Optimisation of System Dynamics Models Using a Real-Coded Genetic Algorithm with Fuzzy Control

Gayane L. Beklaryan, Andranik S. Akopov, Nerses K. Khachatryan

Plekhanov Russian University of Economics, Moscow, Russian Federation
E-mails: glbeklaryan@gmail.com   andranik.s.akopov@ieee.org   nerses-khachatryan@yandex.ru

Abstract: *This paper presents a new real-coded genetic algorithm with Fuzzy control for the Real-Coded Genetic Algorithm (F-RCGA) aggregated with System Dynamics models (SD-models). The main feature of the genetic algorithm presented herein is the application of fuzzy control to its parameters, such as the probability of a mutation, type of crossover operator, size of the parent population, etc. The control rules for the Real-Coded Genetic Algorithm (RCGA) were suggested based on the estimation of the values of the performance metrics, such as rate of convergence, processing time and remoteness from a potential extremum. Results of optimisation experiments demonstrate the greater time-efficiency of F-RCGA in comparison with other RCGAs, as well as the Monte-Carlo method. F-RCGA was validated by using well-known test instances and applied for the optimisation of characteristics of some system dynamics models.*

Keywords: *real-coded genetic algorithm, fuzzy control, system-dynamics, optimisation methods.*

1. Introduction

This paper considers continuous optimization problems solved with the use of the developed real-coded genetic algorithm with Fuzzy control for the Real-Coded Genetic Algorithm (F-RCGA). The algorithm is based on the early-developed parallel Multi-Agent Real-Coded Genetic Algorithm (MA-RCGA) [1] that is improved through including the fuzzy control for setting main parameters having an influence on the optimisation process. The objective functions of considered continuous optimisation problems are computed in the result of simulation modelling with the use of the system dynamics method. Main advantages of the suggested F-RCGA are related to solving continuous optimisation problems because real-coded heuristic operators provide better precision of solutions without the need to have large-sized populations of potential decisions. However, as shown in [1] real-coded genetic algorithms can have various types of heuristic operators including the discrete crossover and discrete mutation. These operators allow forming the values of potential decisions in the discrete space. Thus, F-RCGA can be applied for solving

combinatorial optimisation problems too, if discrete heuristic operators are used exceptionally. Moreover, there is a possibility to execute both continuous and discrete real-coded heuristic operators with defined probabilities that allows solving discrete-continuous optimisation problems.

Further, the following black-box single-objective optimisation problem will be considered:

(1) $$\min F(\mathbf{x}),$$

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, $\Omega = \prod_{i=1}^{n} [a_i, b_i]$ is the feasible region of the search space

($i = 1, 2, ..., n$ is the index of decision variables), and $F : \Omega \to \mathbb{R}$ is the objective function that is computed in the result of the simulation modelling. Such problem statement is suggested in works [1, 2, 3]. In particular, the parallel MA-RCGA aggregated with the ecological model implemented in the simulation tool is suggested in [1]. Moreover, MA-RCGA uses the method of Finite State Automata (FSA) at the level of each agent-process to control the main parameters of the genetic algorithm, such as the probability of a mutation, type of crossover operator, size of the parent population, etc. Parallel MA-RCGA based on FSA has a deterministic adaptive system that allows switching to different states of the optimisation algorithm, which are characterized by appropriate values of control parameters. In contrast, this work aims to design a soft adaptive system of control for F-RCGA based on fuzzy logic. It allows using probabilistic rules of switching GA to possible states by setting appropriate values of control parameters. Thus, the degree of membership of various performance metrics belonging to existing agent-processes clusters is used to form the control action in the suggested F-RCGA. Such an approach is more effective because the optimal threshold values used in FSA-based methods are not known for black-box optimisation problems.
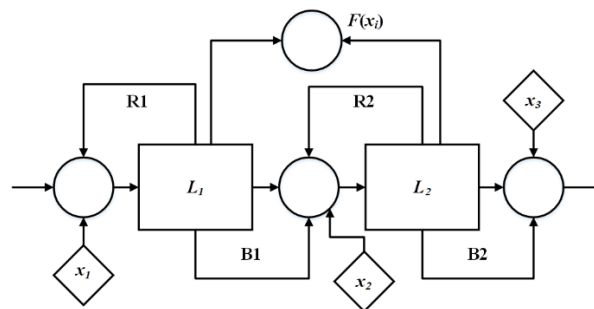

Fig. 1. An example of a typical system dynamics model consisting
of two connected reservoirs with feedbacks

The system dynamics model can be considered as a black box device with a single-objective function and a vector of decision variables. The theory of system dynamics was developed by J. Forrester in the 50-ties of the past century [1]. The important feature of such models is the availability of sequences of system levels (reservoirs) connected to each other through direct and feedback relations (Fig. 1).

Thus, such a model can have some reinforcing (e.g., R1, R2) and balanced (e.g., B1, B2) feedbacks. Each system level (e.g., L1, L2) is a combiner having several input and output flows with their rates and describes the behaviour of a simple railway station. The description of such complex systems is given in works [4-8].

Such a model can be described by the following system of ordinary differential equations:

(2) $$\dot{y}_i(\xi) = f_i(\xi) x_i(\xi) y_i(\xi - 1) - f_{i+1}(\xi) x_{i+1}(\xi) y_{i+1}(\xi - 1),$$

$$i = 1, 2, ..., n - 1,$$

and the objective function is

(3) $$F(\mathbf{x}(\xi)) = \sum_{i=1}^{n-1} y_i(\xi),$$

where $y_i(\xi)$ is the value of the i-th combiner, ξ is the continuous time, and $f_i(\xi)$ are the internal functions of the model influencing the rates of i-ths flows ($i = 1, 2, ..., n$).

System (2) defines a complex landscape of objective function (3) which depends on the types of internal functions present. As a rule, objective function (3) is not convex and can have breakpoints. As a result, standard Newton and quasi-Newton optimisation methods [9, 10] are not applicable. Therefore, RCGA with fuzzy control can be applied for such optimisation problems.

This paper focuses on creating a novel genetic algorithm using fuzzy control to set the main parameters that influence the probability of a mutation, the rate of convergence, the diversity of the potential decisions and other performance characteristics to provide the optimisation flows in system dynamics models. Using standard test instances it is shown that F-RCGA is more effective than other implementations of genetic algorithms. The important section of this paper involves testing the typical system dynamics model for the validation of F-RCGA. The aggregated problem of the optimal control of flows in system dynamics models is considered. The objective function minimizes the total occupancy time of railway stations which are modelled as reservoirs connected with each other through flows with variable rates. The model is implemented in Powersim and aggregated with the developed F-RCGA through the objective function (summarized cargo residue). This paper demonstrates the possibility of applying F-RCGA for the optimisation of characteristics of complex system-dynamics models with direct and feedback relations.

