INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGIES
BULGARIAN ACADEMY OF SCIENCES
CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **25**, No 4
Sofia • 2025 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2025-0033

A Hybrid Multi-Swarm Particle Swarm Optimization Algorithm for Solving Agent-Based Epidemiological Model

Andranik S. Akopov[1,2,3]
[1]*Central Economics and Mathematics Institute of the 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 paper presents a new agent-based epidemiological model, which is solved using the proposed Hybrid Multi-Swarm Particle Swarm Optimization Algorithm* (*HMSPSO Algorithm*)*. The HMSPSO is based on a combination of a parallel multi-swarm particle swarm optimization algorithm and real-coded genetic operators, including crossover and mutation. Unlike other well-known particle swarm optimization algorithms, this method uses alternating real-coded heuristic operators applied to parent solutions selected from sub-swarms obtained through agglomerative clustering. The performance of the HMSPSO Algorithm was compared to that of other established single-objective evolutionary algorithms, and the results show that the HMSPSO achieves the best performance in terms of both time efficiency and accuracy. HMSPSO was combined with the developed agent-based epidemiological model. As a result, optimal strategies for anti-epidemic measures such as vaccination intensity, self-quarantine intensity, and other parameters were calculated to maximize the share of surviving individuals.*

Keywords**:** *Particle swarm optimization, Epidemiological models, Multi-swarm optimization, Agent-based modeling, Genetic operators, Global optimization.*

1. Introduction

The study of the dynamics of epidemics, the prediction of their outcomes, and the development of effective strategies for implementing preventive and anti-epidemic measures are critical aspects of decision-making in healthcare. In recent years, challenges have arisen due to SARS-CoV-2, the COVID-19 pandemic, and other infectious diseases such as measles and different types of influenza viruses. At the same time, decision-makers in healthcare have a primary responsibility to minimize the impact of preventative and anti-epidemic measures to reduce the potential consequences of epidemics and maximize population survival. The complexity of addressing these issues stems from the need for a tailored approach that considers age distribution within the population, social factors, and resource limitations. Therefore,

the exploration of strategies to maximize the survival of a population during epidemics presents a complex, multidimensional, large-scale optimization challenge that necessitates the development of dedicated decision-making frameworks based on simulation modeling and heuristic optimization methods.

The search for optimal anti-epidemic strategies can be supported by the use of methods such as system dynamics [1-3], agent-based modeling [4, 5], genetic algorithms [6, 7], differential evolution and particle swarm optimization [8, 9], machine learning and clustering techniques [10, 11]. The seminal models in the field of epidemiology are the Susceptible-Infected-Recovered (SIR) models [1, 2]. These models employ systems of differential equations to forecast the rates of infection, recovery, and the overall survival of the population. In particular, modern SIR (susceptible-infected-recovered) models and their variations allow for the study of vaccination strategies based on age groups [3]. The primary limitations of these dynamic models can be attributed to their averaging and the impossibility of micro-level anti-epidemic control over individual humans or social groups. Therefore, new approaches are being developed that utilize agent-based modeling to study epidemiological processes. For instance, in [5], an agent-based model is proposed to assess strategies for human isolation during the spread of infectious diseases. The advantages of using agent-based modeling in epidemiology are enhanced when it is combined with evolutionary optimization methods such as Particle Swarm Optimization (PSO) [12-14] and Genetic Algorithms (GA) [15-17]. This is achieved through personalized optimization and anti-epidemic control at the individual level for each person. For example, optimal vaccination and self-quarantine strategies can be found within this approach for different age groups to maximize overall survival during an epidemic. The application of evolutionary search algorithms extends beyond solving epidemiological models. For instance, these algorithms are used to estimate the convexity of the objective function landscape during the extremum search process [18], for bi-level optimization of inventory and production [19], and for other purposes.

The aim of this paper is to develop a novel agent-based epidemic model and integrate it with a proposed Hybrid Multi-Swarm Particle Swarm Optimization Algorithm (HMSPSO Algorithm) to identify optimal strategies for epidemic control and prevention measures that maximize the share of survival rate of infected individuals. The remainder of this paper is organized as follows: Section 2 introduces the developed agent-based epidemiological model. Section 3 considers the HMSPSO Algorithm proposed in this study. Section 4 presents the results of optimization experiments, and finally. Section 5 discusses the main advantages and limitations of the approach and concludes this paper.

2. Agent-based epidemic model

The model concept is based on early, justified principles that emphasize the importance of implementing individual anti-epidemic and preventive measures, such as vaccination, self-quarantine, and hospitalization, taking into account different age groups and the specifics of people's spatial locations [3-5]. The proposed agent-based

model simulates the spread of an epidemic using a well-known cellular automata approach. It considers various scenarios for the initial proportion of infected individuals and their spatial distribution, as illustrated in Fig. 1.

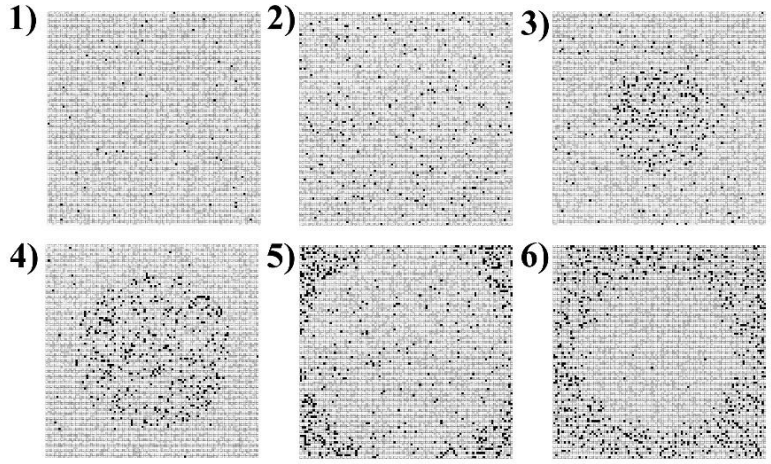

Fig. 1. Illustration of various scenarios for the initial proportion of infected individuals and their spatial distribution

In Fig. 1, the following scenarios are presented: 1) a minimum proportion of infected individuals (0.1%), which are uniformly distributed throughout the model space; 2) a small proportion of infected individuals (0.2%), which are uniformly distributed throughout the model space; 3) an average proportion of infected individuals (0.3%), which are located inside a circle with a radius of R; 4) an average proportion of infected individuals (0.5%), which are located inside a circle with a radius of $1.5R$; 5) a significant proportion of infected individuals (0.75%), which are located outside a circle with a radius of $2R$; 6) all individuals are infected and located outside a circle with a radius of $1.5R$. As shown in Fig. 1, the infected agents are indicated by black dots. The model space is a finite-dimensional lattice with a dimension of 100×100 cells. Each individual, if alive, can be in one of the following possible states: healthy, infected, vaccinated, self-quarantined, or hospitalized, as shown in Fig. 2.

