A Clustering-Based Hybrid Particle Swarm Optimization Algorithm for Solving a Multisectoral Agent-Based Model

Andranik S. AKOPOV

Central Economics and Mathematics Institute of Russian Academy of Sciences, 47, Nachimovski Prosp., 117418, Moscow, Russia

MIREA - Russian Technological University, 78, Prospekt Vernadskogo, 119454, Moscow, Russia

Moscow Institute of Physics and Technology, 9, Institutsky lane, 141700, Dolgoprudny, Moscow region, Russia
andranik.s.akopov@ieee.org

Abstract: This paper presents the new Clustering-Based Hybrid Particle Swarm Optimization (CBHPSO) algorithm. This algorithm was designed to solve biobjective optimization problems and it was used for finding trade-offs in a multisectoral agent-based model of trade interactions. This model includes multiple interacting agent enterprises belonging to different economic sectors. At the same time, the values of the control parameters of this stochastic multiagent system (MAS) need to be optimized. Therefore, CBHPSO has been developed and aggregated with this MAS by means of the objective functions. The main feature of the CBHPSO algorithm consists in the use of clustering techniques, such as the k-means algorithm, to form subsets of non-dominated solutions shared among the swarm particles in each cluster. The values of the performance metrics employed for CBHPSO and other well-known multi-objective evolutionary algorithms (SPEA2, NSGA-II, FCGA and BORCGA-BOPSO and MOPSO) were compared. As a conclusion, it was found that the velocities of decision variables in the particle swarm depend on specific non-dominant solutions of the clusters involved. It can be said that the main advantage of CBHPSO lies in the quality of the Pareto front approximation. Thus, it was demonstrated that the CBHPSO algorithm can be applied to search for improved characteristics of the employed MAS.

Keywords: Particle swarm optimization, Agent-based modeling, Genetic algorithms, Clustering, Simulation of trade interactions, Multiagent systems, Multiobjective optimization, Multisectoral models.

1. Introduction

Over the past two decades, several control and decision-making systems have been developed using Particle Swarm Optimization algorithms (Kennedy & Eberhart, 1995). Many modern PSO algorithms combine different optimization methods, such as genetic algorithms, differential evolution, and other bio-inspired techniques (Akopov & Beklaryan, 2024; Molaei et al., 2021). Such hybridization helps to overcome the problem of premature convergence that is inherent to PSO, while improving the quality of the obtained solutions. There are single-objective and multi-objective versions of PSO known as MOPSO.

The PSO and MOPSO algorithms have been used to solve optimization problems in various systems (Demidova & Gorchakov, 2022). For example, they have been applied to the scheduling problem in the Internet of vehicles (Li & Wang, 2023), to control magnetic levitation systems (Engda et al., 2023), to improve prediction accuracy, to design PID controllers (Gao et al., 2021) and in other systems.

The PSO combined with the Real-Coded Genetic Algorithm (RCGA) enables the optimization of the characteristics of Multiagent Systems (MASs), as demonstrated in (Akopov & Beklaryan, 2024; Akopov et al., 2020).

MASs are simulation-based systems comprising multiple intelligent agents. Optimizing MAS characteristics is a complex computational challenge because the values of the objective function are computed as a result of simulation modeling. In MAS, a large number of agents with individual decision-making systems interact with each other and contribute to the objectives of the whole system (Balaji & Srinivasan, 2009).

Some studies propose using PSO for efficient data clustering (van der Merwe & Engelbrecht, 2003; Miles et al., 2020). At the same time, the PSO and the MOPSO algorithms can be enhanced through clustering techniques, as it was demonstrated in this paper.

There are some known limitations to existing particle swarm optimization (PSO) algorithms, such as their difficulty in finding optimal solutions with high accuracy, premature convergence to local extremes, and other issues (Gbenga & Ramlan, 2016). These limitations are particularly significant when solving multiobjective optimization problems in MAS with dynamic control over agent states. In such systems, even a minor change in the values of control parameters can cause a significant shift in the values of the objective functions due to the impact of scaling in agent-based models. Therefore, the development of new evolutionary optimization algorithms is crucial for MASs.

https://doi.org/10.24846/v33i2y202408 ICI Bucharest © Copyright 2012-2024. All rights reserved

This paper presents a novel clustering-based hybrid particle swarm optimization algorithm (CBHPSO) for finding trade-offs in a multi-sectoral agent-based model of trade interactions.

The multisectoral agent-based model considered here is an assembly of agent enterprises belonging to different sectors of the economy which interact with each other to buy intermediate products and produce end products. Within the system, each agent aims to maximize its cumulative profit and the number of transactions with other agents. As a rule, these objectives are in conflict, as the growth of sales requires a decrease in prices, which can lead to a fall in the profit.

The trade interactions discussed here are transactions between agents, as a result of which some enterprises receive products for their intermediate consumption, while other firms receive money from selling the end products. At the same time, the production of these end products is only possible if agents have all necessary intermediate products in quantities determined by technological requirements.

The proposed algorithm (CBHPSO) has several advantages over other multi-objective optimization methods, including the accuracy of the Pareto front approximation and its higher time efficiency. The CBHPSO algorithm can be used to find solutions within the developed MAS.

The main differences between the CBHPSO algorithm and other PSO algorithms can be summarized as follows:

- CBHPSO uses clustering techniques to improve the quality of solutions obtained in solving biobjective optimization problems. This means that the local archives of nondominated solutions of particles in the swarm are combined and shared among all particles belonging to appropriate clusters;

- CBHPSO periodically interacts with the multiobjective real-coded genetic algorithm (MORCGA) to update its local and global archives of non-dominated solutions with the non-dominated solutions of MORCGA. This approach aims to overcome the problem of premature convergence and improve the quality of the Pareto front approximation.

The remainder of this paper is organized as follows. Section 2 introduces the developed multisectoral agent-based model of trade interactions and the proposed CBHPSO optimization algorithm. Section 3 presents the results of the optimization experiments. Section 4 discusses the main advantages and limitations of this approach, while Section 5 concludes this paper.

2. Material and Methods

2.1 Multisectoral Agent-Based Model

The proposed approach is based on principles of the previously developed stochastic agent-based model of goods exchange (Akopov et. al., 2023). In this first simple model, agents are individuals who get involved in bartering or monetary transactions maximizing the utility of the future consumption. Such agents do not need the raw materials and time to create new products.

The multiagent system presented in this paper consists of agent enterprises. Agent enterprises can be in the stock formation state, in the production state or in the selling state. Such agents also act as both sellers and consumers making individual decisions on the purchasing intermediate products or selling the end product at any moment of time. In such a multisectoral model the input-output coefficients (technical coefficients) and prices of intermediate and end products are significant characteristics (Brems, 1957). At the same time, these prices depend on the spatial density of agent enterprises belonging to the same economic sector (Gu & Wenzel, 2009).

A brief abstract description of the model is presented below.

Here,

- $T=\{t_0,\ t_1\ ...,\ |T|\}$ is the set of time moments (by days), $|T|$ is the total number of time moments; $t_0 \in T$, $t_{|T|} \in T$ are the initial and final moments of the model;

- $J=\{j_1, j_2, ..., j_{|J|}\}$ is the set of indices of the economy sectors, where $|J|$ is the total number of the economy sectors;

- $I_j=\{i_{j_1},\ i_{j_2},\ ...,\ i_{j_{|I_j|}}\}$ is the set of indices of agent enterprises of the *j*-th economy sector, $|I_j|$ is the total number of agent enterprises of the *j*-th economy sector.