2. Related works

The main feature of many system-dynamics models is the aggregation of separate sub-models within a united simulation model [5]. This work demonstrates the approach to designing integrated system dynamics models for an oil company whose components are connected with each other through direct and feedback relations. It allows modelling the behaviour of such complex systems by taking into account the mutual influence of each node of the production supply chain.

In another work, a model for organizing cargo transportation between two node stations connected by a railway line that contains a certain number of intermediate stations was suggested [11]. In this model, the movement of cargo goes in one direction. Such behaviour is typical of transport systems in which one of the node stations is located in a region, which produces raw materials for manufacturing industry located in another region where another node station is located.

Most such problems are characterized by increased demands for precision of solutions to provide the minimum total cargo residues at stations. Therefore, real-coded genetic algorithms should be applied instead of binary-coded heuristic algorithms.

Examples of the development and application of parallel genetic algorithms for black-box optimisation problems are presented in the following works [1, 12-15]. In these works, parallel genetic algorithms based on the multi-agent agent architecture (MAGAMO) have been suggested. The important advantage of such an approach is the co-evolution process based on the periodic exchange of the best potential decisions belonging to agent-processes through a global population. Such agent interaction can be applied for designing the fuzzy control to provide probabilistic rules for transitions to different states of RCGA by setting appropriate control parameters. There are other important works devoted to developing genetic algorithms [16-20]. Among them, works should be highlighted concerning the development of RCGAs [17] and coupled using genetic algorithms and fuzzy logic systems described in the book [18]. These pioneering works represent the foundations of the development of RCGAs with fuzzy control.

Currently, GAs are applied to the optimisation of ship passage planning [21], the design of genetic fuzzy systems [22], and control of agent-rescuer behaviour based on the fuzzy clustering [23], amongst other problems.

3. Proposed methods

3.1. Genetic algorithm with fuzzy control

The first important feature to note of the suggested genetic algorithm with fuzzy control (F-RCGA) is that it uses the fuzzy set of possible states of the algorithm to provide probabilistic transitions between various states of agent-processes by changing the values of associated parameters. Because the efficiency of F-RCGA depends on many factors, the following integral performance metric will be used at the level of each agent-process:

$$(4) \quad M_k(\tilde{t}_k) = c_1 \frac{\mathrm{RC}_k(\tilde{t}_k)}{\frac{1}{|K|}\sum_{k=1}^{|K|}\mathrm{RC}_a(\tilde{t}_k)} + c_2 \frac{\mathrm{DP}_k(\tilde{t}_k)}{\frac{1}{|K|}\sum_{a=1}^{|K|}\mathrm{DP}_k(\tilde{t}_k)} + c_3\left(1 - \left|\frac{F_k(\tilde{t}_k) - \tilde{F}}{\tilde{F}}\right|\right),$$

$$(5) \quad \mathrm{DP}_a(\tilde{t}_k) = \sum_{i=1}^{|I|}\left|x_{ai}(\tilde{t}_k) - \tilde{x}_i(\tilde{t}_k)\right|,$$

$$c_1 + c_2 + c_3 = 1, \quad 0 \le c_l \le 1 \text{ for all } l = 1,\ 2,\ 3, \quad k \in K, \quad \tilde{t}_k \in T_k, \quad i \in I,$$

where: K is the set of indexes of agent-processes; $|K|$, $|I|$ are numbers of elements in appropriate sets; T_k is the set of external iteration indexes; $\tilde{t}_k = \tilde{t}_1,\ \tilde{t}_2,\ \dots,\ \tilde{T}_{|K|}$ is the index of external iterations of F-RCGA, where $\tilde{T}_{|K|}$ is the total number of external iterations; I is the set of indexes of decision variables; $RC_k(\tilde{t}_k)$ is the rate of convergence of the genetic algorithm of the k-th-agent-process $(k \in K)$ at iteration $\tilde{t}_k$ $(\tilde{t}_k \in T_k)$; $DP_k(\tilde{t}_k)$ is the level of the population diversity of the k-th-agent-process $(k \in K)$ at iteration $\tilde{t}_k$ $(\tilde{t}_k \in T_k)$; $F_k(\tilde{t}_k)$ is the value of the objective function computed at the level of the k-th-agent-process $(k \in K)$ at iteration $\tilde{t}_k$ $(\tilde{t}_k \in T_k)$; $\tilde{F}$ is the known reference objective function value; $x_{ki}(\tilde{t}_k)$ is the value of the i-th-decision variable $(i \in I)$ computed at the level of the k-th-agent-process $(k \in K)$ with the help of the crossover, mutation and selection operators at iteration $\tilde{t}_k$ $(\tilde{t}_k \in T_k)$; $\{c_1,\ c_2,\ c_3\}$ are the preference coefficients set by a decision-maker.

The following states of the k-th-agent-process $(k \in K)$ will be considered for F-GA at iteration $\tilde{t}_k$ $(\tilde{t}_k \in T_k)$.

To define the current state of the agent-process the following set of three characteristic functions will be used:

• the smooth Z-function membership of the k-th -agent-process $(k \in K)$ to the first state when the low performance metrics are observed is

(6) $\qquad \chi_1\big(M_k(\tilde{t}_k)\big) = \begin{cases} 0 \text{ if } M_k(\tilde{t}_k) \le m_{11}, \\[2mm] 1 - 2\left(\dfrac{M_k(\tilde{t}_k) - m_{12}}{m_{12} - m_{11}}\right)^2 \text{ if } m_{11} \le M_k(\tilde{t}_k) \le \dfrac{m_{12} + m_{11}}{2}, \\[2mm] 2\left(\dfrac{M_k(\tilde{t}_k) - m_{12}}{m_{12} - m_{11}}\right) \text{ if } \dfrac{m_{12} + m_{11}}{2} \le M_k(\tilde{t}_k) \le m_{12}, \\[2mm] 0 \text{ if } M_k(\tilde{t}_k) \ge m_{12}; \end{cases}$

• the trapezoidal membership function of the k-th -agent-process $(k \in K)$ to the second state when the average performance metrics are observed is

(7) $\qquad \chi_2\big(M_k(\tilde{t}_k)\big) = \begin{cases} 0 \text{ if } M_k(\tilde{t}_k) \le m_{21}, \\[2mm] \dfrac{M_k(\tilde{t}_k) - m_{22}}{m_{22} - m_{21}} \text{ if } m_{21} \le M_k(\tilde{t}_k) \le m_{22}, \\[2mm] 1 \text{ if } m_{22} \le M_a(\tilde{t}_k) \le m_{23}, \\[2mm] \dfrac{m_{24} - M_k(\tilde{t}_k)}{m_{24} - m_{23}} \text{ if } m_{23} \le M_k(\tilde{t}_k) \le m_{24}, \\[2mm] 0 \text{ if } M_k(\tilde{t}_k) \ge m_{24}; \end{cases}$