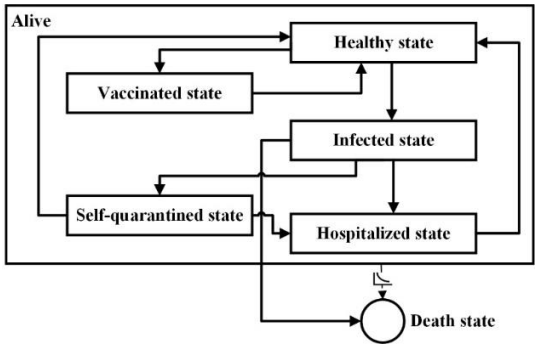

Fig. 2. Flowchart of potential states for an individual within the epidemiological model

Transitions between different states of the agent take place according to specific rules, similar to those found in a finite-state machine model (Fig. 2). In particular, an individual's initial state can be either healthy or infected. A healthy individual can become vaccinated if they have not previously been vaccinated. At the same time, the probability of this transition depends on the vaccination intensity, which is a control parameter in the model. Vaccination can significantly reduce the rate of infection, although it does not eliminate it. Therefore, a vaccinated individual can still become infected. An individual can transition from a healthy to the infected state when the level of infection surpasses a certain threshold value. When an individual becomes infected, they can serve as a source of transmission to others. Therefore, infection of a healthy person occurs through contact with an infected person or persons, and this depends on the duration of such contact. In order for this to happen, healthy and infected individuals need to be within a specific distance given by the radius length between them. At the same time, a person who is infected and is hospitalized or self-quarantined is not a source of infection. An infected individual can transition to these two states due to the exceeding of certain infection thresholds. In the first case, the threshold value is lower (self-quarantine), but the transition to this state is also controlled and carried out at a given rate of self-quarantine intensity. This rate reflects the capacity of the healthcare system to detect the presence of disease in individuals through timely testing. During self-quarantine, an individual recovers at a slower rate compared to being hospitalized. The transition back to a healthy state occurs after a certain period of time, such as 7 days, through timeout. If a second threshold value is exceeded, an individual may transition to the hospitalized state. In this state, the recovery process provides significantly higher rates, and the transition of a person to a healthy state occurs no earlier than 3 days later, for example. The transition to the hospitalized state is also controlled and can be carried out with a given rate of hospitalization intensity. At the same time, transitions to self-quarantined and hospitalized states are only implemented if there are available residential spaces for self-quarantine and hospital spaces, respectively. These are the resource characteristics of the model and its control parameters.

The following is an abstract description of the agent-based epidemiological model that has been developed.

Here,

- $T = \{t_0, t_1 \ldots, t_{|T|}\}$ 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, and are all moments;
- $J = \{j_1, j_2, \ldots, j_{|J|}\}$ is the set of indices of age groups for people (e.g., 0-10 years old, 11-20 years old, etc.), where $|J|$ is the total number of age groups;
- $I_j = \{i_{j1}, i_{j2}, \ldots, i_{|jI|}\}, j \in J$ is the set of indices of agents of the j-th age group, where $|I_j|$ is the total number of agents of the j-th age group;

- $s_{i_j}(t_k) \in \{1,\ 2,\ ...,\ 6\},\ i_j \in I_j,\ j \in J,\ t_k \in T,$ is the state of the i_j-th agent of the j-th age group at the moment t_k: $s_{i_j}(t_k) = 1$ means the healthy state, $s_{i_j}(t_k) = 2$ means the infected state, $s_{i_j}(t_k) = 3$ means the hospitalized state, $s_{i_j}(t_k) = 4$ means the vaccinated state, $s_{i_j}(t_k) = 5$ means the self-quarantined state, $s_{i_j}(t_k) = 6$ means the death state;
- $\tau_{i_j}(t_k)$, $\{\underline{\tau}_3,\ \underline{\tau}_5\}$ are the staying time of the i_j-th agent of the j-th age group in the hospital or self-quarantine state, and the given time limits for staying there (in days), respectively (e.g., $\underline{\tau}_3 = 3$ and $\underline{\tau}_5 = 7$);
- $\chi_{i_j} \in \{0,\ 1\},\ i_j \in I_j,\ j \in J,$ is the feature that reflects the fact that the i_j-th agent of the j-th age group agent has not been vaccinated $(\chi_{i_j} = 0)$ or has been vaccinated $(\chi_{i_j} = 1)$;
- $\alpha_{i_j}(t_k),\ i_j \in I_j,\ j \in J,\ t_k \in T,$ is the variable that reflects the infection level of the i_j-th agent of the j-th age group at the moment t_k, $\alpha_{i_j}(t_0) > 1$ with the given probability of $p_0 \in [0,\ 1]$ and $\alpha_{i_j}(t_0) = 0$ in a common case. The value of the variable at the moment t_k $(t_k \in T,\ k = 1,\ 2,\ ...\ K)$ can be calculated as follows:

$$(1) \qquad \alpha_{i_j}(t_k) = \begin{cases} \alpha_{i_j}(t_{k-1}) + \Theta(\chi_{i_j}, j) \text{ if } s_{i_j}(t_{k-1}) \in \{1,\ 4\} \text{ and } d_{i_j\tilde{i}_j}(t_{k-1}) \le \underline{d}, \\ \alpha_{i_j}(t_{k-1}) + \theta_1 \text{ if } s_{i_j}(t_{k-1}) = 2, \\ \alpha_{i_j}(t_{k-1}) - \theta_2 \text{ if } s_{i_j}(t_{k-1}) = 3, \\ \alpha_{i_j}(t_{k-1}) - \theta_3 \text{ if } s_{i_j}(t_{k-1}) = 5, \end{cases}$$

where $\Theta(\chi_{i_j}, j),\ i_j \in I_j,\ j \in J,$ is the functional dependence of the infection rate on the vaccination status and the age group of the agent; $d_{i_j\tilde{i}_j}(t_{k-1}),\ i_j, \tilde{i}_j \in I_j,\ j \in J$ is the distance between the i_j-th healthy agent of the j-th age group and the $\tilde{i}_j$-th infected agent; $\underline{d}$ is the given threshold distance for infection spread; $\theta_1,\ \theta_2,\ \theta_3$ are known as fixed rates of change in the infection rate when an agent is without treatment or with treatment in a hospital or at home $(\theta_1, \theta_2 = 0.1,\ \theta_3 = 0.01)$;
- $\delta_1,\ \delta_2,\ \delta_3$ are the given threshold values determining conditions for the transition to states of infection, self-quarantine, etc., $\delta_1 < \delta_2 < \delta_3$;
- $\{r_{1j},\ r_{2j},\ r_{3j}\} \in [0,\ 1],\ j \in J,$ are the intensities (i.e., the number of events per one day) of vaccination, self-quarantine, and hospitalization for the j-th age group at the moment t_k;

- $\{\tau_{1j}(r_{1j}),\ \tau_{2j}(r_{2j}),\ \tau_{3j}(r_{3j})\} \in T,\ j \in J,$ are the time moments for the implementation of vaccination, self-quarantine, and hospitalization, which are computed based on the given intensities for the j-th age group;
- $h(0,1),\ \tilde{h}(0,1),$ are the random values that are uniformly distributed in the range of 0 to 1;
- $\tilde{p} \in [0,\ 1]$ is the given possibility of a sudden decline in health during self-quarantine;
- $\{\lambda_1(t_k),\ \lambda_2(t_k)\},\ t_k \in T,$ are the known numbers of hospitalized and self-guaranteed agents, respectively, at the moment t_k;
- $\{H(\rho_1,\ \lambda_1(t_k)),\ Q(\rho_2,\ \lambda_2(t_k))\},\ t_k \in T,$ are the numbers of hospital spaces and residential spaces (i.e., cells) for self-quarantine, which depend on control parameters ρ_1 and ρ_2, where $\rho_1 \in (\bar{\rho},\ 1],\ \rho_2 \in [\underline{\rho},\ \bar{\rho}]$ These are the control parameters that determine the share of cells in the discrete model space that are used as hospital spaces and self-quarantine spaces, respectively, where $1-\bar{\rho}$ is the share of hospital spaces and $\bar{\rho}-\underline{\rho}$ is the share of self-quarantine spaces;

