BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **21**, No 3
Sofia • 2021 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2021-0034

Simulation-Based Optimisation for Autonomous Transportation Systems Using a Parallel Real-Coded Genetic Algorithm with Scalable Nonuniform Mutation

Andranik S. Akopov[1,2], *Levon A. Beklaryan*[2], *Armen L. Beklaryan*[1]

[1]*National Research University Higher School of Economics, 20, Myasnitskaya Str., 101978 Moscow, Russian Federation*
[2]*Central Economics and Mathematics Institute of Russian Academy of Sciences, 47, Nachimovski Prosp., 117418 Moscow, Russian Federation*
E-mails: aakopov@hse.ru beklar@cemi.rssi.ru abeklaryan@hse.ru

***Abstract*: *This work presents a novel approach to the simulation-based optimisation for Autonomous Transportation Systems* (*ATS*) *with the use of the proposed parallel genetic algorithm. The system being developed uses GPUs for the implementation of a massive agent-based model of Autonomous Vehicle* (*AV*) *behaviour in an Artificial Multi-Connected Road Network* (*AMCRN*) *consisting of the "Manhattan Grid" and the "Circular Motion Area" that are crossed. A new parallel Real-Coded Genetic Algorithm with a Scalable Nonuniform Mutation* (*RCGA-SNUM*) *is developed. The proposed algorithm* (*RCGA-SNUM*) *has been examined with the use of known test instances and compared with parallel RCGAs used with other mutation operators* (*e.g., standard mutation, Power Mutation* (*PM*)*, mutation with Dynamic Rates* (*DMR*)*, Scalable Uniform Mutation* (*SUM*)*, etc.*)*. As a result, RCGA-SNUM demonstrates superiority in solving large-scale optimisation problems when decision variables have wide feasible ranges and multiple local extrema are observed. Following this, RCGA-SNUM is applied to minimising the number of potential traffic accidents in the AMCRN.*

***Keywords*: *Simulation-based optimisation, autonomous transportation systems, real-coded genetic algorithms, multi-agent systems, scalable nonuniform mutation.*

1. Introduction

In modern times, a growing interest related to the deployment of Autonomous Vehicles (AVs) within the concept of building 'smarter cities' is observed. AVs have many advantages due to the possibilities of Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure communications (V2I) (e.g., [16, 17]), that provide road safety and traffic efficiency. At the same time, there are many limitations, which prevent the wide propagation of AVs, which relate mainly to an increased risk of traffic accidents in manoeuvring and an interaction with various agents of the traffic system (e.g., pedestrians, Manned Vehicles (MVs), etc.). An optimisation of Autonomous Transportation Systems (ATS) allows for improved traffic safety, because the best characteristics of both road networks (e.g., the internal configuration of a traffic area) and AVs parameters (e.g., the velocity in conditions of normal and insufficient visibility) can be provided.

There are many studies in the field of intelligent transportation systems (e.g., [14, 18, 19, 23, 25, 26, 28]) which confirm the particular importance of providing traffic safety in the deployment of AVs. For instance, in works [14] and [25], simulations of artificial road networks related to the "Manhattan Grid" are suggested. Such models are based on the dynamic motion planning of AVs [26], usage of predictive controls to generate an AV's motion locally [19], and simulation of V2V interactions to avoid potential collisions [28], etc. Thus, the main objective function of autonomous transportation systems is minimising the total number of traffic accidents. Such a problem is related to a large-scale simulation-based optimisation problem, where decision-variables that define ATS parameters have wide feasible ranges and the objective function is calculated as a result of simulation modelling.

This paper considers the following simulation-based single-objective optimisation problem:

$$\min F(\mathbf{x}), \quad \text{s.t.} \quad \mathbf{x} = (x_1, x_2, ..., x_n)' \in \Omega, \tag{1}$$

where $\mathbf{x} = (x_1, x_2, ..., x_n)'$ is a decision variable vector defining parameters of the simulated system with dimension n, $\Omega = \prod_{i=1}^{n} [a_i, b_i]$ is the feasible region of the search space ($i = 1, 2, ..., n$ is the index of decision variables), and $F : \Omega \to \mathbb{R}$ is the objective function (the number of traffic accidents) that is computed as a result of the simulation modelling. A similar problem statement is formulated in works [2, 7, 10]. To solve such a problem, various derivative-free methods can be applied, in particular, genetic optimisation algorithms (e.g., [1, 2, 4, 8, 24]), particle swarm optimisation (e.g., [29, 30]), differential evolution and various other techniques. Among these methods, parallel Real-Coded Genetic Algorithms (RCGAs) are the most preferable in solving large-scale optimisation problems when the objective function has a complex relief (e.g., in an availability of multiple local extrema and discontinuities, etc.). This is due to real-coded heuristic operators for both crossover and mutation, which provide the best conditions for seeking optimal decisions in wide feasible ranges, overcoming jamming in the local optima. Moreover, RCGAs allow more effective parallelisation of the heuristic search procedure, providing the best potential decision exchanges between processes through the global population [2]. At the same time, standard mutation operators such as Power Mutation (PM), Mutation with Dynamic Rate (DMR) and Scalable Uniform Mutation (SUM), etc., (e.g., [2, 11, 13, 15]), have some drawbacks. These are caused by limited possibilities of the effective use of computing resources to generate the best potential decisions along iterations. In particular, most of the solutions obtained through the standard mutation operators are highly randomised and remote from a global minimum. The use of SUM