• the smooth Z-function membership of the k-th -agent-process $(k \in K)$ to the third state when the high performance metrics are achieved is

$$(8) \quad \chi_3\left(M_k(\tilde{t}_k)\right) = \begin{cases} 0 \ \text{if} \ M_k(\tilde{t}_k) \le m_{31}, \\[6pt] 2\left(\dfrac{M_k(\tilde{t}_k) - m_{31}}{m_{32} - m_{31}}\right)^2 \ \text{if} \ m_{31} \le M_a(\tilde{t}_k) \le \dfrac{m_{31} + m_{32}}{2}, \\[6pt] 1 - 2\left(\dfrac{m_{32} - M_k(\tilde{t}_k)}{m_{32} - m_{31}}\right)^2 \ \text{if} \ \dfrac{m_{31} + m_{32}}{2} \le M_k(\tilde{t}_k) \le m_{32}, \\[6pt] 1 \ \text{if} \ m_{32} \le M_k(\tilde{t}_k) \le m_{33}, \\[6pt] 1 - 2\left(\dfrac{M_k(\tilde{t}_k) - m_{33}}{m_{34} - m_{33}}\right)^2 \ \text{if} \ m_{32} \le M_a(\tilde{t}_k) \le \dfrac{m_{33} + m_{34}}{2}, \\[6pt] 2\left(\dfrac{m_{34} - M_k(\tilde{t}_k)}{m_{34} - m_{33}}\right)^2 \ \text{if} \ \dfrac{m_{33} + m_{34}}{2} \le M_k(\tilde{t}_k) \le m_{34}, \\[6pt] 0 \ \text{if} \ M_k(\tilde{t}_k) \ge m_{34}. \end{cases}$$

Here, threshold values $\{m_{11}, m_{12}\}$, $\{m_{21}, m_{22}, m_{23}, m_{24}\}$ and $\{m_{31}, m_{32}, m_{33}, m_{34}\}$ are known and updated for each solved optimisation problem, $m_{11} < m_{12}$, $m_{21} < m_{22}$, $m_{23} < m_{24}$, $m_{31} < m_{32}$.

Fig. 2 shows the single-objective real-coded genetic algorithm with fuzzy control.

The suggested algorithm (Fig. 2) is based on the parallel multi-agent real-coded genetic algorithm for large-scale black-box single-objective optimisation that was previously proposed in the work [1]. Combining different crossover and mutation operators was suggested in this work by using the FSA-based control. These are the following heuristic operators:

• Laplace crossover (LX) generates potential decisions in the continuous search space with a high precision and low rate of convergence [24].

• Simulated Binary crossover (SBX) generates potential decisions in the continuous search space with the average rate of convergence [25].

• Modified Simulated Binary crossover (MSBX) generates potential decisions in the continuous search space with a fast rate of convergence and low precision [1].

• Modified Discrete SBX-crossover (DMSBX) generates potential decisions in the discrete search space [1].

• Power Mutation (PM) generates potential decisions in the continuous search space near the parent solutions [26].

• Uniform Mutation (UM) generates potential decisions in the continuous search space regardless of parent solutions [17].

• Discrete Uniform Mutation (DUM) generates potential decisions in the discrete search space regardless of parent solutions [1].

• Scalable Uniform Mutation operator (SUM) allows quantizing the feasible ranges of decision variables into uniform intervals to obtain potential solutions outside the area of local extrema [1].

The following control parameters are suggested for F-RCGA:

- $L_k(\tilde{t}_k) \in [2,\ \bar{L}_k]$ is the size of the local population of parent-individuals belonging to the k-th-agent-process $(k \in K)$ that is limited by the size of the initial population $\bar{L}_k$ at iteration $\tilde{t}_k$ $(\tilde{t}_k \in T_k)$;

- $\omega_k(\tilde{t}_k) \in [1,\ \tilde{T}_k]$ is the frequency of exchanging the best potential decisions between the k-th-agent-process and other processes at iteration $\tilde{t}_k$ $(\tilde{t}_k \in T_k)$;

- $p_{k,\mathrm{cross}}(\tilde{t}_k) \in [0,\ 1]$ is the probability of a crossover operator at iteration $\tilde{t}_k$ $(\tilde{t}_k \in T_k)$;

- $p_{k,\mathrm{mut}}(\tilde{t}_k) \in [0,\ 1]$ is the probability of a mutation operator at iteration $\tilde{t}_k$ $(\tilde{t}_k \in T_k)$;

- $C_k(\tilde{t}_k) \in \tilde{C}_k$ is the subset of crossover operators that can be used at iteration $\tilde{t}_k$ $(\tilde{t}_k \in T_k)$:

$$C_k(\tilde{t}_k) = \{\{LX\},\ \{LX,\ SBX\},\ \{SBX,\ MSBX\}, \\ \{MSBX,\ DMSBX\},\ \{DMSBX\}\};$$

- $M_k(\tilde{t}_k) \in \tilde{M}_k$ is the subset of mutation operators that can be used at iteration $\tilde{t}_k$ $(\tilde{t}_k \in T_k)$:

$$\tilde{M}_k = \{\{PM\},\ \{PM,\ UM\},\ \{UM,\ SUM\},\ \{SUM\}\}.$$

The current values of these parameters are set through the fuzzy control (4)-(8) at each external iteration $\tilde{t}_k$ $(\tilde{t}_k \in T_k)$.

The following test instances that can be related to functions with complex relief were used for the validation of the developed F-RCGA:

- FT1 is the Griewank function:

(9)
$$F(\mathbf{x}) = 1 + 4000 \sum_{i=1}^{n} x_i - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right),$$

$$x_i \in [-600,\ 600],\quad i = 1,\ 2,\ ...,\ n,\quad F(0,0,...,0) = 0.$$

- FT2 is the Lunacek bi-Rastrigin function:

(10)
$$F(\mathbf{x}) = \min\left[\sum_{i=1}^{n}(x_i - \mu_1)^2,\ d \cdot n + s\sum_{i=1}^{n}(x_i - \mu_2)^2\right] + $$
$$+ 10\sum_{i=1}^{n}\left(1 - \cos(2\pi(x_i - \mu_1))\right),$$
$$x_i \in [-5.12,\ 5.12],\quad i = 1,\ 2,\ ...,\ n,\quad F(2.5, 2.5, ..., 2.5) = 0,$$

where

$$\mu_1 = 2.5,\quad \mu_2 = -\sqrt{\frac{\mu_1 - d}{s}},\quad d = 1,\quad s = 1 - \frac{1}{2\sqrt{2 + 20} - 8.2}.$$

- FT3 is Schaffer's F6 function

$$(11) \qquad F(\mathbf{x}) = 0.5 + \frac{\sin^2\left(\sum_{i=1}^{n}\sqrt{x_i^2}\right) - 0.5}{\left(1 + 0.001\sum_{i=1}^{n} x_i^2\right)^2},$$