The rules of transition to possible states of the i_j-th agent $(i_j \in I_j,\ j \in J)$ at moment t_k $(t_k \in T,\ k = 1,\ 2,\ \ldots K)$ can be given as follows:

(2) $$s_{i_j}(t_k) = \begin{cases} 1 \text{ if I is true,} \\ 2 \text{ if II is true,} \\ 3 \text{ if III is true,} \\ 4 \text{ if IV is true,} \\ 5 \text{ if V is true,} \\ 6 \text{ if VI is true,} \end{cases}$$

where

I. $(t_k = t_0 \text{ and } h(0,1) \geq p_0)$ or $\left(s_{i_j}(t_{k-1}) = 5 \text{ and } \tau_{i_j}(t_{k-1}) > \underline{\tau}_5\right)$ or

$$\left(s_{i_j}(t_{k-1}) = 3 \text{ and } \tau_{i_j}(t_{k-1}) > \underline{\tau}_3 \text{ and } \alpha_{i_j}(t_{k-1}) \leq \delta_1\right),$$

that means the i_j-th agent $(i_j \in I_j,\ j \in J)$ is in the initial state and is **healthy**, or it was in a self-quarantined or hospitalized state at the previous moment, and its stay time has exceeded the appropriate time limit, taking into account the infection level;

II. $(t_k = t_0 \text{ and } h(0,1) < p_0)$ or $\left(s_{i_j}(t_{k-1}) \in \{1,\ 4\} \text{ and } \alpha_{i_j}(t_{k-1}) > \delta_1\right),$

that means the i_j-th agent $(i_j \in I_j,\ j \in J)$ is in the initial state and is **infected**, or it was in a healthy or vaccinated state at the previous moment, and its infection level has exceeded the first threshold value;

III. $\left.\begin{pmatrix} s_{i_j}(t_{k-1}) = 2 \text{ and } \alpha_{i_j}(t_{k-1}) > \delta_2 \text{ and } t_{k-1} = \tau_{3j}(r_{3j}) \\ \text{and } H\left(\underline{\rho}_1,\ \lambda_1(t_{k-1})\right) > 0 \end{pmatrix}\right.$ or

$\left(s_{i_j}(t_{k-1}) = 5 \text{ and } \alpha_{i_j}(t_{k-1}) > \delta_2 \text{ and } \tilde{h}(0,1) < \tilde{p} \text{ and } H\left(\underline{\rho}_1,\ \lambda_1(t_{k-1})\right) > 0\right)$,

that means the i_j-th agent $(i_j \in I_j,\ j \in J)$ was in an infected or self-quarantined state at the previous moment, and its infection level has exceeded the second threshold value, and a hospital space was available, taking into account the **hospitalization intensity** r_{3j} as a control parameter;

IV. $\left(s_{i_j}(t_{k-1}) = 1 \text{ and } t_{k-1} = \tau_{1j}(r_{1j}) \text{ and } \chi_{i_j} = 0\right)$,

that means the i_j-th agent $(i_j \in I_j,\ j \in J)$ was in a healthy state at the previous moment, and the agent had not been vaccinated earlier, taking into account the **vaccination intensity** r_{1j} as a control parameter;

V. $\left.\begin{pmatrix} s_{i_j}(t_{k-1}) = 2 \text{ and } \alpha_{i_j}(t_{k-1}) \le \delta_2 \text{ and } t_{k-1} = \tau_{2j}(r_{2j}) \\ \text{and } \mathrm{Q}\left(\rho_2,\ \lambda_2(t_{k-1})\right) > 0 \end{pmatrix}\right.$,

that means the i_j-th agent $(i_j \in I_j,\ j \in J)$ was in an infected state at the previous moment, and its infection level had not exceeded the second threshold value, and a hospital space was available, taking into account the **self-quarantine intensity** r_{2j} as a control parameter;

VI. $\alpha_{i_j}(t_{k-1}) > \delta_3$,

that means the infection level of the i_j-th agent $(i_j \in I_j,\ j \in J)$ had exceeded the third threshold value, and as a result, the agent has **died**.

The share of surviving agents at a finite moment $|T|$ can be determined as follows:

$$S = \frac{A}{\sum_{j=1}^{|J|} |I_j|}, \tag{3}$$

where

$$A = \sum_{t_k=1}^{|T|} \sum_{j=1}^{|J|} \sum_{i_j=1}^{|I_j|} m_{i_j}(t_k), \tag{4}$$

$$m_{i_j}(t_k) = \begin{cases} 1 \text{ if } s_{i_j}(t_{k-1}) \neq 6, \\ 0 \text{ if } s_{i_j}(t_{k-1}) = 6. \end{cases} \tag{5}$$

In the studied multi-agent model, the decision variables $\{r_{1j},\ r_{2j},\ r_{3j}\}_{j=1}^{|J|}$ and $\{\rho_1,\ \rho_2\}$ are control parameters, and the values of these parameters are set at the initial moment. Therefore, a decision maker aims to set optimal values for control

variables that maximize the share of surviving agents by moment $|T|$. As a result, the following optimization problem can be formulated.

Problem A. The need to maximize the share of surviving agent **individuals** through the sets of control parameters that determine the intensities of vaccinations, self-quarantine, and hospitalization $\{r_{1j},\ r_{2j},\ r_{3j}\}_{j=1}^{|J|}$ and environmental characteristics $\{\rho_1,\ \rho_2\}$ that influence the available number of self-quarantine and hospital spaces:

$$\max_{\{r_{1j},\, r_{2j},\, r_{3j}\}_{j=1}^{|J|},\ \{\rho_1,\, \rho_2\}} S, \tag{6}$$

s.t.

$$\{r_{1j},\ r_{2j},\ r_{3j}\} \in [0,\ 1],\ j \in J,\ \rho_1 \in (\bar{\rho},\ 1],\ \rho_2 \in [\underline{\rho},\ \bar{\rho}].$$

Problem A pertains to a specific category of simulation-based derivative-free optimization problems [16, 20]. In such systems, the values of objective functions are calculated as the result of simulation modelling. At the same time, a significant requirement for evolutionary algorithms used in solving derivative-free optimization problems based on simulations is their high level of time efficiency. This efficiency is greatly influenced by the overall number of recalculations of objective function values. The agent-based model discussed in this paper is implemented within the AnyLogic framework [21], and thus, the time taken to calculate the objective function value is influenced by both the performance of this simulation tool and the overall number of interacting agents in the model. Therefore, the Dynamic Multi-Swarm Particle Swarm (DMS-PSO) Algorithm [22] was selected as the basis for developing an evolutionary algorithm that potentially provides a solution to Problem A with high time efficiency.