- $s_i(t_k) \in \{1,\ 2,\ 3\}$, $i \in I_j$, $j \in J$ represents the states of the agent enterprise at moment t_k, $(t_k \in T)$: 1 is the stock formation state, 2 is the production state, 3 is the selling state;

- $k_{j\tilde{j}} \in [0,\ 1]$, $j, \tilde{j} \in J$ represents input-output coefficients the values of which depend on the production technologies used in the j-th economy sector;

- $d_{i\tilde{j}}(t_k)$, $\Delta d_{i\tilde{j}}(t_k)$, $i \in I_j, j, \tilde{j} \in J$ are the volumes of the intermediate products of the i-th agent of the j-th economy sector, produced in the j-th economy sectors and their increments due to purchasing at moment t_k, $(t_k \in T)$;
- $p_i(t_k)$, $i \in I_j, j \in J$ is the volume of end product produced by the i-th agent of the j-th economy sector at moment t_k, $(t_k \in T)$;
- $\tilde{p}_i(t_k)$, $\Delta\tilde{p}_i(t_k)$, $i \in I_j, j \in J$ represent the sales volume for the end product of the i-th agent of the j-th economy sector and its decrement due to selling at moment t_k, $(t_k \in T)$;
- $r_i(t_k)$, $e_i(t_k) \in \{0, 1\}$, $i \in I, j \in J$ are the states of readiness of the i-th agent enterprise for purchasing intermediate products or selling the end product (i.e. when an agent is willing to buy or sell) for each moment of time t_k, $(t_k \in T)$: 0 indicates that the transactions are not allowed, 1 indicates that the transactions are allowed. This means that the parameter $r_i(t_k)$ influences the number of intermediate products intended to be purchased by the i-th agent and the parameter $e_i(t_k)$ influences the number of intermediate products intended to be sold by the i-th agent, allowing or prohibiting appropriate deals at moment t_k, $(t_k \in T)$. The values of $r_i(t_k)$, $e_i(t_k)$ can be generated by using log-normal or heavy-tailed distribution and other distributions.

The state of the i-th agent enterprise $(i \in I_j)$ at moment t_k, $(t_k \in T)$ is given with the following rule:

$$s_i(t_k) = \begin{cases} 1, \text{ if I is true,} \\ 2, \text{ if II is true,} \\ 3, \text{ if III is true,} \end{cases} \tag{1}$$

where:

I. for the i-th agent $(i \in I_j, j \in J, r_i(t_k) = 1)$, $d_{i\tilde{j}}(t_{k-1}) + \Delta d_{i\tilde{j}}(t_k) < p_i(t_k)k_{j\tilde{j}}$ is fulfilled for at least one $\tilde{j} \in J$, and there exists at least one $\tilde{i}$-th agent such that $\tilde{i} \in I_{\tilde{j}}, \tilde{j} \in J$, $e_{\tilde{i}}(t_k) = 1$, *which means that stocks of intermediate products are being created and the stock formation is possible;*

II. for the i-th agent $(i \in I_j, j \in J)$, $d_{i\tilde{j}}(t_{k-1}) + \Delta d_{i\tilde{j}}(t_k) \geq p_i(t_k)k_{j\tilde{j}}$ for all $\tilde{j} \in J$, *which means that all stocks of intermediate products are created and the agent enterprise is in the production state;*

III. for the i-th agent $(i \in I_j, j \in J, e_i(t_k) = 1)$, $\tilde{p}_i(t_{k-1}) + p_i(t_k) \geq \Delta\tilde{p}_i(t_k)$ is fulfilled, and there exists at least one i-th agent such that $\tilde{i} \in I_{\tilde{j}}$, $\tilde{j} \in J$, $r_{\tilde{i}}(t_k) = 1$, *which means that the stock of the end product is created and the sale is possible.*

The dynamics of stocks for intermediate products, and for the production and the sale of the end product for the i-th agent enterprise $(i \in I_j, j \in J)$ are given as:

$$d_{i\tilde{j}}(t_k) = \begin{cases} d_{i\tilde{j}}(t_{k-1}) + \Delta d_{i\tilde{j}}(t_k), \text{ if } s_i(t_k) = 1, \\ \begin{pmatrix} d_{i\tilde{j}}(t_{k-1}) \\ +\Delta d_{i\tilde{j}}(t_k) \\ -p_i(t_k)k_{j\tilde{j}} \end{pmatrix}, \text{ if } s_i(t_k) = 2, \end{cases} \tag{2}$$

$$p_i(t_k) = \begin{cases} \begin{pmatrix} p_i(t_{k-1}) \\ +\sum_{j=1}^{|J|}\sum_{\tilde{j}=1}^{|J|} p_i(t_k)k_{j\tilde{j}} \end{pmatrix}, \text{ if } s_i(t_k) = 2, \\ p_i(t_{k-1}), \text{ if } s_i(t_k) \neq 2, \end{cases} \tag{3}$$

$$\tilde{p}_i(t_k) = \begin{cases} \begin{pmatrix} \tilde{p}_i(t_{k-1}) \\ +p_i(t_k) \\ -\Delta\tilde{p}_i(t_k) \end{pmatrix}, \text{ if } s_i(t_k) = 3, \\ \tilde{p}_i(t_{k-1}), \text{ if } s_i(t_k) \neq 3. \end{cases} \tag{4}$$

When an agent enterprise is in the selling state, it searches for an agent buyer who needs an appropriate intermediate product. At the same time, the product price depends on the ratio between the number of sellers of a product and buyers located within the bounded area and the distance between them.

The price of the i-th agent seller $(i \in I_j, j \in J)$ for the i-th agent buyer $(\tilde{i} \in I_{\tilde{j}}, \tilde{j} \in J)$ can be given as:

$$\tilde{\pi}_{i\tilde{i}}(t_k) = \pi_j(t_{k-1})\left(\frac{M_i(t_k)}{N_i(t_k)}\right)^{\alpha}\left(\tilde{\delta}_{i\tilde{i}}(t_k)\right)^{\beta}, \tag{5}$$

$$M_i(t_k) = \sum_{\tilde{i}=1}^{|I_j|} m_{i\tilde{i}}(t_k), \tag{6}$$

$$N_i(t_k) = \sum_{\hat{i}=1}^{|I_j|} n_{i\hat{i}}(t_k), \tag{7}$$

$$m_{i\tilde{i}}(t_k) = \begin{cases} 1, \text{ if } \tilde{\delta}_{i\tilde{i}}(t_k) \leq \rho \text{ and } s_{\tilde{i}}(t_k) = 1, \\ 0, \text{ if } \tilde{\delta}_{i\tilde{i}}(t_k) > \rho \text{ or } s_{\tilde{i}}(t_k) \neq 1, \end{cases} \tag{8}$$

$$n_{i\hat{i}}(t_k) = \begin{cases} 1, \text{ if } \hat{\delta}_{i\hat{i}}(t_k) \leq \rho \text{ and } s_{\hat{i}}(t_k) = 3, \\ 0, \text{ if } \hat{\delta}_{i\hat{i}}(t_k) > \rho \text{ or } s_{\hat{i}}(t_k) \neq 3, \end{cases} \tag{9}$$

ICI Bucharest © Copyright 2012-2024. All rights reserved

where:

- $\{\tilde{\delta}_{i\tilde{i}}(t_k),\ \hat{\delta}_{i\hat{i}}(t_k)\},\ i,\tilde{i},\hat{i}\in I_j$ are the Euclidean distances between the i-th agent and the $\tilde{i}$-th agent buyer and between the i-th agent and the $\hat{i}$-th agent seller, respectively. These characteristics define the metric distances between relevant agents with the specified coordinates $\{x_i(t_k),\ y_i(t_k)\}$, $\{\tilde{x}_{\tilde{i}}(t_k),\ \tilde{y}_{\tilde{i}}(t_k)\}$, and $\{\hat{x}_{\hat{i}}(t_k),\ \hat{y}_{\hat{i}}(t_k)\}$:
$$\hat{\delta}_{i\tilde{i}}(t_k)=\sqrt{\left(x_i(t_k)-\tilde{x}_{\tilde{i}}(t_k)\right)^2+\left(y_i(t_k)-\tilde{y}_{\tilde{i}}(t_k)\right)^2},$$
$$\hat{\delta}_{i\hat{i}}(t_k)=\sqrt{\left(x_i(t_k)-\hat{x}_{\hat{i}}(t_k)\right)^2+\left(y_i(t_k)-\hat{y}_{\hat{i}}(t_k)\right)^2};$$
- $\pi_j(t_k),\ j\in J,\ t_k\in T$ is the average price for a product of the j-th economy sector;
- p is the radius of trade interactions (*control parameter*). The parameter defines the maximum distance between two agents at which their trade interactions can occur;
- α, β ∈ [0, 1] are coefficients.

