastic agent-based model of goods exchange [4]. At the same time, the total number of economic agents equals 2000, and half of them with money.

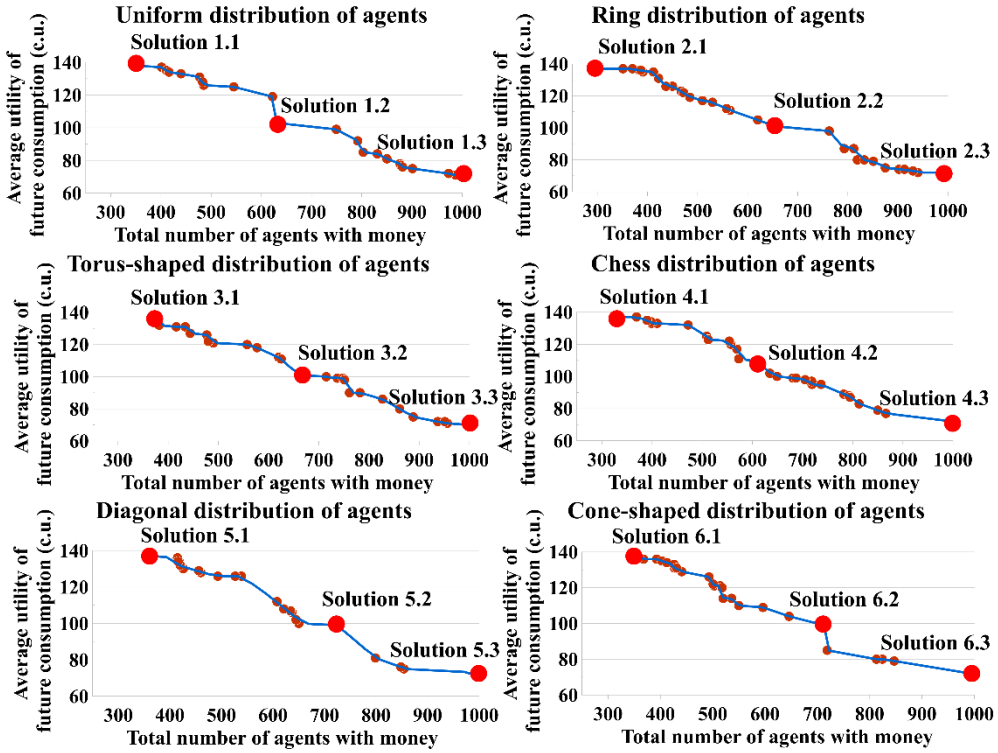

Fig. 4. The Pareto fronts computed with the use of the MORCGA-MOPSO-II and the stochastic agent-based model of goods exchange

Table 4 presents the values of decision variables corresponding to the most different optimal solutions belonging to the Pareto fronts shown in Fig. 4.

Table 4. The values of decision variables

Configurations of the initial distribution of agents	Optimal solutions	Decision variables (control parameters)						
		μ_b	σ_b^2	μ_m	σ_m^2	α	r	h
Uniform distribution of agents	Solution 1.1	0.432	0.078	0.645	0.034	0.012	12	0.157
	Solution 1.2	0.087	0.048	–0.618	0.677	0.461	7	0.044
	Solution 1.3	0.084	0.060	–0.750	0.133	0.295	13	0.157
Ring distribution of agents	Solution 2.1	0.746	0.078	0.554	0.293	0.736	19	0.615
	Solution 2.2	0.576	0.235	0.139	0.928	0.045	5	0.150
	Solution 2.3	0.619	0.111	–0.899	0.239	0.100	10	0.626
Torus-shaped distribution of agents	Solution 3.1	0.185	0.121	0.082	0.028	0.868	17	0.531
	Solution 3.2	0.555	0.433	–0.040	0.688	0.761	4	0.918
	Solution 3.3	0.595	0.112	–0.821	0.031	0.761	6	0.702
Chess distribution of agents	Solution 4.1	0.488	0.110	0.313	0.073	0.115	17	0.246
	Solution 4.2	0.012	0.073	0.023	0.522	0.327	4	0.000
	Solution 4.3	0.703	0.110	–0.869	0.146	0.115	15	0.846
Diagonal distribution of agents	Solution 5.1	0.385	0.203	0.043	0.041	0.004	14	0.164
	Solution 5.2	0.722	0.095	0.499	0.070	0.595	2	0.654
	Solution 5.3	0.143	0.149	–0.838	0.228	0.105	13	0.242
Cone-shaped distribution of agents	Solution 6.1	0.165	0.027	0.248	0.031	0.236	14	0.185
	Solution 6.2	0.772	0.098	0.270	0.078	0.541	2	0.322
	Solution 6.3	0.424	0.009	–0.558	0.024	0.732	11	0.064