3. A Hybrid Particle Multi-Swarm Optimization Algorithm

HMSPSO is an optimization technique that aims to iteratively improve a given solution with respect to a specified criterion, such as the value of the objective function in the case of a single-objective optimization problem. The method employs a population of potential solutions, known as particles, that are moved through a search space based on simple mathematical formulas derived from the particles' current positions and velocities. The advantage of well-known PSO and Multi-Swarm PSO (MSPSO) algorithms is their time efficiency [12, 13], which is superior to that of standard genetic optimization algorithms that use resource-intensive heuristic operations such as selection, crossover, mutation, and updating individuals in populations [15, 16]. However, one significant disadvantage of swarm-based optimization techniques is that they often get stuck in local extrema, which can lead to issues with premature convergence and skipping optimal solutions [17, 18]. Therefore, in modern times, various approaches to the hybridization of swarm-based optimization algorithms are actively explored. These include well-known PSO with crossover, PSO mutation, as well as RSGA-PSO [17]. However, these approaches have not yet addressed the combined use of multi-swarm PSO algorithms, alternating real-coded heuristic operators, and clustering techniques. In order to address this gap

and ensure the efficient and accurate solution of the developed agent-based epidemiological model (Problem A), the HMSPSO Algorithm has been developed.

The following is a brief description of the proposed HMSPSO Algorithm.

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

$$\min F(\mathbf{x}), \quad \text{s.t.} \quad \mathbf{x} = (x_1, x_2, ..., x_m)^{\mathrm{T}} \in \Omega, \tag{7}$$

where is a decision variable vector with dimension m, $\Omega = \prod_{i=1}^{m} [a_i, b_i]$ is the feasible region of the search space ($i = 1, 2, ..., m$ is the index of decision variables) and $F(\mathbf{x})$ is the objective function, which is computed as the result of the simulation modeling.

At the same time, the previously stated optimization problem of maximizing the number of surviving agents (Problem A) can be reduced to an equivalent minimization problem by simply replacing the objective function with $1/S$.

In the developed HMSPSO Algorithm, particles are implemented in the agent processes whose functionality is implemented using a parallel Graphics Processing Unit architecture. Therefore, the occurrence of multiple swarms is due to the presence of a set of agent processes that have their own swarms.

Here,

- $T = \{t_0, t_1 ..., t_{|T|}\}$ is the set of iterations in the HMSPSO algorithm, $|T|$ is the total number of iterations, $t_0 \in T$, $t_{|T|} \in T$ are the initial and final iterations of the HMSPSO Algorithm, $t_k \in T$, $k = 1, 2, ... K,$ is all iterations;
- $N = \{n_1, n_2, ..., n_{|N|}\}$ is the set of indices of the parallel agent processes in the HMSPSO, where $|N|$ is the total number of agent-processes;
- $J_n = \{j_{n1}, j_{n2}, ..., j_{|nJ|}\}$, $n \in N,$ is the set of indices of particles of the n-th agent-process, where $|nJ|$ is the total number of particles;
- $I = \{i_1, i_2, ..., i_{|I|}\}$ is the set of indices of decision variables, where $|I|$ is the total number of decision variables;
- $\mathbf{x}_{j_n}(t_k) = \left\{x_{j_n i_1}(t_k), x_{j_n i_2}(t_k), ..., x_{|j_n I|}(t_k)\right\}$, $j_n \in J$, $n \in N$, $i \in I$, $t_k \in T,$ is the vector of values of the decision variables of the j_n-th particle of the n-th agent-process at iteration t_k, where $|j_n I|$ is the total number of decision variable values;
- $\underline{\mathbf{x}}$, $\overline{\mathbf{x}}$ are the vectors for the values of the lower limits and upper limits, respectively, of decision variables;

- $\mathbf{x}^{*}_{j_n}(t_k),\ j_n \in J_n,\ n \in N,\ t_k \in T,$ is the best local decision found by the j_n-th particle of the n-th agent-process during the whole search period by moment t_k;
- $\mathbf{x}^{g}_{n}(t_k),\ n \in N,\ t_k \in T,$ is the best global decision found by all particles of the n-th agent-process at iteration t_k;
- $\tilde{\mathbf{x}}^{g}(t_k),\ t_k \in T,$ is the best global decision found by all particles across all the agent processes at iteration t_k;
- $h(0,\ 1),\ q(0,\ 1),\ e(0,\ 1)$ are random values uniformly distributed within the interval $[0,\ 1]$;
- $\theta,\ c_1,\ c_2,\ c_3$ are constants, the values of which, as a rule, are set in the following ranges: $\theta \in [0.5,\ 1.5],\ c_1 \in [1.5,\ 2],\ c_2, c_3 \in [2,\ 2.5]$.

The velocity vector for the decision variables of the j_n-th particle $(j_n \in J_n, n \in N)$ of the n-th agent-process $(n \in N)$ is calculated, determining the position of the particle in the space of potential solutions at iteration t_k, $(t_k \in T,\ k = 1,\ 2,\ \ldots\ K)$:

$$\begin{aligned}\mathbf{v}_{j_n}(t_k) = \theta \mathbf{v}_{j_n}(t_{k-1}) &+ c_1 h(0,\ 1)\left(\mathbf{x}^{*}_{j_n}(t_{k-1}) - \mathbf{x}_{j_n}(t_{k-1})\right) + \\ &+ c_2 q(0,\ 1)\left(\mathbf{x}^{g}_{n}(t_{k-1}) - \mathbf{x}_{j_n}(t_{k-1})\right) + \\ &+ c_3 e(0,\ 1)\left(\tilde{\mathbf{x}}^{g}(t_{k-1}) - \mathbf{x}_{j_n}(t_{k-1})\right).\end{aligned} \tag{8}$$

Then, the decision variables vector of the j_n-th particle $(j_n \in J_n, n \in N)$ of the n-th agent-process $(n \in N)$ at iteration t_k $(t_k \in T,\ k = 1,\ 2,\ \ldots\ K)$ is calculated as follows:

$$\mathbf{x}_{j_n}(t_k) = \begin{cases} \mathbf{x}_{j_n}(t_{k-1}) + \mathbf{v}_{j_n}(t_{k-1}) \text{ if } \mathbf{x}_{j_n}(t_{k-1}) + \mathbf{v}_{j_n}(t_{k-1}) \in [\underline{\mathbf{x}},\ \overline{\mathbf{x}}], \\ \mathbf{x}_{j_n}(t_{k-1}) \text{ if } \mathbf{x}_{j_n}(t_{k-1}) + \mathbf{v}_{j_n}(t_{k-1}) \notin [\underline{\mathbf{x}},\ \overline{\mathbf{x}}]. \end{cases} \tag{9}$$

In each iteration of the HMSPSO algorithm, the solutions found by the particles of different swarms are divided into groups using an agglomerative hierarchical clustering technique [23]. These subgroups are then used to select a pair of parent solutions located close to the centroids of appropriate clusters in order to include them in a crossover operator to obtain offspring solutions. The use of crossover and mutation as heuristic operators allows overcoming the issue of premature convergence by introducing new solutions into swarms instead of their worst solutions, as in genetic algorithms. The identification of the most promising clusters of particles to select a pair of parent solutions can be done randomly or based on outbreeding. In other words, the centers of these groups should be as far apart from each other as possible.