The distance between the product of the i-th agent seller $(i\in I_j,\ j\in J)$ and the product of the $\tilde{i}$-th agent buyer $(\tilde{i}\in I_{\tilde{j}},\ \tilde{j}\in J)$, measured along the length of the arc of a numerical circle with evenly distributed numbers $1, 2, ..., |J|$ can be expressed as:

$$\delta_{i\tilde{i}}(t_k)=\frac{1}{|J|-1}\vartheta_{i\tilde{i}}(t_k)+\lambda\hat{\delta}_{i\tilde{i}}(t_k), \tag{10}$$

where

$$\vartheta_{i\tilde{i}}(t_k)=\min\left\{\begin{matrix}\left|j_i'(t_k)-\tilde{j}_{\tilde{i}}''(t_k)\right|,\\ \left|J\right|-\left|j_i'(t_k)-\tilde{j}_{\tilde{i}}''(t_k)\right|\end{matrix}\right\} \tag{11}$$

Here:

- λ is the coefficient (a small number) that determines the impact of the cost (related to the loss of benefits for the seller) due to the Euclidean distance between a seller and a buyer;
- $j_i'(t_k)\in J$ is the index of the product proposed by the i-th agent seller $(i\in I_j,\ j\in J)$ at moment t_k, $(t_k\in T)$;
- $\tilde{j}_{\tilde{i}}''(t_k)\in J$ is the index of the most desired product for the $\tilde{i}$-th agent buyer $(\tilde{i}\in I_{\tilde{j}},\ \tilde{j}\in J)$ at moment t_k, $(t_k\in T)$.

The average number of agents' deals is:

$$D=\frac{1}{\sum_{j=1}^{|J|}\left|I_j\right|}\sum_{t_k}^{|T_k|}\sum_{i}^{|I_j|}\sum_{\tilde{i}}^{|I_j|}g_{i\tilde{i}}(t_k), \tag{12}$$

$$g_{i\tilde{i}}(t_k)=\begin{cases}1, \text{ if } \begin{pmatrix}\delta_{i\tilde{i}}(t_k)\le\gamma \text{ and}\\ \Delta\pi_{i\tilde{i}}\le\eta \text{ and}\\ \hat{\delta}_{i\tilde{i}}(t_k)\le\rho\end{pmatrix},\\ 0, \text{ if } \begin{pmatrix}\delta_{i\tilde{i}}(t_k)>\gamma \text{ or}\\ \Delta\pi_{i\tilde{i}}>\eta \text{ or}\\ \hat{\delta}_{i\tilde{i}}(t_k)>\rho\end{pmatrix},\end{cases} \tag{13}$$

$$\Delta\pi_{i\tilde{i}}=\frac{\tilde{\pi}_{i\tilde{i}}(t_k)}{\pi_j(t_k)}, \tag{14}$$

where:

- $\gamma,\eta>0$ are the coefficients of contractuality (*control parameters*);
- $\tilde{\pi}_{i\tilde{i}}(t_k)$, $\pi_j(t_k),\ i_j\in I_j,\ j,\tilde{j}\in J$ are the price of the i-th agent seller for the i-th agent buyer and the average price in the j-th economy sector, respectively.

The average cumulative profit of agents is:

$$P=\frac{1}{\sum_{j=1}^{|J|}\left|I_j\right|}\sum_{t_k}^{|T_k|}\sum_{i}^{|I_j|}\sum_{\tilde{i}}^{|I_j|}\Delta d_{i\tilde{i}}(t_k)\pi_{i\tilde{i}}(t_k), \tag{15}$$

Problem A. The decision maker wants to find $\{\mu,\ \sigma^2,\ \tilde{\mu},\ \tilde{\sigma}^2\}$, $\{\rho,\ \gamma,\ \eta\}$, $\{d_{i\tilde{j}}(t_k),\ \Delta d_{i\tilde{j}}(t_k),\ p_i(t_k),\ \Delta\tilde{p}_i(t_k),\ t_k\}_{k=0}^{|T|}$ such that the average cumulative profit and the average number of deals of agents be maximized. The mathematical formulation of the bi-objective optimization problem is as follows:

$$\begin{cases}\max\limits_{\substack{\{\mu,\ \sigma^2,\ \tilde{\mu},\ \tilde{\sigma}^2\},\ \{\rho,\ \gamma,\ \eta\},\\ \{d_{i\tilde{j}}(t_k),\ \Delta d_{i\tilde{j}}(t_k),\ p_i(t_k),\ \Delta\tilde{p}_i(t_k),\ t_k\}_{k=0}^{|T|}}} P\\ \max\limits_{\substack{\{\mu,\ \sigma^2,\ \tilde{\mu},\ \tilde{\sigma}^2\},\ \{\rho,\ \gamma,\ \eta\},\\ \{d_{i\tilde{j}}(t_k),\ \Delta d_{i\tilde{j}}(t_k),\ p_i(t_k),\ \Delta\tilde{p}_i(t_k),\ t_k\}_{k=0}^{|T|}}} D\end{cases} \tag{16}$$

s.t.

$$\mu,\tilde{\mu}\in[-1,\ 1],\ \sigma^2,\tilde{\sigma}^2\in(0,\ 1],$$
$$\rho\in[1,\ \overline{\rho}],\ \gamma\in[0,\ 1],\ \eta\in[0,\ \overline{\eta}],$$
$$d_{i\tilde{j}}(t_k)\in[\underline{d}_{i\tilde{j}},\ \overline{d}_{i\tilde{j}}],\ \Delta d_{i\tilde{j}}(t_k)\in[\Delta\underline{d}_{i\tilde{j}},\ \Delta\overline{d}_{i\tilde{j}}],$$
$$p_i(t_k)\in[\underline{p}_i,\ \overline{p}_i],\ \tilde{p}_i(t_k)\in[\underline{\tilde{p}}_i,\ \overline{\tilde{p}}_i],$$
$$\Delta\tilde{p}_i(t_k)\in[\Delta\underline{\tilde{p}}_i,\ \Delta\overline{\tilde{p}}_i] \text{ for all } i\in I_j,\ j,\tilde{j}\in J,\ t_k\in T.$$

Here, $\overline{\rho}$, $\overline{\eta}$ are the upper limits of the control parameter values and $\underline{d}_{i\tilde{j}}$, $\Delta\underline{d}_{i\tilde{j}}$, $\underline{p}_i$, $\underline{\tilde{p}}_i$, $\Delta\underline{\tilde{p}}_i$ and $\overline{d}_{i\tilde{j}}$, $\Delta\overline{d}_{i\tilde{j}}$, $\overline{p}_i$, $\overline{\tilde{p}}_i$, $\Delta\overline{\tilde{p}}_i$ are the lower and upper bounds of the values for internal variables of the model, respectively.

2.2 Clustering-Based Hybrid PSO

There is a number of multiobjective evolutionary algorithms, such as SPEA2 (Zitzler et al., 2001), NSGA-II (Deb et al., 2002), MOPSO (Xiaohui & Eberhart, 2002), FCGA (Akopov et al., 2022), and BORCGA-BOPSO (Akopov & Beklaryan, 2024). These bio-inspired methods, especially the hybrid ones (based on using particle swarm optimization and genetic algorithms) show a high performance in solving large-scale simulation-based optimization problems. However, further improvements to these algorithms in terms of their accuracy in approximating the Pareto fronts are still required.