[2] allows for quantifying feasible regions in small equal subranges, making it scalable [6]. Thus, the effectiveness of SUM depends on the number of computational processes involved. However, the transition from SUM towards Scalable NonUniform Mutation (SNUM) can significantly improve the evolutionary searching procedure because it narrows the floating borders of feasible subranges and focuses on clusters with the highest number of potential decisions.

The purpose of this work is to suggest an approach for the simulation-based optimisation for Autonomous Transportation Systems (ATS) with the use of the proposed parallel Real-Coded Genetic Algorithm with Scalable NonUniform Mutation (RCGA-SNUM). The proposed algorithm (RCGA-SNUM) is aggregated with AMCRN through an objective function, such as the total number of traffic accidents. Moreover, it can be applied for optimising parameters of various simulation-based systems, such as environmental agent-based models [3], cargo transportation models [18] and evacuation models (e.g., [1, 9]).

2. A model of an autonomous transportation system's behaviour

The behaviour of AVs interacting with MVs in an Artificial Multi-Connected Road Network (AMCRN) consisting of the "Manhattan Grid" (MG) and "Circular Motion Areas" (CMAs) is considered here (Fig. 1).

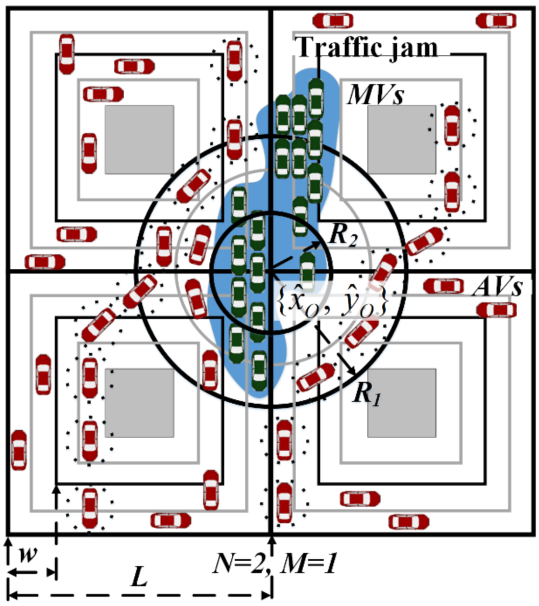

Fig. 1. Artificial multi-connected road network

The AMCRN consists of N rows, N columns and M embedded circular motion areas crossed by the road grids, containing two-lane, one-way roads, which connect adjacent nodes in each section (Fig. 1). The main challenge of such a system is the prevalence of traffic jams caused by traffic density together with arrival intensities of both AVs and MVs in conditions of insufficient vehicle velocities. AVs use the circular motion area to avoid such traffic congestion.

Traffic density is estimated with the help of the hierarchical clustering method [21] using data on all agent-vehicles (AVs and MVs) located in the AMRCN within one-way road lanes. At the same time, each AV chooses the most preferable road, characterised by the lowest traffic density when the agent-vehicle has reached the decision-making area where the MG crosses the CMAs (Fig. 2).

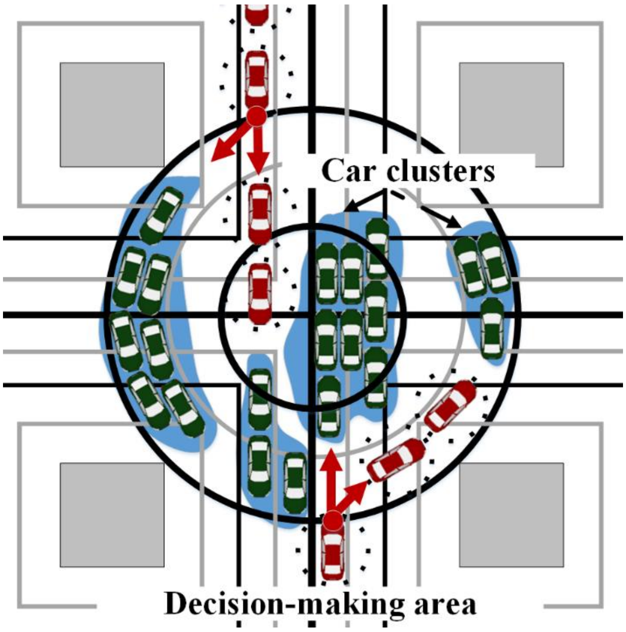

Fig. 2. Choosing the most preferable road based on the traffic density estimation

The proposed multi-connected traffic system (AMCRN) has various control parameters, such as the number of rows and columns in the 'Manhattan Grid', the number of circular motion areas crossing the road grid, and the edge length of each node, etc. All of these parameters impact on the traffic density and the number of potential traffic accidents. The first time such dependencies have been described is in [12], where relationships between many-particle interactions, traffic density and collisions have been studied. Further, in [4], a phenomenological approach have been proposed for the simulation of human crowd behaviour in an emergency. As shown in [4], the main reason for agent collisions are "turbulence" and "crush", caused by panic and high traffic density, respectively (Fig. 3).