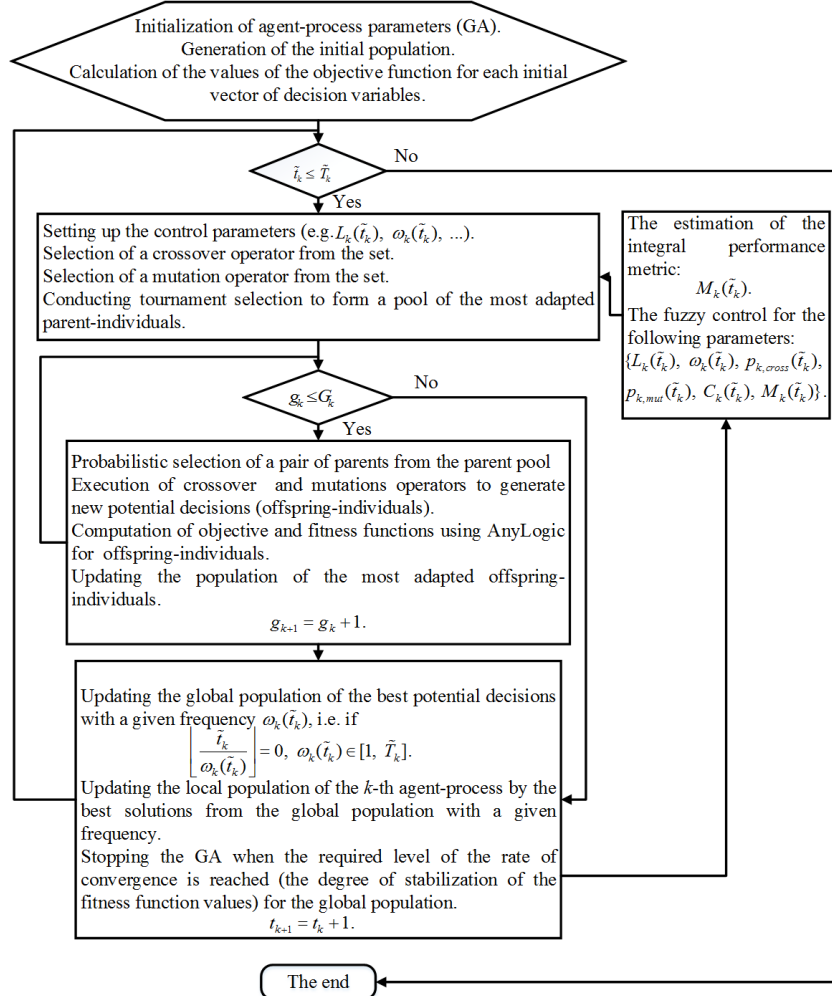$$x_i \in [-100,\ 100],\quad i = 1,\ 2,\ ...,\ n,\quad F(0,0,...,0) = 0.$$

Fig. 2. Real-coded genetic algorithm with fuzzy control (F-RCGA)

The results of testing MA-RCGA with FSA and F-RCGA with fuzzy control are shown in Table 1. The number of decision variables $n = 100$. The number of agent-processes $K = 20$.

Here, $F(\mathbf{x})$ is the value of the objective function, PT, s is the processing time. All tests were conducted using an Intel Core (TM) i7-4980HQ CPU @2.8 GHz four core processor.

Table 1. Performance metrics of F-RCGA in comparison with MA-RCGA

Instance		MA-RCGA with FSA	F-RCGA with Fuzzy Control	Extremum (reference value)
FT1	$F(\mathbf{x})$	0.0510	0.0001	0
	PT, s	2600	1100	
FT2	$F(\mathbf{x})$	0.0064	0.0002	0
	PT, s	1700	680	
FT3	$F(\mathbf{x})$	0.0950	0.0003	0
	PT, s	950	321	

Comparison of the performance metrics of F-RCGA with other optimisation methods was conducted also, e.g., well-known quasi-Newton methods, parallel RCGAs and nonparallel binary GAs. F-RCGA demonstrated the greatest time-efficiency and precision.

The locations of three agent-processes regarding the global extremum at the tenth iteration of F-RCGA $(t_k = 10)$ using Schaffer's F6 function with two decision variables $(n = 2)$ as an example is shown in Fig. 3.
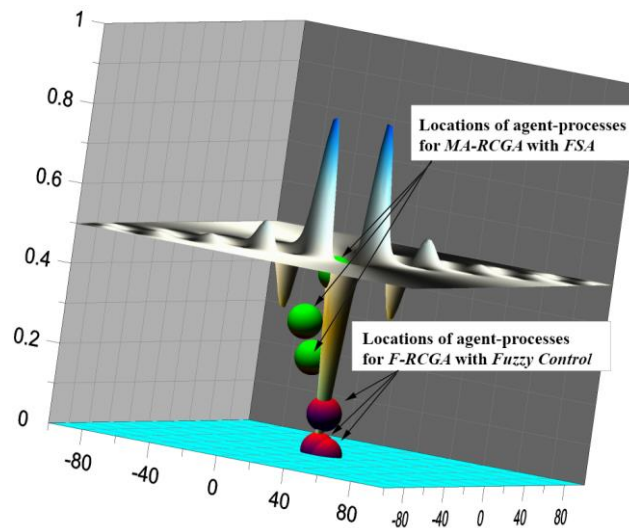

Fig. 3. Locations of three agent-processes regarding the global extremum
using Schaffer's F6 function

3.2. System dynamical modelling

The section is devoted to designing several typical SD-models that can be aggregated with the suggested F-RCGA through their objective functions.

The first SD-model consists of four system levels (combiners), five flows with rates and five decision variables. The implementation of the model in the Powersim simulation tool is shown in Fig. 4.
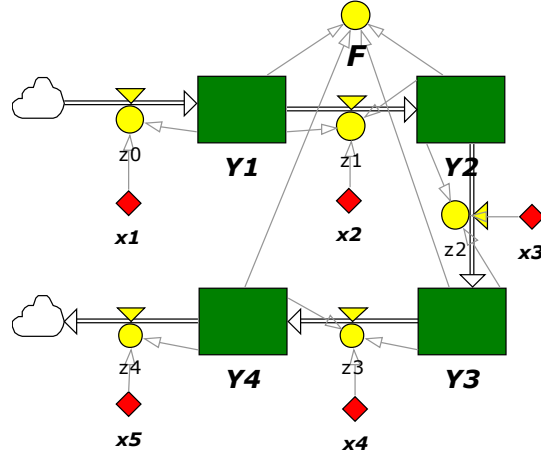
Fig. 4. The implementation of the first SD-model in Powersim

The model is described by the following system of ordinary differential equations in the interval $t_s \leq \xi \leq t_{s+1}$:

(12)
$$\begin{cases} \dot{y}_1(\xi) = x_1(\xi)y_1(\xi-1) - z_1(\xi), \\ \dot{y}_2(\xi) = z_1(\xi) - z_2(\xi), \\ \dot{y}_3(\xi) = z_2(\xi) - z_3(\xi), \\ \dot{y}_4(\xi) = z_3(\xi) - z_4(\xi), \end{cases}$$

where

$$z_1(\xi) = \begin{cases} x_2(\xi)\big(y_1(\xi-1) + y_2(\xi-1)\big) \text{ if } y_1(\xi-1) - x_2(\xi)\big(y_1(\xi-1) + y_2(\xi-1)\big) > 0, \\ y_1(\xi-1) \text{ if } y_1(\xi-1) - x_2(\xi)\big(y_1(\xi-1) + y_2(\xi-1)\big) \leq 0, \end{cases}$$

$$z_2(\xi) = \begin{cases} x_3(\xi)\big(y_2(\xi-1) + y_3(\xi-1)\big) \text{ if } y_2(\xi-1) - x_3(\xi)\big(y_2(\xi-1) + y_3(\xi-1)\big) > 0, \\ y_2(\xi-1) \text{ if } y_2(\xi-1) - x_3(\xi)\big(y_2(\xi-1) + y_3(\xi-1)\big) \leq 0, \end{cases}$$

$$z_3(\xi) = \begin{cases} x_4(\xi)\big(y_3(\xi-1) + y_4(\xi-1)\big) \text{ if } y_3(\xi-1) - x_4(\xi)\big(y_3(\xi-1) + y_4(\xi-1)\big) > 0, \\ y_3(\xi-1) \text{ if } y_3(\xi-1) - x_4(\xi)\big(y_3(\xi-1) + y_4(\xi-1)\big) \leq 0, \end{cases}$$

$$z_4(\xi) = \begin{cases} x_5(\xi)y_4(\xi-1) \text{ if } x_4(\xi) - x_5(\xi)y_4(\xi-1) > 0, \\ y_4(\xi-1) \text{ if } x_4(\xi) - x_5(\xi)y_4(\xi-1) \leq 0. \end{cases}$$

Here, $\{y_1(\xi), y_2(\xi), y_3(\xi), y_4(\xi)\}$, $\{z_1(\xi), z_2(\xi), z_3(\xi), z_4(\xi)\}$ and $\{x_1(\xi), x_2(\xi), x_3(\xi), x_4(\xi), x_5(\xi)\}$ are the values of system levels, flow rates and decision variables at moment ξ respectively, $t_s = t_1, t_2,..., t_S$ is the index of time moments (by days), and S is the simulation period (e.g., ten days).

The values of the control parameters are set at initial moment t_1. The solution of system (12) satisfied the restrictions given below under fixed values of control parameters $\mathbf{x} = \{x_1(t_1), x_2(t_1), x_3(t_1), x_4(t_1), x_5(t_1)\}$ are provided by the fourth-order Runge-Kutta integration method. At the same time, the non-negative of $\{y_1(\xi), y_2(\xi), y_3(\xi), y_4(\xi)\}$ is provided.

The objective function of such a system estimated at the end of the simulation period is

(13) $$F(\mathbf{x}) = y_1(t_S) + y_2(t_S) + y_3(t_S) + y_4(t_S).$$

The target function represents the cumulative residue that is summated by all system levels. As a rule, the residue should be non-negative for each system level during the entire period of modelling (e.g., cargo residues, credit restudies, etc.). Then, the following single-objective box-constrained optimisation problem can be formulated for the system being considered.

Problem A. The need to minimize the value of residue through the set of control parameters $\{x_1(t_1),\ x_2(t_1),\ x_3(t_1),\ x_4(t_1),\ x_5(t_1)\}$ estimated by the end of the simulation period:

(14) $$\min_{\{x_1(t_1),\ x_2(t_1),\ x_3(t_1),\ x_4(t_1),\ x_5(t_1)\}} F(\mathbf{x}),$$

s.t. $0 \le x_1(t_1) \le \bar{x}_1$, $0 \le x_2(t_1) \le \bar{x}_2$, $0 \le x_3(t_1) \le \bar{x}_3$, $0 \le x_4(t_1) \le \bar{x}_4$, $0 \le x_5(t_1) \le \bar{x}_5$.

Here, $\{\bar{x}_1,\ \bar{x}_2,\ \bar{x}_3,\ \bar{x}_4,\ \bar{x}_5\}$ are the upper limits of the decision variables.

Note that problem (14) has multiple solutions depending on feasible ranges set through $\{\bar{x}_1,\ \bar{x}_2,\ \bar{x}_3,\ \bar{x}_4\}$. Thus, it is a multimodal objective function with the extremum $\hat{F}(\hat{\mathbf{x}}) = 0$.

The second SD-model extends the first model through adding two bi-directional flows and additional input flows on the right side. The implementation of the model in the Powersim simulation tool is shown in Fig. 5.
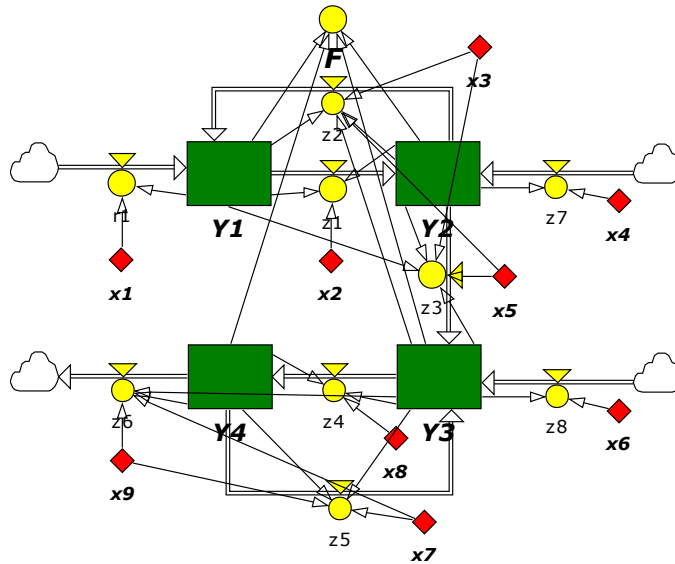