The proposed algorithm (CBHPSO) uses clustering techniques to improve the quality of the obtained solutions in solving biobjective optimization problems.

In general terms, the following problem is considered:

$$\min F(\mathbf{x}) = \left(f_1(x), f_2(x)\right), \tag{17}$$

s.t.

$\mathbf{x} = (x_1, x_2, ..., x_n)' \in \Omega$ where $\mathbf{x} = (x_1, x_2, ..., x_n)'$ is a decision variable vector of dimension n, $\Omega = \prod_{i=1}^{n}[a_j, b_j]$ is the feasible region of the search space ($j = 1, 2, ..., n$ is the index of decision variables) and $f_m(x)$ represents the m-th objective functions $(m = 1, 2)$ computed with the use of the agent-based model.

The CBHPSO algorithm uses a swarm of particles to find nondominated solutions. Its main steps are presented in compact form below:

Algorithm 1. CBHPSO: main steps	
1:	**Initialization**
2:	**repeat**
3:	Calculating objective functions for all particles
4:	**If** condition of domination is fulfilled **then**
5:	Updating the archives of nondominated solutions
6:	**end if**
7:	Clustering particles in the criterion space
8:	Selection of the local and global archives of nondominated solutions
9	Updating the velocities and values of decision variables
10:	Interacting with the multiobjective genetic algorithm to exchange the optimal solutions
11:	**until** All iterations are completed

In Algorithm 1, the condition of domination can be given as:

$$\begin{aligned} & f_{\tilde{i}1}\left(\tilde{\mathbf{x}}_{\tilde{i}}(t_k)\right) \succ f_{i1}\left(\mathbf{x}_i(t_k)\right) \text{ and} \\ & f_{\tilde{i}2}\left(\tilde{\mathbf{x}}_{\tilde{i}}(t_k)\right) \succeq f_{i2}\left(\mathbf{x}_i(t_k)\right) \text{ or} \\ & f_{\tilde{i}2}\left(\tilde{\mathbf{x}}_{\tilde{i}}(t_k)\right) \succ f_{i2}\left(\mathbf{x}_i(t_k)\right) \text{ and} \\ & f_{\tilde{i}1}\left(\tilde{\mathbf{x}}_{\tilde{i}}(t_k)\right) \succeq f_{i1}\left(\mathbf{x}_i(t_k)\right) \text{ for } \forall\ \tilde{i} \in I, \end{aligned} \tag{18}$$

where:

- $T = \{t_1, t_2, ..., t_{|T|}\}$ is the set of iterations of the CBHPSO, where $|T|$ is the total number of iterations;
- $I = \{i, i, ..., i_{|I|}\}$ is the set of particles in the swarm, $i, \tilde{i} \in I$ represent indices of nondominated and other (i.e. existing) particles;
- $\mathbf{x}_i(t_k),\ \tilde{\mathbf{x}}_{\tilde{i}}(t_k),\ i, \tilde{i} \in I,\ t_k \in T$ are the decision variable vectors of the given i-th particle and other i-th particles.

The clustering of particles in the criterion space in Algorithm 1 is completed by means of *the k-means algorithm* (Lloyd, 1982). Unlike other PSO-based evolutionary algorithms (e.g. MOPSO, BORCGA-BOPSO), the CBHPSO operates with the subsets of particles clustered in the criterion space by means of the k-means algorithm. As a result, the best (nondominated) potential decisions obtained by all particles within the appropriate cluster $\mathbf{x}_i^*(t_k),\ i \in I,\ t_k \in T$ are formed.

Periodically, there is interaction between the multiobjective particle swarm optimization algorithm and the multiobjective genetic algorithm to exchange optimal solutions and prevent premature convergence in the CBHPSO algorithm.

In the CBHPSO algorithm, the velocity vector for the decision variables is calculated, which determines the position of the i-th particles $(i \in I)$ in the space of potential decisions at the moment t_k $(t_k \in T)$:

$$\begin{aligned} \mathbf{v}_i(t_k) &= \theta \mathbf{v}_i(t_{k-1}) \\ &+ \tilde{c}_1 q(0, 1)\begin{pmatrix} \mathbf{x}_i^*\left((t_{k-1}, h(1, |\tilde{I}_c|)\right) \\ -\mathbf{x}_i(t_{k-1}) \end{pmatrix} \\ &+ \tilde{c}_2 e(0, 1)\left(\mathbf{x}^g(t_{k-1}) - \mathbf{x}_i(t_{k-1})\right). \end{aligned} \tag{19}$$

Here:

$C = \{c_1, c_2, ..., c_{|C|}\}$ is the set of particle clusters, where $|C|$ is the total number of particle clusters;

ICI Bucharest © Copyright 2012-2024. All rights reserved

- $\tilde{I}_c = \{\tilde{i}_{c_1}, \tilde{i}_{c_2}, \ldots, \tilde{i}_{c_{|\tilde{I}_c|}}\}$ represents the set of indices belonging to the c-th cluster, where $|\tilde{I}_c|$ is the total number of particles in the c-th cluster;
- $\mathbf{x}_i^*\left((t_{k-1}, h(1, |\tilde{I}_c|)\right)$, $\mathbf{x}^g(t_{k-1})$, $t_{k-1} \in T$ are the best (nondominated) potential decisions obtained by particles belonging to the same cluster as the given i-th particle $(i \in \tilde{I}_c)$ and chosen randomly in the range of $[1, |\tilde{I}_c|]$, and those obtained by all particles in the swarm, respectively;
- $\theta, \tilde{c}_1, \tilde{c}_2$ are constants, the values of which, as a rule, are set in the following ranges: $\theta \in [0.4, 1.4]$, $\tilde{c}_1 \in [1.5, 2]$, and $\tilde{c}_2 \in [2, 2.5]$;
- $q(0, 1)$, $e(0, 1)$ are random values uniformly distributed in the interval $[0, 1]$.

The boundary values for the feasible ranges of these constants were selected based on the recommendations from previous studies (Hassan et al., 2005).

At the same time, the values of decision variables are computed with the subsequent updating of the archive of nondominated solutions if the domination condition (20) is fulfilled:

$$\mathbf{x}_i(t_k) = \begin{cases} \begin{pmatrix} \mathbf{x}_i(t_{k-1}) \\ +\mathbf{v}_i(t_{k-1}) \end{pmatrix}, & \text{if } \begin{pmatrix} \mathbf{x}_i(t_{k-1}) \\ +\mathbf{v}_i(t_{k-1}) \end{pmatrix} \in [\underline{\mathbf{x}}, \overline{\mathbf{x}}], \\ \mathbf{x}_i(t_{k-1}), & \text{if } \begin{pmatrix} \mathbf{x}_i(t_{k-1}) \\ +\mathbf{v}_i(t_{k-1}) \end{pmatrix} \notin [\underline{\mathbf{x}}, \overline{\mathbf{x}}], \end{cases} \tag{20}$$

where $[\underline{\mathbf{x}}, \overline{\mathbf{x}}]$ are the feasible ranges of the decision variables $\mathbf{x}_i(t_k)$, $i \in I$, $t_k \in T$.

3. Results

Firstly, the developed algorithm (CBHPSO) was tested and evaluated to assess its performance.

In Table 1 the test instances (Zitzler et al., 2000) that were used for testing the CBHPSO are presented.

The number of decision variables varies in the range from 2 to 150 depending on the used test instance.