As evident from Fig. 4 and Table 4, the MORCGA-MOPSO-II Algorithm allows us to find optimal solutions and approximate the Pareto front for the considered bi-objective optimization problem (i.e., Problem A). As the total number of agents with money decreases, the average utility of future consumption also increases, as this leads to an increase in the number of trade transactions. However, when the concentration of money becomes too high (i.e., there is significant monetary inequality between agents), average consumption decreases. Therefore, the optimal number of agents with money lies in the range of 300-1000 agents, where 1000 corresponds to the initial number of these agents (Fig. 4).

5. Conclusion

This paper proposes an improved parallel biobjective hybrid real-coded genetic algorithm with clustering-based selection (MORCGA-MOPSO-II). The main feature of this algorithm is the proposed selection procedure, which is based on the use of one of the clustering techniques (e.g., the k-Means, the c-Means, etc.). Moreover, it is suggested to use a novel clustering-based mutation operator (CL-mutation) directed towards the obtained cluster centers. The CL-mutation generates multiple offspring

individuals with the use of heavy-tailed distributions. At the same time, Multi-Objective Particle Swarm Optimization (MOPSO) and some other genetic algorithms such as SPEA2 can achieve the best values for time efficiency. However, hybrid genetic algorithms such as MORCGA-MOPSO-II have greater opportunities to improve the quality of the obtained solutions, in particular, due to their ability to use clustering-based selection and subsequent guided mutations, such as the CL mutation proposed in this study.

The proposed algorithm (MORCGA-MOPSO-II) allows us to find optimal solutions and approximate the Pareto fronts for biobjective optimization problems. In particular, it aims to maximize the average utility of future consumption for an ensemble of agents and the total number of agents with money over the sets of control parameters, such as the readiness of agents to conclude barter and monetary transactions, the radius of trading interactions, etc.

Further research will focus on applying the proposed hybrid real-coded genetic algorithm in large-scale multiagent socioeconomic systems.

***Acknowledgments*:** The research was supported by RSF (Project No 23-21-00012).

References

1. Whitley, D. A Genetic Algorithm Tutorial. – Statistics and Computing, Vol. **4,** 1994, No **2**, pp. 65-85.
2. Herrera, F., M. Lozano, J. L. Verdega. Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioral Analysis. – Artificial Intelligence Review, Vol. **12**, 1998, No 4, pp. 265-319.
3. Kennedy, J., R. Eberhart. Particle Swarm Optimization. – In: Proc. of IEEE International Conference on Neural Networks, Vol. **4**. 1995, pp. 1942-1948.
4. Akopov, A. S., A. L. Beklaryan, A. A. Zhukova. Optimization of Characteristics for a Stochastic Agent-Based Model of Goods Exchange with the Use of Parallel Hybrid Genetic Algorithm. – Cybernetics and Information Technologies, Vol. **20**, 2023, No 3, pp. 45-63.
5. Akopov, A. S., A. L. Beklaryan. Optimization of Behavior Strategies within the Simulation Model of a Multi-Agent Socio-Economic System. – Ekonomika i Matematicheskie Metody, Vol. **59**, 2023, No 3, pp. 117-131 (In Russian).
6. Akopov, A. S., L. A. Beklaryan, A. L. Beklaryan. Simulation-Based Optimisation for Autonomous Transportation Systems Using a Parallel Real-Coded Genetic Algorithm with Scalable Nonuniform Mutation. – Cybernetics and Information Technologies, Vol. **21**, 2021, No 3, pp. 127-144.
7. Akopov, A. S., L. A. Beklaryan, A. L. Beklaryan. Cluster-Based Optimization of an Evacuation Process Using a Parallel Bi-Objective Real-Coded Genetic Algorithm. – Cybernetics and Information Technologies, Vol. **20**, 2020, No 3, pp. 45-63.
8. Akopov, A. S., L. A. Beklaryan, M. Thakur. Improvement of Maneuverability within a Multiagent Fuzzy Transportation System with the Use of Parallel Biobjective Real-Coded Genetic Algorithm. – IEEE Transactions on Intelligent Transportation Systems, Vol. **23**, 2022, No 8, pp. 12648-12664.
9. Anfyorov, M. A. Genetic Clustering Algorithm. – Russian Technological Journal, Vol. **7**, 2019, No 6, pp. 134-150.
10. Lloyd, S. Least Squares Quantization in PCM. – IEEE Transactions on Information Theory, Vol. **28**, 1982, No 2, pp. 129-137.
11. Ward, J. H., Jr. Hierarchical Grouping to Optimize an Objective Function. – Journal of the American Statistical Association, Vol. **58**, 1963, pp. 236-244.