To calculate the distance matrix D, agglomerative clustering can use the conventional Euclidean distance [23], calculated using the widely accepted formula,

as a metric for measuring the proximity between neighboring solutions of the j_n-th and $\tilde{j}_n$-th particles $(j_n, \tilde{j}_n \in J_n,\ j_n \neq \tilde{j}_n,\ n \in N)$ at iteration t_k, $(t_k \in T,\ k = 1,\ 2,\ \ldots K)$:

$$D = \begin{pmatrix} d_{11} & \cdots & d_{1|nJ|} \\ \vdots & \ddots & \vdots \\ d_{|nJ|1} & \cdots & d_{|nJ||nJ|} \end{pmatrix}, \text{ where } d_{j_n \tilde{j}_n} = \sqrt{\left(f_{j_n}\left(\mathbf{x}_{j_n}(t_{k-1})\right) - \tilde{f}_{\tilde{j}_n}\left(\mathbf{x}_{j_n}(t_{k-1})\right)\right)^2}, \tag{10}$$

where $f_{j_n}(t_{k-1}), \tilde{f}_{\tilde{j}_n}(t_{k-1})$ are the particular solutions of the j_n-th and $\tilde{j}_n$-th particles $(j_n, \tilde{j}_n \in J_n,\ j_n \neq \tilde{j}_n,\ n \in N)$ at iteration t_k $(t_k \in T,\ k = 1,\ 2,\ \ldots K)$.

An important feature of the HMSPSO algorithm is the alternating crossover operator applied to each element of the decision-variable vector for a pair of parent solutions. $\left\{\hat{\mathbf{x}}_{1,j_n}(t_k),\ \hat{\mathbf{x}}_{2,j_n}(t_k)\right\}$ previously selected from clusters at iteration t_k $(t_k \in T,\ k = 1,\ 2,\ \ldots K)$ to obtain offspring solutions with the following updated values of the decision variables:

$$\breve{\mathbf{x}}_{1,2,j_n}(t_k) = \begin{cases} \text{SBX}\left(\hat{\mathbf{x}}_{1,j_n}(t_{k-1}),\ \hat{\mathbf{x}}_{2,j_n}(t_{k-1})\right) \text{ if VII is true,} \\ \text{LX}\left(\hat{\mathbf{x}}_{1,j_n}(t_{k-1}),\ \hat{\mathbf{x}}_{2,j_n}(t_{k-1})\right) \text{ if VIII is true,} \\ \text{MSBX}\left(\hat{\mathbf{x}}_{1,j_n}(t_{k-1}),\ \hat{\mathbf{x}}_{2,j_n}(t_{k-1})\right) \text{ if IX is true,} \\ \text{DMSBX}\left(\hat{\mathbf{x}}_{1,j_n}(t_{k-1}),\ \hat{\mathbf{x}}_{2,j_n}(t_{k-1})\right) \text{ if X is true,} \end{cases} \tag{11}$$

where:

VII. $h(0,\ 1) \le p_1$ and $t_{k-1} - \omega\left\lfloor \frac{t_{k-1}}{\omega} \right\rfloor \neq 0$,

VIII. $p_1 < h(0,\ 1) \le p_2$ and $t_{k-1} - \omega\left\lfloor \frac{t_{k-1}}{\omega} \right\rfloor \neq 0$,

IX. $p_2 < h(0,\ 1) \le p_3$ and $t_{k-1} - \omega\left\lfloor \frac{t_{k-1}}{\omega} \right\rfloor \neq 0$,

X. $p_3 < h(0,\ 1) \le p_4$ and $t_{k-1} - \omega\left\lfloor \frac{t_{k-1}}{\omega} \right\rfloor \neq 0$.

Here:

- $p_1, p_2, \ldots, p_4$ are the threshold values used for applying the real-coded crossover operator with a given probability.
- SBX, LX, MSBX, and DMSBX are the Simulated Binary Crossover (SBX), Laplace Crossover (LX), Modified Simulated Binary Crossover (MSBX), and Discrete Modified Simulated Binary Crossover (DMSBX), as genetic operators, respectively [15, 16];

- $h(0, 1)$ is the random number that is uniformly distributed in the range $[0, 1]$;
- $\omega \in T$ is the frequency of completing a crossover operator (e.g., $\omega \in [5, 10]$ when $|T| = 100$).
- $\lfloor . \rfloor$ is the integer part of the number.

After the crossover operator has been applied to the resulting offspring solutions, a real-coded mutation operator such as Uniform Mutation (UM) or Power Mutation (PM), or another operator is then applied [15, 16].

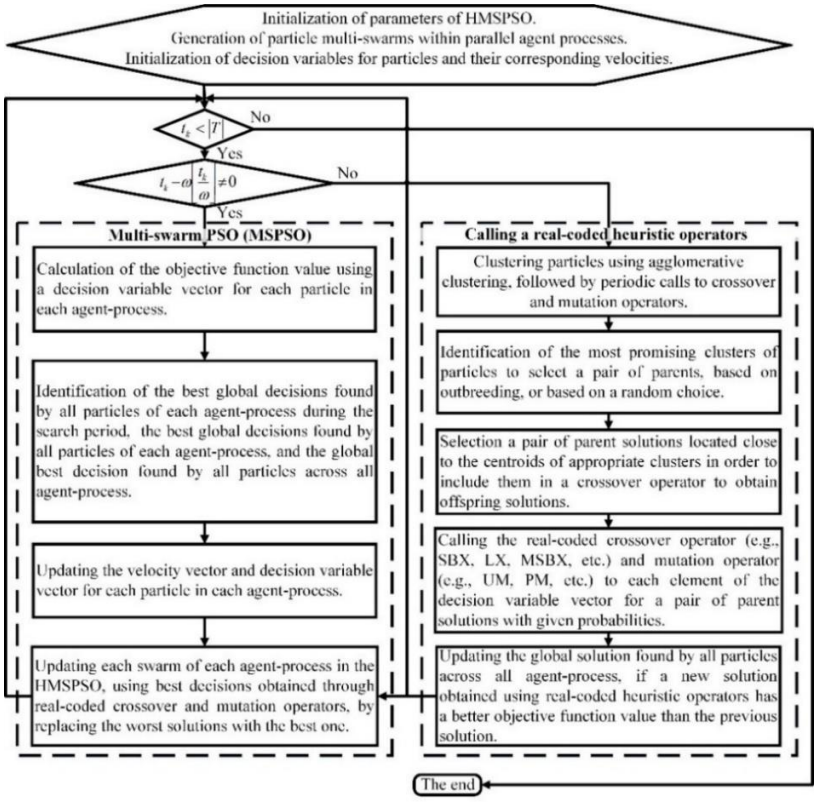

Fig 3. Flowchart of the Hybrid Multi-Swarm Particle Swarm Optimization Algorithm

The flowchart of the HMSPSO Algorithm, in a simplified form, is presented in Fig. 3. The proposed HMSPSO Algorithm operates as follows. Firstly, the initialization of parameters of the HMSPSO is performed, followed by the generation of particle multi-swarms within parallel agent processes and the initialization of values for decision variables for the particles and their corresponding velocities. The HMSPSO Algorithm is then iterated. During these iterations, the calculation of the objective function value is performed using the decision variable vector for each particle in each parallel agent process. The identification of the best global decisions

and the updating of the velocity vector and decision variable vector for each particle in each agent process are also carried out. During the iterations, real-coded heuristic operators are periodically called. To accomplish this, particles are clustered periodically using an agglomerative clustering technique. The most promising groups of particles are identified in order to select a pair of parent solutions. Then, real-coded crossover and mutation operations are applied to each component of the decision vectors of parent solutions.