Table 1. Test instances for the CBHPSO

Test instances	Problem statement (objectives to be minimized)
FT1	$\begin{cases} f_1 = 4x^2 + 4y^2, \\ f_2 = (x-5)^2 + (y-5)^2 \end{cases};$ s.t. $\begin{cases} g_1 = (x-5)^2 + y^2 \le 25, \\ g_2 = (x-8)^2 + (y+3)^2 \ge 7.7 \end{cases};$ $0 \le x \le 5;$ $0 \le y \le 3$
FT2	$\begin{cases} f_1 = 1 - \exp\left(-\sum_{j=1}^{n}\left(x_j - n^{-1/2}\right)^2\right), \\ f_2 = 1 - \exp\left(-\sum_{j=1}^{n}\left(x_j + n^{-1/2}\right)^2\right) \end{cases};$ $-4 \le x_j \le 4$, $1 \le j \le 2$
FT3	$\begin{cases} f_1 = \sum_{j=1}^{2}\left(-10\exp\left(-0,2\sqrt{x_j^2 + x_{j+1}^2}\right)\right), \\ f_2 = \sum_{j=1}^{3}\left(\lvert x_j\rvert^{0,8} + 5x_j^3\right) \end{cases};$ $-5 \le x_j \le 5$, $1 \le j \le 3$
FT4	$\begin{cases} f_1 = 1 + (A_1 - B_1)^2 + (A_2 - B_2)^2, \\ f_2 = (x+3)^2 + (y+1)^2 \end{cases};$ where $\begin{cases} A_1 = 0,5\sin 1 - 2\cos 1 + \sin 2 - 1,5\cos 2, \\ A_2 = 1,5\sin 1 - \cos 1 + 2\sin 2 - 0,5\cos 2, \\ B_1 = 0,5\sin x - 2\cos x + \sin y - 1,5\cos y, \\ B_2 = 1,5\sin x - \cos x + 2\sin y - 0,5\cos y \end{cases};$ $-\pi \le x \le \pi$ $-\pi \le y \le \pi$
FT5	$\begin{cases} f_1 = x_1, \\ f_2 = gh \end{cases};$ where $\begin{cases} g = 1 + \frac{9}{29}\sum_{j=2}^{30} x_j, \\ h = 1 - \sqrt{f_1/g} - (f_1/g)\sin(10\pi f_1) \end{cases};$ $0 \le x_j \le 1$, $1 \le j \le 30$
FT6	$\begin{cases} f_1 = x_1, \\ f_2 = gh \end{cases};$ where $\begin{cases} g = 1491 + \sum_{j=2}^{150}\left(x_j^2 - 10\cos(4\pi x_j)\right), \\ h = 1 - \sqrt{x_1/g} \end{cases};$ $0 \le x_1 \le 1,\ -5 \le x_j \le 5,\ 2 \le j \le 150$

In Table 1, the following notations are used:

- FT1 – Binh and Korn function;
- FT2 – Fonseca-Fleming function;
- FT3 – Kursawe function;
- FT4 – Poloni's two objective function;
- FT5 – Zitzler-Deb-Thiele's function N3;
- FT4 – Zitzler-Deb-Thiele's function N4.

The results of optimization experiments conducted with the CBHPSO in comparison with other evolutionary algorithms are presented in Table 2.

The optimization experiments were carried out on a portable supercomputer DSWS PRO (2x Intel Xeon Silver 4114, 1x NVIDIA QUADRO RTX 6000) using 100 parallel processes of evolutionary search by means of particle swarm.

The following values of control parameters were used: total number of iterations: $|T| = 100$; population size: $|I| = 100$, and total number of clusters in the CBHPSO: $|C| = 10$.

Here, LHV is the logarithmic hypervolume, CPF is the Pareto front cardinality related to the number of obtained Pareto optimal solutions, PT (in seconds) is the processing time spent to obtain solutions.

As shown in Table 2, the CBHPSO (using clustering particles in a swarm) performs better than the BORCGA-BOPSO and other multi-objective genetic algorithms, in terms of LHV and CPF for most test instances.

The main advantage of the CBHPSO algorithm over other evolutionary algorithms is its ability to optimize important performance metrics such as LHV (logarithmic hypervolume) and CPF (the Pareto front cardinality).

Table 2. Evaluation of the performance metrics for CBHPSO and the other evolutionary algorithms employed

Performance metrics	**CBHPSO**	**Other evolutionary algorithms**				
		BORCGA-BOPSO	**SPEA2**	**NSGA-II**	**FCGA**	**MOPSO**
FT1 – Binh and Korn function						
LHV	3.8087	3.8081	3.8079	3.8079	3.8079	3.8075
CPF	2993	1635	754	3044	3063	842
PT, sec.	142.1	61.7	42.6	95.7	157.6	20.8
FT2 – Fonseca-Fleming function						
LHV	-0.0007	-0.0014	-0.0073	-0.0022	-0.0022	-0.0020
CPF	2007	1281	89	5415	5417	318
PT, sec.	157.0	49.1	12.3	143.9	276.6	24.8
FT3 –Kursawe function						
LHV	1.9092	1.9032	1.8497	1.9041	1.9033	1.8596
CPF	178	124	23	390	371	37
PT, sec.	126.0	34.6	13.1	33.9	50.0	24.1
FT4 – Poloni's two objective function						
LHV	2.7092	2.7063	2.7054	2.7058	2.7054	2.7062
CPF	161	109	78	79	82	104
PT, sec.	101.6	32.9	13.1	25.1	36.1	24.1
FT5 – Zitzler-Deb-Thiele's function N3						
LHV	0.20210	0.2018	0.2015	0.2017	0.2017	0.2010
CPF	260	260	260	260	260	256
PT, sec.	260.0	259.8	259.5	259.6	259.6	256.1
FT6 – Zitzler-Deb-Thiele's function N4						
LHV	1.59862	1.59843	1.59838	1.59841	1.59836	1.59185
CPF	1523	915	919	980	904	76
PT, sec.	216.0	97.0	139.0	201.0	259.0	3.0

ICI Bucharest © Copyright 2012-2024. All rights reserved

By maximizing these metrics, the proposed algorithm can help to find better solutions and obtain a larger number of trade-offs.

Although MOPSO and some genetic algorithms such as SPEA2 obtained a better value for time efficiency, the CBHPSO allows for a significant increase in the number of Pareto optimal solutions (CPF), while improving the quality of the Pareto front approximation (LHV). At the same time, an increase in processing time (PT can be observed for the search for optimal solutions with CBHPSO due to the time spent on clustering particles in the swarm (Table 2).

The value ranges for LHV, which was assessed as a result of repeated multiple optimization experiments for the considered algorithms and test instances, are shown in Figure 1.

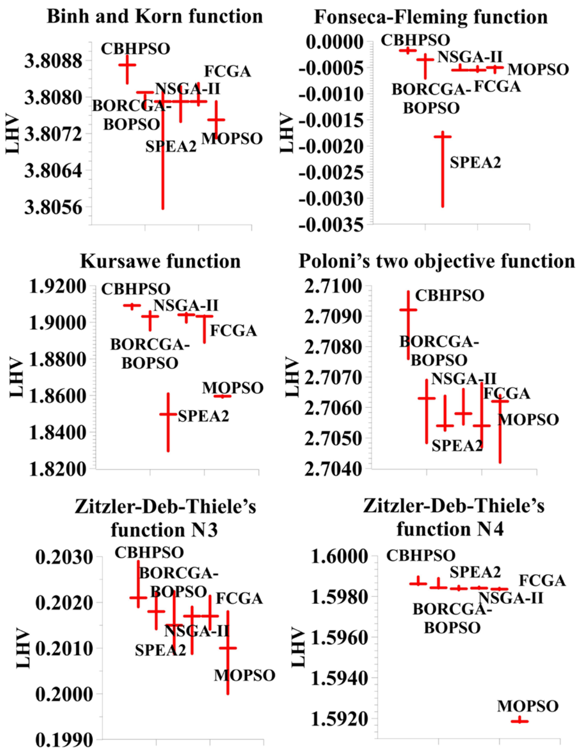

Figure 1. Value ranges for LHV

As shown in Figure 1, CBHPSO features a stable outperformance of other evolutionary algorithms with respect to the quality of the Pareto front approximation (LHV). Thus, the obtained results are statistically significant.