12. D u n n, J. C. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. – Journal of Cybernetics, Vol. **3**, 1973, No 3, pp. 32-57.
13. E s t e r, M., H.-P. K r i e g e l, J. S a n d e r, X. X u. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. – In: Proc. of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), AAAI Press, 1996, pp. 226-231.
14. Z e e b a r e e, D. Q., H. H a r o n, A. M. A b d u l a z e e z, S. R. Z e e b a r e e. Combination of k-Means Clustering with Genetic Algorithm: A Review. – International Journal of Applied Engineering Research, Vol. **12**, 2017, pp. 14238-14245.
15. K u m a r, A., K. D e b. Real-Coded Genetic Algorithms with Simulated Binary Crossover: Studies on Multimodal and Multiobjective Problems. – Complex Systems, Vol. **9**, 1995, pp. 431-454.
16. D e e p, K., M. T h a k u r. A New Crossover Operator for Real Coded Genetic Algorithms. – Applied Mathematics and Computation, Vol. **188**, 2007, No 1, pp. 895-911.
17. D e e p, K., M. T h a k u r. A New Mutation Operator for Real Coded Genetic Algorithms. – Applied Mathematics and Computation, Vol. **193**, 2007, No 1, pp. 211-230.
18. M i l l e r, B., D. G o l d b e r g. Genetic Algorithms, Tournament Selection, and the Effects of Noise (PDF). – Complex Systems, Vol. **9**, 1995, pp. 193-212.
19. F r i e d r i c h, T., K. B r i n g m a n n, T. V o ß, C. I g e l. The Logarithmic Hypervolume Indicator. – In: Proc. of 11th Workshop Found. Genetic Algorithms, 2011, pp. 81-92.
20. L e h t o n e n, J. The Lambert W Function in Ecological and Evolutionary Models. – Methods in Ecology and Evolution, Vol. **7**, 2016, No 9, pp. 1110-1118.
21. Z i t z l e r, E., D. K a l y a n m o y, T. L o t h a r. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. – Evolutionary Computation, Vol. **8**, 2000, No 2, pp. 173-195.
22. Z i t z l e r, E., M. L a u m a n n s, L. T h i e l e. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. – Swiss Federal Inst. Technol., Zürich, Switzerland, TIK-Rep. 103, 2001.
23. D e b, K., A. P r a t a p, S. A g a r w a l, T. M e y a r i v a n. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. – IEEE Transactions on Evolutionary Computation, Vol. **6**, 2002, No 2, pp. 182-197.
24. X i a o h u i, H., R. E b e r h a r t. Multiobjective Optimization Using Dynamic Neighborhood Particle Swarm Optimization – In: Proc. of 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No 02TH8600), Honolulu, HI, USA, Vol. **2**, 2002, pp. 1677-1681.
25. J i a n g, S., Y.-S. O n g, J. Z h a n g, L. F e n g. Consistencies and Contradictions of Performance Metrics in Multiobjective Optimization. – IEEE Transactions on Cybernetics, Vol. **44**, 2014, No 12, pp. 2391-2404.

Received: 06.02.2024; Second Version: 29.02.2024; Accepted: 05.03.2024 (fast track)