The current global solution found by all particles across all agent processes is subsequently updated if a new solution obtained with real-coded heuristic operators has a better objective function value. Finally, each swarm of each agent-process in the HMSPSO is updated with the best solutions obtained through real-coded crossover and mutation operations by replacing the worst solutions with the best one.

4. Results of simulation and optimization experiments

In the first stage, the HMSPSO Algorithm was evaluated and verified. Table 1 presents the test instances used to test and validate the single-objective evolutionary algorithm (HMSPSO). The table also includes known test instances [24] that were used to verify the HMSPSO algorithm. These test functions are scalable and multimodal, with several decision variables set at 100 (m=100). This allows for the solution of large-scale optimization problems. Among the used test instances, there are differentiable functions (e.g., FT2, FT3) and non-differentiable functions (e.g., FT6).

Table 1. Test instances for HMSPSO

Test instances	Problem statement and global minimum	Feasible ranges
FT1 – Rastrigin function	$f(\mathbf{x}) = 10m + \sum_{i=1}^{m}\left(x_i^2 - 10\cos(2\pi x_i)\right)$, $f(0,0,...,0) = 0$.	$-5.12 \le x_i \le 5.12$, $i = 1, 2, ..., m.$
FT2 – Rosenbrock function	$f(\mathbf{x}) = \sum_{i=1}^{m-1}\left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right)$, $f(1,1,...,1) = 0$.	$-5 \le x_i \le 5$, $i = 1, 2, ..., m.$
FT3 – Ackley function	$f(\mathbf{x}) = -20e^{-0.2\sqrt{\left(\frac{1}{n}\right)\sum_{i=1}^{m} x_i^2}} - e^{\frac{1}{n}\cos(2\pi x_i^2)} + e^1$, $f(0,0,...,0) = 0$.	$-32 \le x_i \le 32$, $i = 1, 2, ..., m.$
FT4 – Schwefel function	$f(\mathbf{x}) = -\frac{1}{m}\sum_{i=1}^{m} x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$, $f(420.9685,...,420.9685) = -418.98289$.	$-500 \le x_i \le 500$, $i = 1, 2, ..., m.$
FT5 – Styblinkski-Tang	$f(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{m}\left(x_i^4 - 16x_i^2 + 5x_i\right)$, $f(-2.903534,...,-2.903534) = -39.16616m$.	$-5 \le x_i \le 5$, $i = 1, 2, ..., m.$
FT6 – Cosine mixture function	$f(\mathbf{x}) = -0.1\sum_{i=1}^{m}\cos(5\pi x_i) - \sum_{i=1}^{m} x_i^2$, $f(\tilde{\mathbf{x}}) = -m,\ \tilde{\mathbf{x}} \in \{-1,\ 1\}$.	$-1 \le x_i \le 1$, $i = 1, 2, ..., m.$

Furthermore, the effectiveness of the proposed HMSPSO Algorithm was compared with other evolutionary algorithms. An evaluation of performance metrics values was performed for the following algorithms:

- PSO is the standard particle swarm optimization algorithm implemented using GPUs in parallel [12].
- DMS-PSO is a well-known dynamic multi-swarm particle swarm optimization algorithm that is also implemented in parallel [22].
- MA-RCGA is the parallel multi-agent real-coded genetic algorithms that use the combined crossover operator and agents' cooperation [16].
- RCGA-PSO is the early suggested parallel hybrid genetiA c algorithm that combines RCGA and PSO [17].
- **HMSPSO is the proposed optimization algorithm.**

Optimisation experiments were carried out on a DSWS PRO portable supercomputer (2 x Intel Xeon Silver 4114, 1 x NVIDIA QUADRO RTX 6000) with a limited number of iterations. $|T| = 100$, the total number of agent processes for HMSPSO and DMS-PSO $\left(|J_n| = 10 \text{ for all } n \in N\right)$, while for other algorithms it is 100. The number of decision variables $|I| = 100$. All optimization experiments were conducted multiple times (at least 100 iterations for each test instance and algorithm), and HMSPSO demonstrated stable results.

Table 2. Evaluation of performance metrics of HMSPSO

Test instances	Metrics	**HMSPSO**	PSO and RCGA algorithms				Optimum
			PSO	DMS-PSO	MA-RCGA	RCGA-PSO	
FT1	$\tilde{F}(\mathbf{x})$	**0**	810.33	0	0	0.167	**0**
	α	**1.571**	0.858	1.238	1.272	1.401	
	PT (s)	**4**	2	3	23	6	
FT2	$\tilde{F}(\mathbf{x})$	**0.141**	21474.8	98.94	0	0.568	**0**
	α	**1.553**	1.534	1.198	0.987	1.375	
	PT (s)	**5**	1	4	24	7	
FT3	$\tilde{F}(\mathbf{x})$	**0**	21.589	0	0.118	0.112	**0**
	α	**1.572**	0.809	1.477	1.497	1.011	
	PT (s)	**4**	1	3	22	6	
FT4	$\tilde{F}(\mathbf{x})$	**–390.34**	–122.17	–141.99	–380.32	–367.695	**–418.829**
	α	**1.477**	1.270	1.311	1.465	1.041	
	PT (s)	**4**	1	3	23	6	
FT5	$\tilde{F}(\mathbf{x})$	**–3916.44**	–2568.19	–2922.20	–3916.5	–3916.49	**–3916.612**
	α	**1.214**	1.103	1.203	0.924	1.032	
	PT (s)	**5**	1	4	22	7	
FT6	$\tilde{F}(\mathbf{x})$	**–89.23**	–55.13	–61.168	–89.98	–88.42	**–100**
	α	**1.461**	1.272	1.242	1.521	1.103	
	PT (s)	**5**	1	4	23	6	

The optimization results are presented in Table 2. Here, $\tilde{F}(\mathbf{x})$ is the calculated value of the objective function, which is the mean of multiple runs; α is the convergence rate of the objective function, as estimated using the tangent slope angle

of the logarithmic dependency graph $\left\|\tilde{F}(\mathbf{x}_{t_k}) - \hat{F}(\hat{\mathbf{x}})\right\| \le \alpha \left\|\tilde{F}(\mathbf{x}_{t_{k-1}}) - \hat{F}(\hat{\mathbf{x}})\right\|^{\beta}$, where $\hat{F}(\hat{\mathbf{x}})$ is the known optimum.

As can be seen from Table 2, the HMSPSO Algorithm exhibits superior performance, especially when compared to DMS-PSO and RCGA-PSO. This is primarily due to its accuracy (i.e., the closeness to a global optimum) and convergence rate. The accuracy of HMSPSO is comparable to that of MA-RCGA, which is one of the most advanced genetic algorithms. However, the HMSPSO Algorithm offers significantly improved time efficiency. HMSPSO is only slightly inferior to that of standard particle swarm optimization algorithms in terms of process time (PT in s). This is because of the additional processing time required for implementing crossover operators and updating solutions within populations of processing agents, in contrast to DMS-PSO. On the other hand, HMSPSO provides a more efficient optimization of multimodal function values, such as FT2, FT4, FT5, and FT6.