Fig. 5. Implementation of the second SD-model in Powersim

The model is described by the following system of ordinary differential equations in the interval $t_s \le \xi \le t_{s+1}$:

$$
(15) \quad
\begin{cases}
\dot{y}_1(\xi) = x_1(\xi)\,y_1(\xi-1) + z_2(\xi) - z_1(\xi), \\
\dot{y}_2(\xi) = x_4(\xi)\,y_2(\xi-1) + z_1(\xi) - z_2(\xi) - z_3(\xi), \\
\dot{y}_3(\xi) = x_6(\xi)\,y_3(\xi-1) + z_3(\xi) + z_5(\xi) - z_4(\xi), \\
\dot{y}_4(\xi) = z_4(\xi) - z_6(\xi) - z_5(\xi),
\end{cases}
$$

where

$$
z_1(\xi) =
\begin{cases}
x_2(\xi)\big(y_1(\xi-1) + y_2(\xi-1)\big) & \text{if I is true,} \\
y_1(\xi-1), \text{ if } y_1(\xi-1) - x_2(\xi) & \text{if I is false,}
\end{cases}
$$

$$
z_2(\xi) =
\begin{cases}
x_3(\xi)\big(y_1(\xi-1) + y_2(\xi-1)\big) & \text{if II is true,} \\
\dfrac{y_2(\xi-1)}{2} & \text{if II is false,}
\end{cases}
$$

$$
z_3(\xi) =
\begin{cases}
x_5(\xi)\big(y_2(\xi-1) + y_3(\xi-1)\big) & \text{if II is true,} \\
\dfrac{y_2(\xi-1)}{2} & \text{if II is false,}
\end{cases}
$$

$$
z_4(\xi) =
\begin{cases}
x_8(\xi)\big(y_3(\xi-1) + y_4(\xi-1)\big) & \text{if III is true,} \\
y_3(\xi-1) & \text{if III is false,}
\end{cases}
$$

$$
z_5(\xi) =
\begin{cases}
x_7(\xi)\big(y_3(\xi-1) + y_4(\xi-1)\big) & \text{if IV is true,} \\
\dfrac{y_4(\xi-1)}{2} & \text{if IV is false,}
\end{cases}
$$

$$
z_6(\xi) =
\begin{cases}
x_9(\xi)\,y_4(\xi-1) & \text{if IV is true,} \\
\dfrac{y_4(\xi-1)}{2} & \text{if IV is false,}
\end{cases}
$$

I. $y_1(\xi-1) - x_2(\xi)\big(y_1(\xi-1) + y_2(\xi-1)\big) > 0$,

II. $y_2(\xi-1) - x_3(\xi)\big(y_1(\xi-1) + y_2(\xi-1)\big) - x_5(\xi)\big(y_2(\xi-1) + y_3(\xi-1)\big) > 0$,

III. $y_3(\xi-1) - x_8(\xi)\big(y_3(\xi-1) + y_4(\xi-1)\big) > 0$,

IV. $y_4(\xi-1) - x_7(\xi)\big(y_3(\xi-1) + y_4(\xi-1)\big) - x_9(\xi)\,y_4(\xi-1) > 0$.

Here, $\mathbf{x} = \{x_1(t_1),\, x_2(t_1),\, ...,\, x_9(t_1)\}$ is the set of decision variables and $\bar{\mathbf{x}}$ is the set of upper limits for decision variables.

Then, the following single-objective box-constrained optimisation problem can be formulated for the system being considered.

Problem B. The need to minimize the value of residues through the set of control parameters $\{x_1(t_1),\, x_2(t_1),\, ...,\, x_9(t_1)\}$ estimated at the end of the simulation period:

$$
(16) \quad \min_{\{x_1(t_1),\, x_2(t_1),\, ...,\, x_9(t_1)\}} F(\mathbf{x}),
$$

s.t., $0 \le \mathbf{x} \le \bar{\mathbf{x}}$.

The third SD-model is the typical dynamical Supply-Chain (SC) model and is modified by the inclusion of the Input Flow of Material that is distributed between other flows (Fig. 6).
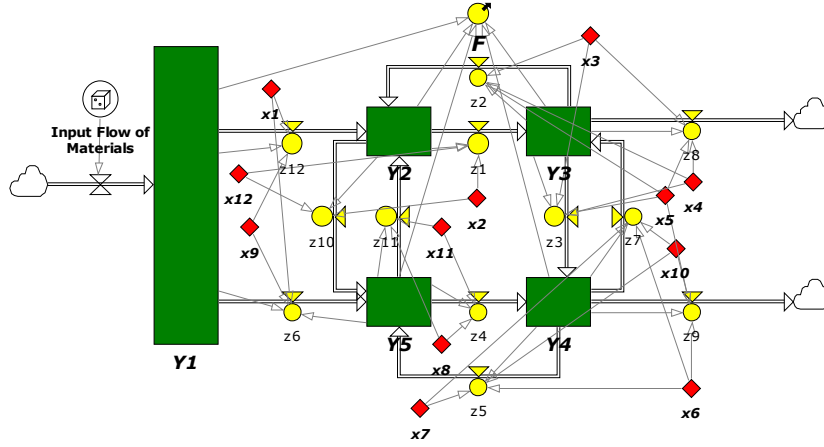
Fig. 6. The implementation of the dynamical SC-model in Powersim

Here, the Input Flows of Materials is defined with the help of the Poisson distribution: $f(\tilde{x}) = \dfrac{\mu^{\tilde{x}}}{\tilde{x}!} e^{-\mu}$. The distribution describes the most realistic dynamics of supply in logistics. At the same time, the objective function (F) of the system is the total residue summarized by all system levels (warehouses $Y1 - Y5$ in Fig. 6).

The model is described by the following system of ordinary differential equations at the interval $t_s \le \xi \le t_{s+1}$:

(17)
$$
\begin{cases}
\dot{y}_1(\xi) = \dfrac{\mu^{\tilde{x}}}{\tilde{x}!} e^{-\mu}, \\
\dot{y}_2(\xi) = z_{12}(\xi) + z_2(\xi) - z_1(\xi) - z_{10}(\xi), \\
\dot{y}_3(\xi) = z_1(\xi) + z_7(\xi) - z_2(\xi) - z_3(\xi) - z_8(\xi), \\
\dot{y}_4(\xi) = z_3(\xi) + z_4(\xi) - z_7(\xi) - z_9(\xi) - z_5(\xi), \\
\dot{y}_5(\xi) = z_3(\xi) + z_4(\xi) - z_7(\xi) - z_9(\xi) - z_5(\xi),
\end{cases}
$$

where $\{z_1(\xi),\ z_2(\xi),\ ...,\ z_{12}(\xi)\}$ are values of input and output flows depending on the values of decision variables $\{x_1(\xi),\ x_2(\xi),\ ...,\ x_{12}(\xi)\}$.

For the considered system, the following single-objective box-constrained optimisation problem can be formulated.

Problem C. The need to minimize the value of residues through the set of control parameters $\{x_1(t_1),\ x_2(t_1),\ ...,\ x_{12}(t_1)\}$ estimated at the end of the simulation period:

(18)
$$
\min_{\{x_1(t_1),\ x_2(t_1),\ ...,\ x_{12}(t_1)\}} F(\mathbf{x}),
$$

s.t., $0 \le \mathbf{x} \le \bar{\mathbf{x}}$.