Figure 2 shows that the convergence rates for the normalized performance metrics (LHV and CPF) increase during the iterations of CBHPSO. The best values of these performance metrics correspond to a value of 1, while the worst values correspond to a value of 0 in Figure 2.

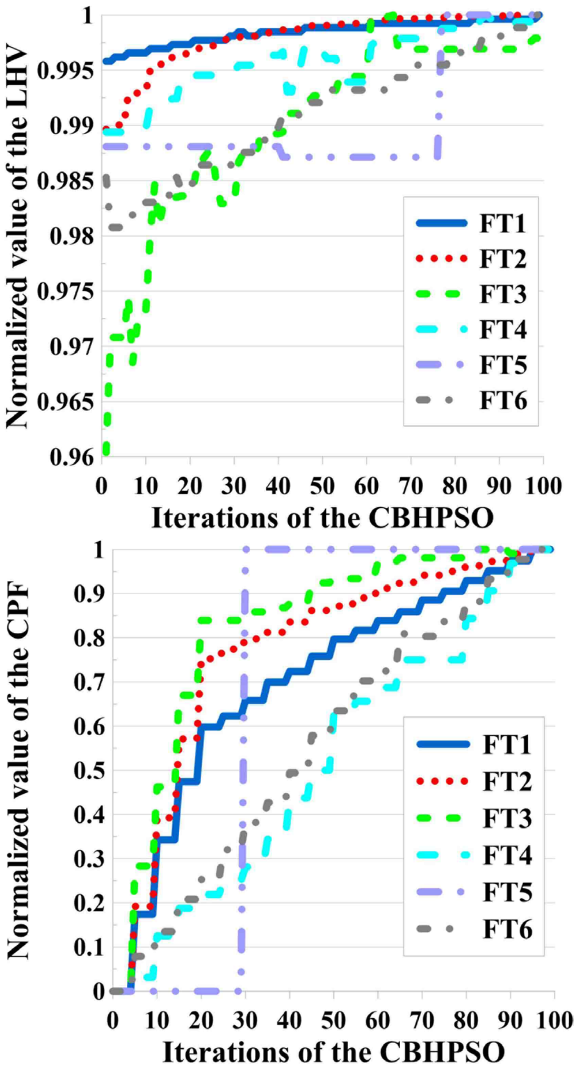

Figure 2. Convergence rates for normalized performance metrics

As it can be seen in Figure 2, the CBHPSO shows a stable convergence in terms of the growth rates of performance metrics values across different test instances.

The sensitivity tests for the normalized value of the LHV completed using the CBHPSO technique are shown in Figure 3.

Here, the best values of the LHV correspond to 1 and the worst values of the LHV correspond to 0.

Figure 3 shows that, as a rule, the quality of the Pareto front approximation (LHV) for the CBHPSO algorithm improves with an increase in the population size. At the same time, the total number of clusters in the swarm to be set

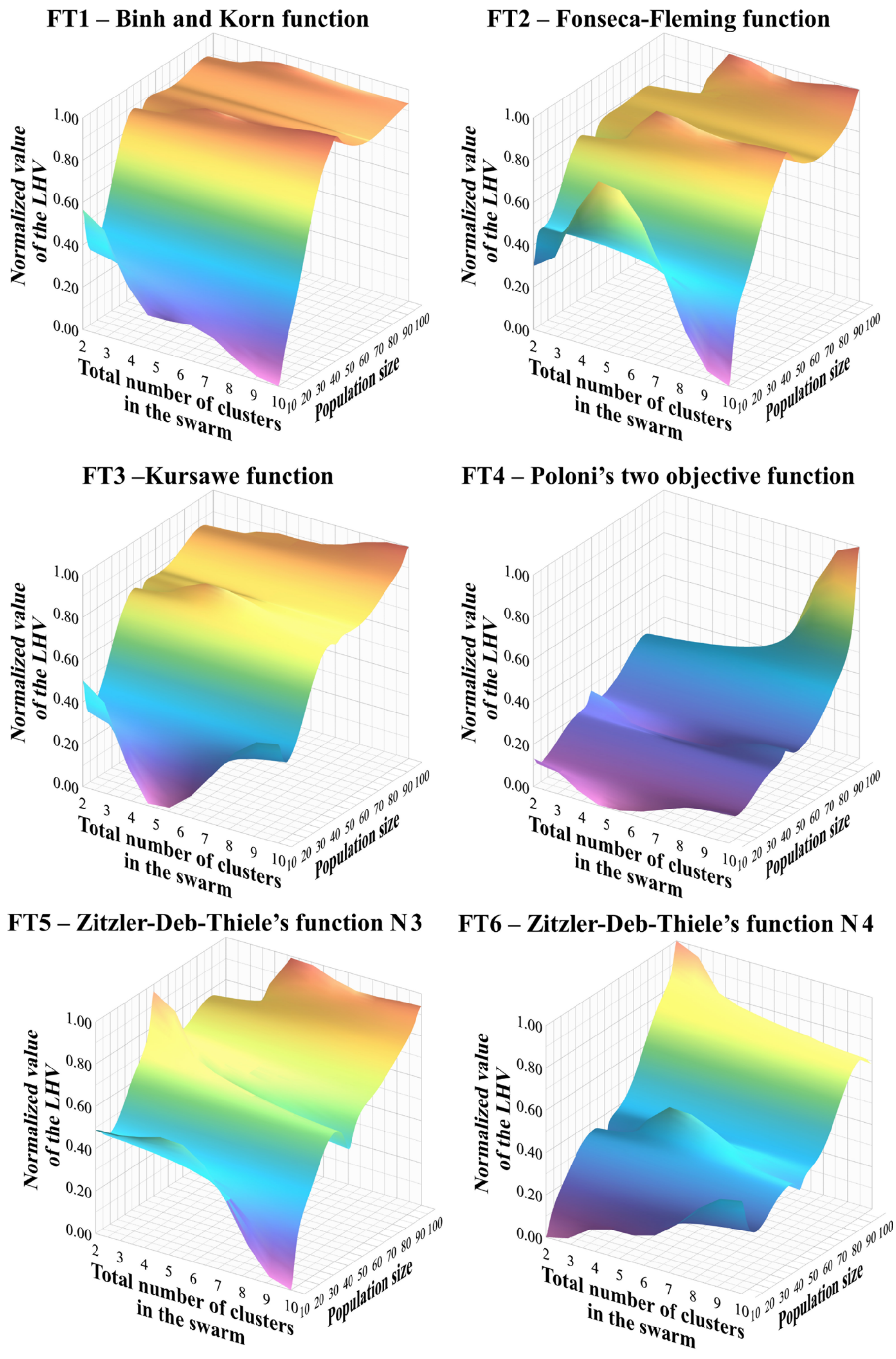

Figure 3. Sensitivity tests completed with the CBHPSO

up depends on the optimization problem to be solved. For minimizing the objectives of some test instances, for example, FT6, it is desirable to reduce the number of swarm clusters to a minimum.

This may be due to the fact that, when the large-scale optimization problem is solved (e.g. the number of decision variables is equal to 150 in this instance), multiple closely located Pareto optimal solutions are formed and the search areas can be in a smaller number of clusters. Sensitivity tests were also carried out for the CPF. By contrast to the LHV, the maximum value of CPF was achieved for the largest number of clusters in the swarm for all the analysed test instances (i.e. $|C| = 10$), because of the closely located solutions, irrespective of the increase in the value of CPF.

Further on, the CBHPSO algorithm was aggregated through objective functions with the employed multisectoral agent-based model

ICI Bucharest © Copyright 2012-2024. All rights reserved

of trade interactions, and it was applied to find trade-offs aimed at maximizing the values of these functions.