In the second stage, the HMSPSO Algorithm was aggregated with the developed agent-based epidemiological model through the objective function and constraints. The optimal values of the decision variables obtained through the use of HMSPSO and the epidemiological model are presented in Table 3. Here: Scenario 1 – a minimum proportion of infected individuals (0.1%) uniformly distributed throughout the model space; Scenario 2 – a small proportion of infected individuals (0.2%) uniformly distributed throughout the model space; Scenario 3 – an average proportion of infected individuals (0.3%), which are located inside a circle with a radius of *R*; Scenario 4 – an average proportion of infected individuals (0.5%), which are located inside a circle with a radius of 1.5 *R*; Scenario 5 – a significant proportion of infected individuals (0.75%), which are located outside a circle with a radius of 2*R*; Scenario 6 – all individuals are infected and located outside a circle with a radius of 1.5*R*.

As shown in Table 3, the optimal values of the decision variables calculated using HMSPSO vary for different scenarios. There is an increase in the need for hospital bed capacity and self-quarantine capacity as the epidemiological situation worsens (see Scenarios 1 and 2 compared to Scenarios 5 and 6). There is no direct correlation between changes in the intensities of vaccination, self-quarantine, and hospitalization. Specific optimal strategies have been calculated using the HMSPSO algorithm aggregated with the developed agent-based epidemiological model for different age groups.

Fig. 4 shows the dynamics of the convergence rate of the objective function (the share of surviving agents), which was calculated using HMSPSO. As can be seen from Fig. 4, convergence is observed in all considered scenarios of epidemic situations. However, for the more complex scenarios characterized by a significant share of infected individuals (Scenarios 5 and 6), the convergence rate is slower.

Table 3. Optimal values of decision variables computed with the HMSPSO

Parameters	Age groups	Scenarios					
		Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
Intensities of vaccinations r_1	0-10	0.962	0.163	0.987	0.228	0.797	0.896
	11-20	0.043	0.341	0.965	0.823	0.894	0.535
	21-30	0.794	0.049	0.317	0.273	0.825	0.941
	31-40	0.762	0.432	0.324	0.858	0.748	0.913
	41-50	0.161	0.889	0.541	0.623	0.555	0.799
	51-60	0.968	0.712	0.323	0.116	0.501	0.756
	61-70	0.882	0.655	0.794	0.312	0.627	0.996
	71-80	0.359	0.764	0.715	0.960	0.746	0.847
	81-90	0.452	0.310	0.585	0.970	0.595	0.980
	91-100	0.116	0.642	0.895	0.127	0.934	0.998
Intensities of self-quarantine r_2	0-10	0.866	0.975	0.033	0.990	0.759	0.992
	11-20	0.873	0.553	0.967	0.999	0.664	0.727
	21-30	0.801	0.648	0.553	0.821	0.778	0.975
	31-40	0.643	0.977	0.115	0.647	0.272	0.552
	41-50	0.385	0.609	0.768	0.910	0.870	0.902
	51-60	0.871	0.292	0.972	0.659	0.928	0.791
	61-70	0.965	0.440	0.990	0.403	0.250	0.639
	71-80	0.997	0.812	0.646	0.314	0.979	0.538
	81-90	0.981	0.965	0.980	0.254	0.033	0.664
	91-100	0.994	0.655	0.424	0.646	0.919	0.414
Intensities of hospitalization r_3	0-10	0.992	0.589	0.400	0.172	0.430	0.801
	11-20	0.341	0.674	0.464	0.394	0.774	0.791
	21-30	0.670	0.265	0.282	0.582	0.599	0.970
	31-40	0.802	0.278	0.54	0.424	0.239	0.794
	41-50	0.604	1.000	0.261	0.235	0.749	0.816
	51-60	0.191	0.421	0.972	0.744	0.707	0.951
	61-70	0.749	0.618	0.429	0.765	0.423	0.714
	71-80	0.027	0.084	0.768	0.486	0.485	0.522
	81-90	0.760	0.708	0.501	0.997	0.649	0.698
	91-100	0.887	0.251	0.440	0.699	0.884	0.951
Hospital bed capacity, cells		248	310	579	789	969	1341
Self-quarantine capacity, cells		1784	1759	2790	2857	2909	3238
Share of surviving agents		**0.964**	**0.924**	**0.869**	**0.792**	**0.695**	**0.577**

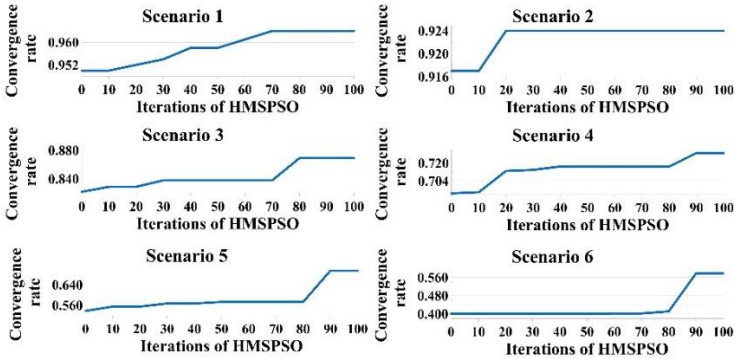

Fig. 4. Dynamics of the convergence rate of the objective function (the share of surviving agents), calculated using HMSPSO

Next, sensitivity analysis was performed to assess the impact of varying vaccination and self-quarantine intensity on the share of surviving agents, as well as

the effect of vaccination and hospitalization intensity. This sensitivity analysis is presented in Fig. 5.

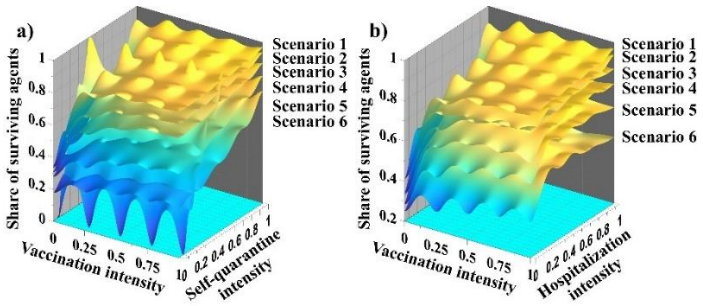

Fig. 5. Sensitivity analysis of the share of surviving agents to variations in: (a) vaccination intensity and self-quarantine intensity, and (b) vaccination intensity and hospitalization intensity

As evident from Fig. 5, the share of surviving agents depends on the intensity of vaccination, self-quarantine, and hospitalization in all scenarios. At the same time, as the epidemiological situation becomes more complicated, the functional dependence on appropriate control parameters for the share of survivors becomes multimodal, as shown in Fig. 5 (for instance, see Scenario 6). Therefore, the use of the HMSPSO Algorithm to solve an agent-based epidemiological model is fully justified.