It has been shown in this work that **Problem A**, **Problem B** and **Problem C** can be solved using the suggested F-RCGA.

4. Results and discussion

The results of optimisation experiments using the first SD-model (**Problem A**) aggregated with F-RCGA (Fig. 2) are presented in Table 2. The simulation period is ten days. The reference values were computed with the help of the well-known quasi-Newton method [10]. The initial values of all system levels $\{y_1(t_1),\ y_2(t_1),\ y_3(t_1),\ y_4(t_1)\} = 100$. The simulation period $t_S = 10$.

Table 2. Results of optimisation experiments using the first SD-model and F-RCGA

Values of upper limits for all decision variables		Decisions	Objective (computed value)	Extremum (reference value)
1	$F(\mathbf{x})$	{0, 0.87, 0.69, 0.73, 1}	0	0
	PT, s	1.8		
100	$F(\mathbf{x})$	{0, 30.71, 67.23, 45.13, 86.66}	0	0
	PT, s	1.9		
1000	$F(\mathbf{x})$	{0, 307.1, 672.32, 451.27, 866.64}	0	0
	PT, s	2.0		
100,000	$F(\mathbf{x})$	{0, 44456, 80496, 24106, 47743}	0	0
	PT, s	2.2		

The results of optimisation experiments using the second SD-model (**Problem B**) aggregated with F-RCGA are presented in Table 3. The simulation period $t_S = 60$.

Table 3. Results of optimisation experiments using the second SD-model and F-RCGA

Values of upper limits of decision variables		Decisions	Objective (computed value)	Extremum (reference value)
1	$F(\mathbf{x})$	{0, 0.59, 0.73, 0, 0.46, 0, 0.45, 0.64, 1.0}	0	0
	PT, s	5.0		
100	$F(\mathbf{x})$	{0, 87.45, 56.97, 0, 50.29, 0, 28.59, 33.66, 33.52}	0	0
	PT, s	5.3		
1000	$F(\mathbf{x})$	{0, 480.52, 769.94, 0, 570.33, 0, 282.88, 585.46, 341.59}	0	0
	PT, s	5.5		
100,000	$F(\mathbf{x})$	{0, 53168, 72692, 0, 72520, 0, 20083, 71475, 35094}	0	0
	PT, s	5.9		

The results of optimisation experiments using the last SD-model are presented in Table 4 and Fig. 7. Because the problem being solved is more complex and the objective function value depends on multiple parameters, the Monte-Carlo method [27, 28] was used for the validation of F-RCGA instead of the quasi-Newton method. Thus, the two scenarios will be considered:

- **Scenario 1.** The Monte-Carlo method is used for identifying the minimum value of the objective function $F(\mathbf{x})$ using about 100 000 simulation experiments. The values of the decision variables are set using the uniform distribution on the interval $[0,\ 1]$.

- **Scenario 2.** F-RCGA is used for identifying the minimum value of the objective function $F(\mathbf{x})$ using about 20 agent-processes.

Table 4. Results of optimisation experiments using the third SD-model and F-RCGA

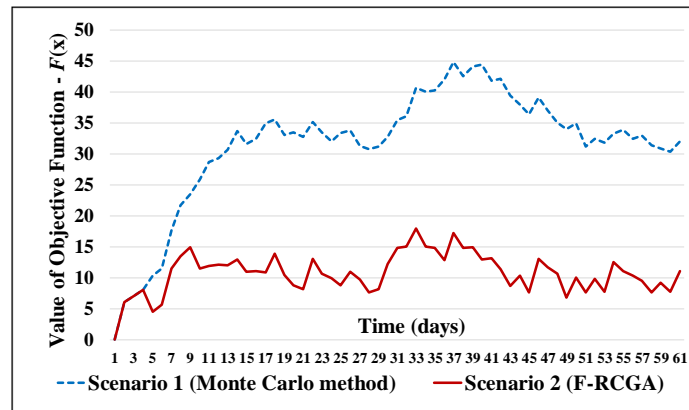| $x_i(t_1)$,	Values of decision variables and an objective function	
$i=1, \ldots, 12$	Scenario 1 (Monte-Carlo method)	Scenario 2 (F-RCGA)
$x_1(t_1)$	0.10	0.00
$x_2(t_1)$	0.92	1.00
$x_3(t_1)$	0.32	0.00
$x_4(t_1)$	0.50	0.96
$x_5(t_1)$	0.40	0.00
$x_6(t_1)$	0.40	0.92
$x_7(t_1)$	0.20	0.46
$x_8(t_1)$	0.50	0.73
$x_9(t_1)$	0.21	0.79
$x_{10}(t_1)$	0.62	0.07
$x_{11}(t_1)$	0.23	0.49
$x_{12}(t_1)$	0.11	0.00
$F(\mathbf{x})$	**31.98**	**11.07**

Fig. 7. Dynamics of the objective function for the third SD-model

The suggested real-coded genetic algorithm with fuzzy control (F-RCGA) has greater performance efficiency than the standard multi-agent genetic algorithm developed for the large-scale black-box single-objective optimisation [6]. This is confirmed by the results optimisation experiments using complex test instances (Table 1 and Fig. 3).

The experiments on three SD-models (Figs 4-6) aggregated with F-RCGA (Fig. 2) showed that the suggested real-coded genetic algorithm with fuzzy control is able to solve appropriate single-objective box-constrained optimisation problems with the required precision and time-efficiency (Tables 2-4). Because well-known Newton and quasi-Newton methods [9, 10] cannot be applied to the complex SD-models described by differential equations with many internal feedbacks, the Monte-Carlo method was used for identifying optimal solutions. However, the suggested F-RCGA is significantly more efficient in comparison (Fig. 7).

Despite the highlighted advantages of F-RCGA, such an algorithm is the most efficient for black-box (derivative-free) optimization when the value of the objective function is computed as a result of the simulation modelling (e.g., using Powersim). In other cases, such methods as the Ant Colony Optimization (ACO) algorithms [29, 30], Particle Swarm Optimization (PSO) [31] and hybrid genetic algorithm with using the Newton method [32] can be more preferable.

5. Conclusion

Developing optimisation algorithms for system dynamics models is an important problem because such systems are characterized by many internal feedback relations and non-liner dependencies. Currently, SD-models are used for managing petroleum companies [5], solving transport problems [11] and optimising supply-chains [6]. At the same time, identifying optimal decisions for such systems can cause difficulties due to the complex landscape of the objective function.

In this work, a new parallel real-coded genetic algorithm with fuzzy control (F-RCGA) was presented and its application in the optimization of SD-model parameters showed high performance efficiency.

As future work, we plan to use the proposed algorithm for the large-scale multi-objective optimisation of parameters of more complex simulations.