The proposed multisectoral agent-based model of trade interactions was developed based on the FLAME GPU 2 framework (Richmond et al., 2021), intended for large-scale agent-based modeling.

The allocation of agent enterprises among the high-technology and low-technology sectors of the economy is illustrated for three scenarios shown in Figure 4.

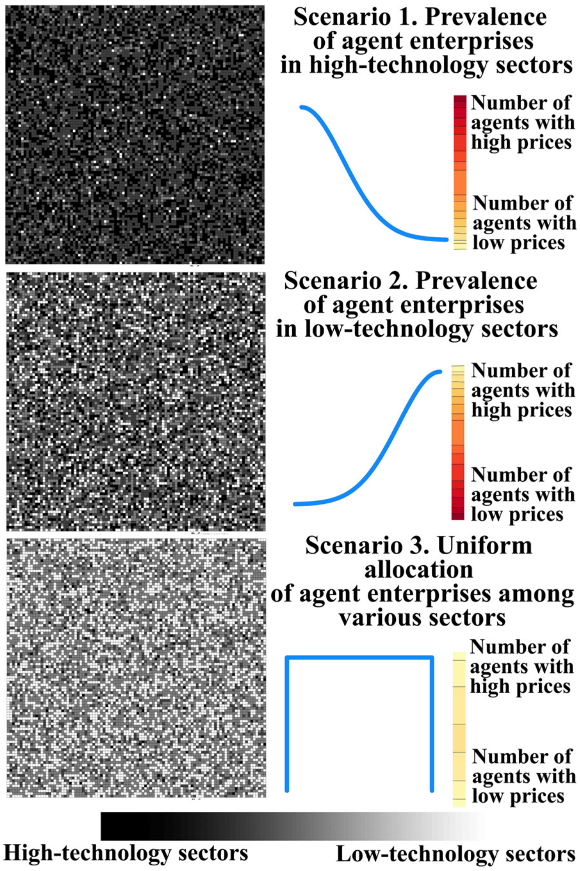

Figure 4. Scenarios for the allocation of agent enterprises among different economic sectors

As it is evident from Figure 4, the following scenarios for the allocation of agent enterprises among the sectors of the economy are considered:

- *Scenario 1*. Prevalence of agent enterprises in high-technology sectors. Most agents have high prices (i.e. added values) for their end products;

- *Scenario 2.* Prevalence of agent enterprises in low-technology sectors. Most agents have low prices for their end products;

- *Scenario 3.* Uniform allocation of agent enterprises among various sectors. The number of agents with high and low prices is the same.

Figure 5 shows the Pareto fronts computed with the CBHPSO algorithm in combination with the developed multisectoral agent-based model for various scenarios. The total number of agent enterprises allocated among the economic sectors is 10000, and the simulation period included 100 (days).

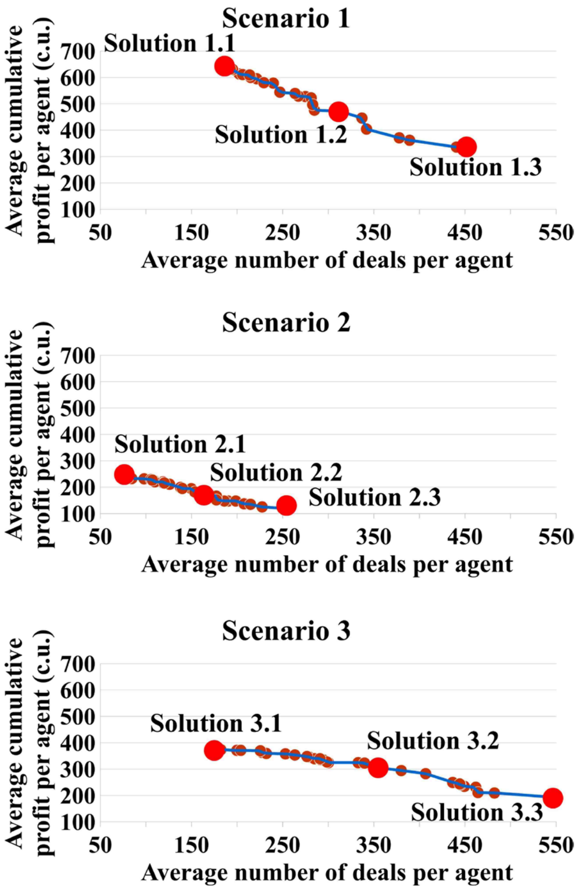

Figure 5. The Pareto fronts computed with the use of CBHPSO in combination with the multisectoral agent-based model

The results illustrated in Figure 5 were obtained by replicating the values of the objective functions multiple times during the evolutionary search in order to ensure their stability. Therefore, the simulation solutions shown in Figure 5 can be considered statistically significant.

Table 3 includes the optimal values of control parameters corresponding to the different optimal solutions for the Pareto fronts shown in Figure 5. As it is evident from Figure 5 and Table 3, the various optimal values of control parameters correspond to the different values of the solutions of the Pareto fronts.

Table 3. The optimal values of control parameters

Control parameters	Scenario 1		
	Solution 1.1	**Solution 1.2**	**Solution 1.3**
μ	0.647	0.277	0.823
σ	0.043	0.209	0.002
$\tilde{\mu}$	0.077	0.229	-0.365
$\tilde{\sigma}$	0.019	0.289	0.019
ρ	11	5	10
γ	0.478	0.532	0.963
η	1.352	1.212	1.001
	Scenario 2		
	Solution 2.1	**Solution 2.2**	**Solution 2.3**
μ	0.898	0.803	0.879
σ	0.239	0.193	0.099
$\tilde{\mu}$	0.524	-0.165	-0.946
$\tilde{\sigma}$	0.292	0.292	0.089
ρ	19	6	15
γ	0.352	0.699	0.865
η	1.347	1.180	0.975
	Scenario 3		
	Solution 3.1	**Solution 3.2**	**Solution 3.3**
μ	0.924	0.375	0.901
σ	0.185	0.158	0.024
$\tilde{\mu}$	0.705	0.644	-0.990
$\tilde{\sigma}$	0.239	0.377	0.238
ρ	19	4	17
γ	0.441	0.666	0.836
η	1.424	1.175	1.099

4. Discussion

In this paper, a new Clustering-Based Hybrid Particle Swarm Optimization (CBHPSO) algorithm was proposed. This algorithm outperforms other evolutionary multi-objective algorithms in terms of the quality of the solutions obtained (Table 2 and Figure 1). However, it has some limitations due to the time spent for clustering particles in the swarm. Therefore, other evolutionary algorithms such as SPEA2 and BORCGA-BOPSO obtained better processing time values (PT) in comparison with the CBHPSO. In the future, these time costs could be reduced thanks to the use of GPU-accelerated clustering algorithms (Li et al., 2023).

The CBHPSO algorithm has been integrated with a multisectoral agent-based model of trade interactions, and various scenarios of enterprise allocation among different economic sectors have been considered (Figure 4).

The Pareto fronts have been computed using the CBHPSO in combination with the multisector model for different scenarios (Figure 5).

When most agent enterprises are in high-technology sectors (the first scenario), the average cumulative profit per agent is high for most Pareto optimal solutions. The number of deals is significant because high-tech agents require a variety of intermediate products (Figure 5).

In the second scenario when most agent enterprises belong to low-technology sectors, both the average cumulative profit and the number of deals decrease due to lower prices for the final products of agents which have the simple structure of the intermediate products and a low added value.

For the uniform allocation of agent enterprises among various sectors (the third scenario), the total number of obtained solutions increases thereby raising the number of trade-offs which makes the values of the objective functions change in wide ranges.