5. Conclusion

This paper presents a novel Hybrid Multi-Swarm Particle Swarm Optimization Algorithm (HMSPSO Algorithm) for solving agent-based epidemiological models. Unlike other well-known particle swarm optimization algorithms, this method uses alternating real-coded genetic operators applied to parent solutions selected from sub-swarms generated through agglomerative clustering. Thanks to the periodic use of real-coded heuristic operators, HMSPSO demonstrates superior performance compared to other well-established evolutionary algorithms, including those based on multi-swarm intelligence. The HMSPSO Algorithm was combined with an agent-based epidemic model, resulting in the calculation of optimal strategies for anti-epidemic measures aimed at maximizing the share of survival agents. The main value of this framework lies in its ability to identify optimal strategies for prevention and anti-epidemic control. These strategies include control of the intensity of various anti-epidemic measures, such as vaccination, self-quarantine, and hospitalization, and factors affecting the capacity of hospital and quarantine facilities, among other parameters, in different scenarios.

Further research will focus on the detailed development of the proposed agent-based epidemiological model to make it more realistic and address real-world epidemiological challenges.

***Acknowledgments*:** This work was supported by the Ministry of Science and Higher Education of the Russian Federation under Grant 075-15-2024-525.

References

1. H a m e r, W. On Epidemic Disease in England – The Evidence of Variability and of Persistency of Type, Lecture III. – The Lancet, Vol. **167**, 1906, No 4305, pp. 733-739.
2. K e n d a l l, D. G. Deterministic and Stochastic Epidemics in Closed Populations. – In: Proc. of 3rd Berkeley Symposium on Mathematical Statistics and Probability. Vol. IV. Contributions to Biology and Problems of Health. University of California Press, 1956, pp. 149-165.
3. C a m p o s, L. C., R. P. C y s n e, A. L. M a d u r e i r a, G. L. Q. M e n d e s. Multi-Generational SIR Modeling: Determination of Parameters, Epidemiological Forecasting, and Age-Dependent Vaccination Policies. – Infectious Disease Modelling, Vol. **6**, 2021, pp. 751-765.
4. C i u n k i e w i c z, P., W. B r o o k e, M. R o g e r s, S. Y a n u s h k e v i c h. Agent-Based Epidemiological Modeling of COVID-19 in Localized Environments. – Computers in Biology and Medicine, Vol. **144**, 2022, Article No 105396.
5. K e r k m a n n, D., S. K o r f, K. N g u y e n, D. A b e l e, A. S c h e n g e n, C. G e r s t e i n, J. H. G ö b b e r t, A. B a s e r m a n n, M. J. K ü h n, M. M e y e r-H e r m a n n. Agent-Based Modeling for Realistic Reproduction of Human Mobility and Contact Behavior to Evaluate Test and Isolation Strategies in Epidemic Infectious Disease Spread. – Computers in Biology and Medicine, Vol. **193**, 2025, Article No 110269.
6. Q i u, Z., Y. S u n, X. H e, J. W e i, R. Z h o u, J. B a i, S. D u. Application of Genetic Algorithm Combined with Improved SEIR Model in Predicting the Epidemic Trend of COVID-19, China. – Scientific Reports, Vol. **12**, 2022, No 1, Article No 8910.
7. G r a n a d o s, B. G., M. C. G. Q u i n t e r o, C. V. N ú ñ e z. Improved Genetic Algorithm Approach for Coordinating Decision-Making in Technological Disaster Management. – Neural Computing and Applications, Vol. **36**, 2024, pp. 4503-4521.
8. H a o u a r i, M., M. M h i r i. A Particle Swarm Optimization Approach for Predicting the Number of COVID-19 Deaths. – Scientific Reports, Vol. **11**, 2021, Article No 16587.
9. P i o t r o w s k i, A. P., A. E. P i o t r o w s k a. Differential Evolution and Particle Swarm Optimization against COVID-19. – Artificial Intelligence Review, Vol. **55**, 2022, pp. 2149-2219.
10. M a r z i a, A., M. H. S u l a i m a n, A. J. M o h a m a d. Improved Barnacle Mating Optimizer-Based Least Squares Support Vector Machine to Predict COVID-19 Confirmed Cases with Total Vaccination. – Cybernetics and Information Technologies, Vol. **23**, 2023, No 1, pp. 125-140.
11. D i d i, Y., A. W a l h a, A. W a l i. Integrating Environmental Clustering to Enhance Epidemic Forecasting with Machine Learning Models. – International Journal of Cognitive Computing in Engineering, Vol. **6**, 2025, pp. 628-642.
12. K e n n e d y, J., R. E b e r h a r t. Particle Swarm Optimization. – In: Proc. of IEEE International Conference on Neural Networks, Vol. **4**, 1995, pp. 1942-1948.
13. A k o p o v, A . S. A Clustering-Based Hybrid Particle Swarm Optimization Algorithm for Solving a Multisectoral Agent-Based Model. – Studies in Informatics and Control, Vol. **33,** 2024, No 2, pp. 83-95.
14. S m i r n o v, A. V. Method for Estimating Objective Function Landscape Convexity during Extremum Search. – Russian Technological Journal, Vol. **13**, 2025, No 2, pp. 121-131.
15. H e r r e r a, F., M. L o z a n o, J. L. V e r d e g a. Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis. – Artificial Intelligence Review, Vol. **12**, 1998, No 4, pp. 265-319.
16. A k o p o v, A. S., L. A. B e k l a r y a n, M. T h a k u r, B. D. V e r m a. Parallel Multi-Agent Real-Coded Genetic Algorithm for Large-Scale Black-Box Single-Objective Optimization. – Knowledge-Based Systems, Vol. **174**, 2019, pp. 103-122.
17. A k o p o v, A. S., A. L. B e k l a r y a n, A. A. Z h u k o v a. Optimization of Characteristics for a Stochastic Agent-Based Model of Goods Exchange with the Use of a Parallel Hybrid Genetic Algorithm. – Cybernetics and Information Technologies, Vol. **20**, 2023, No 3, pp. 45-63.
18. R o m a s e v y c h, Y., L. V i a t c h e s l a v, B. Z i v. Advanced PSO Algorithms Development with Combined LBEST and GBEST Neighborhood Topologies. – Cybernetics and Information Technologies, Vol. **24**, 2024, No 3, pp. 59-77.

19. S t o i l o v, T., K. S t o i l o v a. Bi-Level Optimization of Inventory and Production. – Cybernetics and Information Technologies, Vol. **25**, 2025, No 1, pp. 126-141.
20. A u d e t, C., M. K o k k o l a r a s. Blackbox and Derivative-Free Optimization: Theory, Algorithms and Applications. – Optimization and Engineering. Vol. **17**, 2016, pp. 1-2.
21. B o r s h c h e v, A. The Big Book of Simulation Modeling: Multimethod Modeling with AnyLogic. Hampton. NJ, AnyLogic North America, 2013.
22. L i a n g, J. J., P. N. S u g a n t h a n. Dynamic Multi-Swarm Particle Swarm Optimizer. – In: Proc. of IEEE Swarm Intelligence Symposium (SIS'05), Pasadena, CA, USA, 2005, pp. 124-129.
23. M o j e n a, R. Hierarchical Grouping Methods and Stopping Rules: An Evaluation. – The Computer Journal, Vol. **20**, 1977, No 4, pp. 359-363.
24. L i, X., A. E n g e l b r e c h t, M. G. E p i t r o p a k i s. Benchmark Functions for CEC'2013 Special Session and Competition on Niching Methods for Multimodal Function Optimization. – Technical Report, Evolutionary Computation and Machine Learning Group, RMIT University, Australia, 2013.

Fast-track. Received: 14.08.2025, First revision: 07.10.2025, Accepted: 11.10.2025