R e f e r e n c e s

1. 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 Optimisation. – Knowledge-Based Systems, Vol. **174**, 2019, pp. 103-122.
2. C o n n, A. R., K. S c h e i n b e r g, L. N. V i c e n t e. Introduction to Derivative-Free Optimization. MPS-SIAM Book Series on Optimization. Philadelphia, SIAM, 2009.
3. 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, No 1, pp. 1-2.
4. F o r r e s t e r, J. W. Industrial Dynamics – A Major Breakthrough for Decision Makers. – Harvard Business Review, Vol. **36**, 1958, No 4, pp. 37-66.
5. A k o p o v, A. S. Designing of Integrated System-Dynamics Models for an Oil Company. – International Journal of Computer Applications in Technology, Vol. **45**, 2012, No 4, pp. 220-230.
6. G e, Y., J. B. Y a n g, N. P r o u d l o v e, M. S p r i n g. System Dynamics Modelling for Supply-Chain Management: A Case Study on a Supermarket Chain in the UK. – International Transactions in Operational Research, Vol. **11**, 2004, No 5, pp. 495-509.
7. K e l o h a r j u, R., E. F. W o l s t e n h o l m e. The Basic Concepts of System Dynamics Optimization. – Systems Practice, Vol. **1**, 1988, No 1, pp. 65-86.
8. D a n g e r f i e l d, B. System Dynamics Models, Optimization. – In: R. Meyers, Ed. Encyclopedia of Complexity and Systems Science. New York, NY, Springer, 2013.
9. F l e t c h e r, R. Practical Methods of Optimization. 2nd ed. New York, John Wiley & Sons, 1987.
10. B y r d, R., P. L u, J. N o c e d a l, C. Z h u. A Limited Memory Algorithm for Bound Constrained Optimization. – SIAM Journal on Scientific Computing, Vol. **16**, 1995, No 5, pp. 1190-1208.
11. K h a c h a t r y a n, N. K., A. S. A k o p o v. Model for Organizing Cargo Transportation with an Initial Station of Departure and a Final Station of Cargo Distribution. – Business Informatics, Vol. **1**, 2017, No 39, pp. 25-35.
12. A k o p o v, A. S. Parallel Genetic Algorithm with Fading Selection. – International Journal of Computer Applications in Technology, Vol. **49**, 2014, No 3/4, pp. 325-331.

13. A k o p o v, A. S., M. A. H e v e n c e v. A Multi-Agent Genetic Algorithm for Multi-Objective Optimization. – In: Proc. of IEEE International Conference on Systems, Man and Cybernetics, Manchester: IEEE, 2013, pp. 1391-1395.
14. A k o p o v, A. S., L. A. B e k l a r y a n, A. K. S a g h a t e l y a n. Agent-Based Modelling of Interactions between air Pollutants and Greenery Using a Case Study of Yerevan, Armenia. – Environmental Modelling and Software, Vol. **116**, 2019, pp. 7-25.
15. A k o p o v, A. S., L. A. B e k l a r y a n, A. K. S a g h a t e l y a n. Agent-Based Modelling for Ecological Economics: A Case Study of the Republic of Armenia. – Ecological Modelling, Vol. **346**, 2017, pp. 99-118.
16. D e e p, K., M. T h a k u r. A New Crossover Operator for Real Coded Genetic Algorithms. – Applied Mathematics and Computation, Vol. **188**, 2007, No 1, pp. 895-911.
17. H e r r e r a, F., M. L o z a n o. Gradual Distributed Real-Coded Genetic Algorithms. – IEEE Transactions on Evolutionary Computation, Vol. **4**, 2000, No 1, pp. 43-63.
18. E. Sanchez, T. Shibata, L. A. Zadeh, Eds. Genetic Algorithms and Fuzzy Logic Systems. Vol. **7**. Soft Computing Perspectives. River Edge, NJ, USA, World Scientific Publishing Co., Inc., 1997.
19. K r a m e r, O. A Brief Introduction to Continuous Evolutionary Optimization. – In: Springer Briefs in Computational Intelligence, Springer, 2014.
20. Z i t z l e r, E., L. T h i e l e. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. – IEEE Transactions on Evolutionary Computation, Vol. **3**, 1999, No. 4, pp. 275-271.
21. B e l e v, B., D. D i m i t r a n o v, A. S p a s o v, A. I v a n o v. Application of Information Technologies and Algorithms in Ship Passage Planning. – Cybernetics and Information Technologies, Vol. **19**, 2019, No 1, pp. 190-200.
22. G e o r g i e v a, P. Genetic Fuzzy System for Financial Management. – Cybernetics and Information Technologies, Vol. **18**, 2018, No 2, pp. 22-35.
23. B e k l a r y a n, A. L., A. S. A k o p o v. Simulation of Agent-Rescuer Behaviour in Emergency Based on Modified Fuzzy Clustering. – In: Proc. of International Joint Conference on Autonomous Agents and Multigene Systems, AAMAS, 2016, pp. 1275-1276.
24. D e e p, K., M. T h a k u r. A New Crossover Operator for Real Coded Genetic Algorithms. – Applied Mathematics and Computation, Vol. **188**, 2007, No 1, pp. 895-911.
25. K u m a r, A., K. D e b. Real-Coded Genetic Algorithms with Simulated Binary Crossover: Studies on Multimodal and Multiobjective Problems. – Complex Systems, Vol. **9**, 1995, pp. 431-454.
26. D e e p, K., M. T h a k u r. A New Mutation Operator for Real Coded Genetic Algorithms. – Applied Mathematics and Computation, Vol. **193**, 2007, No 1, pp. 211-230.
27. M e t r o p o l i s, N., S. U l a m. The Monte Carlo Method. – Journal of the American Statistical Association, Vol. **44**, 1949, No 247, pp. 335-341.
28. K a r a i v a n o v a, A., A. A l e x a n d r o v, T. G u r o v, S. I v a n o v s k a. On the Monte Carlo Matrix Computations on Intel MIC Architecture. – Cybernetics and Information Technologies, Vol. **17**, 2017, No 5, pp. 49-59.
29. K u m a r, A., D. K u m a r, S. K. J a r i a l. A Review on Artificial Bee Colony Algorithms and Their Applications to Data Clustering. – Cybernetics and Information Technologies, Vol. **17**, 2017, No 3, pp. 3-28.
30. K u m a r, A., M. T h a k u r, G. M i t t a l. A New Ants Interaction Scheme for Continuous Optimization Problems. – International Journal of Systems Assurance Engineering, Vol. **9**, 2018, No 4, pp. 784-801.
31. R o m a s e v y c h, Y., V. A. L o v e i k i n. Novel Multi-Epoch Particle Swarm Optimization Technique. – Cybernetics and Information Technologies, Vol. **18**, 2018, No 3, pp. 62-74.
32. N o a c k, M. M., S. W. F u n k e. Hybrid Genetic Deflated Newton Method for Global Optimisation. – Journal of Computational and Applied Mathematics, Vol. **325,** 2017, pp. 97-112.