As shown in Table 3, the maximization of the average cumulative profit per agent (Solutions 1.1, 2.1, and 3.1) can be noticed for the low values of γ and high values of η, which contributes to increasing the number of high-margin transactions between agents. By contrast, the relaxation of restrictions at the level of compliance of procurement with demand in a combination with a limitation on price increases leads to an increase in the average number of low-margin and ill-founded deals and to a decrease in the average cumulative profit per agent (Solutions 1.3, 2.3 and 3.3 in Table 3).

ICI Bucharest © Copyright 2012-2024. All rights reserved

The practical significance of the improvements in the values of the optimization metrics is as follows. The obtained nondominated solutions are close to the reference Pareto fronts, which allows to identify new more profitable scenarios for the control of economic agents' behavior. The total number of Pareto optimal solutions is significantly expanded, allowing for the more preferred ones to be selected from a wide range of possible scenarios.

5. Conclusion

This paper presents a new clustering-based hybrid particle swarm optimization algorithm (CBHPSO), which operates with subsets of particles in the criterion space using the k-means algorithm. Unlike other PSO-based evolutionary algorithms, CBHPSO iteratively executes the k-means algorithm, clustering particles in the swarm at each iteration in the criterion space. This allows the local archives of best solutions (i.e. nondominated solutions) to be expanded, and these archives can be used to update the velocities of the decision variables using inertia weights. The CBHPSO has advantages in terms of quality of the approximation of the Pareto front, and it outperforms other multiobjective PSO and genetic optimization algorithms. However, a decrease in time efficiency can be for the CBHPSO, which could be improved in future works by utilizing GPU-accelerated clustering algorithms.

The proposed algorithm (CBHPSO) has been applied to search for trade-offs in the employed multisectoral agent-based model of trade interactions, for various scenarios.

Future research could focus on the further development of this multisectoral agent-based model. This will be achieved by including new economic agents, such as households, banks, and transportation companies. These agents have specific behaviours that impact on trading interactions. An increase in the number of agents will also require an improvement in the efficiency of the evolutionary search process.

Acknowledgements

The research was supported by the Russian Science Foundation (RSF) (project No. 23-21-00012).

REFERENCES

Akopov, A. S. & Beklaryan, L. A. (2024) Traffic Improvement in Manhattan Road Networks With the Use of Parallel Hybrid Biobjective Genetic Algorithm. *IEEE Access*. 12, 19532-19552. doi: 10.1109/ACCESS.2024.3361399.

Akopov, A . S., Beklaryan, L. A. & Beklaryan, A. L. (2020) Cluster-Based Optimization of an Evacuation Process Using a Parallel Bi-Objective Real-Coded Genetic Algorithm. *Cybernetics and Information Technologies*. 20(3), 45-63. doi: 10.2478/cait-2020-0027.

Akopov, A. S., Beklaryan L. A. & Thakur, M. (2022) 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*. 23(8), 12648-12664. doi: 10.1109/TITS.2021.3115827.

Akopov, A. S., Beklaryan, A. L. & Zhukova, A. A. (2023) Optimization of Characteristics for a Stochastic Agent-Based Model of Goods Exchange with the Use of Parallel Hybrid Genetic Algorithm. *Cybernetics and Information Technologies*. 23(2), 87-104. doi: 10.2478/cait-2023-0015.

Balaji, P. G. & Srinivasan, D. (2010) An Introduction to Multi-Agent Systems. In: Srinivasan, D. & Jain, L.C. (eds.) *Innovations in Multi-Agent Systems and Applications - 1. Studies in Computational Intelligence. vol 310.* Berlin, Heidelberg, Springer, pp. 1-27.

Brems, H. (1957) Input-Output Coefficients as Measures of Product Quality. *The American Economic Review*. 47(1), 105-118.

Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*. 6(2), 182-197. doi: 10.1109/4235.996017

Demidova, L. A. & Gorchakov, A. V. (2022) Application of bioinspired global optimization algorithms to the improvement of the prediction accuracy of compact extreme learning machines. *Russian Technological Journal*. 10(2), 59-74. doi: 10.32362/2500-316X-2022-10-2-59-74.

Engda, E. W., Jin, G. G. & Son, Y.-D. (2023) Backstepping Control of a Magnetic Levitation System Using PSO. *Studies in Informatics and Control.* 32(3), 57-65. doi: 10.24846/v32i3y202305.

Gao, S., Gao, S., Pan, W. & Wang, M. (2021) Design of Improved PID Controller Based on PSO-GA Hybrid Optimization Algorithm in Vehicle Lateral Control. *Studies in Informatics and Control.* 30(4), 55-65. doi: 10.24846/v30i4y202105.

Gbenga, D. E. & Ramlan, E. K. (2016) Understanding the Limitations of Particle Swarm Algorithm for Dynamic Optimization Tasks: A Survey Towards the Singularity of PSO for Swarm Robotic Applications. *ACM Computing Surveys*. 49(1), Art. No. 8, 1-25, doi: 10.1145/2906150.

Gu, Y. & Wenzel, T. (2009) A note on the excess entry theorem in spatial models with elastic demand. *International Journal of Industrial Organization*. 27(5), 567-571. doi: 10.1016/j.ijindorg.2009.01.002.

Hassan, R., Cohanim, B., de Weck, O. & Venter, G. (2005) A Comparison of Particle Swarm Optimization and the Genetic Algorithm. In: *Proceedings of the 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, 18-21 April 2005, Austin, USA*. American Institute of Aeronautics and Astronautics, Inc. doi: 10.2514/6.2005-1897

Kennedy, J. & Eberhart, R. (1995) Particle Swarm Optimization. In: *Proceedings of IEEE International Conference on Neural Networks, ICNN'95, 27 November - 1 December 1995, Perth, Australia.* IEEE. pp. 1942-1948.

Li, M., Frank, E. & Pfahringer, B. (2023) Large scale K-means clustering using GPUs. *Data Mining and Knowledge Discovery.* 37, 67-109. doi: 10.1007/s10618-022-00869-6.

Li, H. & Wang, F. (2023) A Novel Trusted Intelligent Computing-aware Routing Service Offloading Method Based on MOPSO for Internet of Vehicles. *Studies in Informatics and Control*. 32(3), 109-118. doi: 10.24846/v32i3y202310.

Lloyd, S. (1982) Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2), 129-137. doi: 10.1109/TIT.1982.1056489.

van der Merwe, D. W. & Engelbrecht, A. P. (2003) Data clustering using particle swarm optimization. In: *Proceedings of the 2003 Congress on Evolutionary Computation, CEC '03.*, *8-12 December 2003*, *Canberra, Australia.* IEEE. pp. 215-220.

Miles, S., Yao, L., Meng, W., Black, C. M. & Ben-Miled, Z. (2022) Comparing PSO-based clustering over contextual vector embeddings to modern topic modelling. *Information Processing & Management*. 59(3), 102921. doi: 10.1016/j.ipm.2022.102921.

Molaei, S., Moazen, H., Najjar-Ghabel, S. & Farzinvash, L. (2021) Particle swarm optimization with an enhanced learning strategy and crossover operator. *Knowledge-Based Systems*. 215, 106768. doi: 10.1016/j.knosys.2021.106768.

Zitzler, E., Kalyanmoy, D. & Thiele, D. (2000) Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation.* 8(2), 173-195. doi: 10.1162/106365600568202.

Zitzler E., Laumanns M. & Thiele, L. (2001) *SPEA2: Improving the strength Pareto evolutionary algorithm*. Swiss Federal Institute of Technology, Zürich. TIK-Report: 103

Richmond, P., Chisholm, R., Heywood, P., Leach, M. & Kabiri Chimeh, M. (2021) *FLAME GPU.* https://zenodo.org/records/5465845 [Accessed 12th January 2024].

Xiaohui, H. & Eberhart, R. (2002) Multiobjective optimization using dynamic neighborhood particle swarm optimization. In: *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600), 12-17 May 2002, Honolulu, USA.* IEEE. pp. 1677-1681.

ICI Bucharest © Copyright 2012-2024. All rights reserved