Under conditions of high-density traffic surrounding an agent-vehicle, its personal space is firstly compressed, and after reaching a super-dense traffic density, the radius of the personal space increases immediately; pushing edge agents outside of their lanes (Fig. 3). The specification for the radius lengths of the personal space of AVs is shown in Fig. 4. Unlike MVs, AVs have the ability to compress their own personal space even in conditions of super-dense traffic.

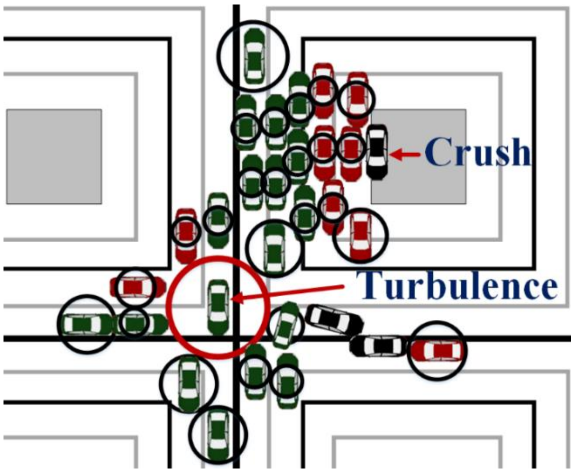

Fig. 3. Illustration of the “turbulence” and “crush” effects

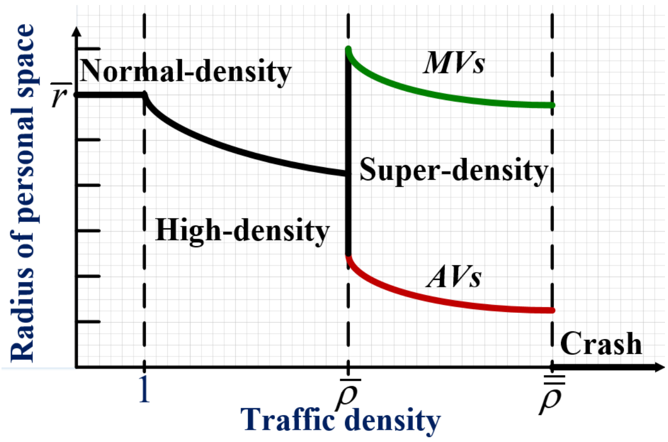

Fig. 4. Personal space of an agent-vehicle depending on traffic density

An abstract description of the proposed model for the behaviour of ATS is presented in a simplified form.

Here:

- T is the set of time moments in the simulation, $|T|$ is the total number of time moments; $t_0 \in T$ and $t_{|T|} \in T$ are initial and finite moments, $t_k \in T$, $k = 0, \ldots, |T|$, is all indices of all time moments;
- $I = \{1, 2, \ldots, |I|\}$ is the set of indices of AVs, where $|I|$ is the total number of AVs;
- $O = \{1, 2, \ldots, |O|\}$ is the set of indices of CMAs, where $|O|$ is the total number of CMAs;

- $\{x_i(t_k),\ y_i(t_k)\},\ i\in I,\ t_k\in T,$ are the coordinates of the *i*-th-AV within AMCRN consisting of MG and CMAs;
- $s_i(t_k)\in\{0,\ 1,\ 2\},\ i\in I,\ t_k\in T,$ is the state of an agent-vehicle: $s_i(t_k)=0$ is the vehicle in the accident state, $s_i(t_k)=1$ is the vehicle in the normal state and located in the MG, $s_i(t_k)=2$ is the vehicle in the normal state and located in the CMA;
- $r_i(t_k),\ i\in I,\ t_k\in T,$ is the radius length of an agent-vehicle's personal space depending on the traffic density surround the agent;
- $v_i(t_k),\ i\in I,\ t_k\in T,$ is the velocity of the *i*-th-AV;
- $\alpha_i(t_k)\in\left\{0,\ \frac{\pi}{2},\ \pi,\ \frac{3\pi}{2},\ 2\pi\right\},\ i\in I,\ t_k\in T,$ is the directional angle of the movement of the *i*-th-AV which is in the state of motion within the MG $(s_i(t_k)=1)$ and moving towards its target direction without meeting any obstacles (e.g., other vehicles);
- $\beta_{io}(t_k),\ i\in I,\ t_k\in T,\ o\in O,$ is the directional angle of movement of the *i*-th -*AV* which is in the state of circular motion within the *o*-th-CMA $(s_i(t_k)=2)$ and moving towards its target direction to avoid traffic jams without meeting any obstacles;
- $\omega_{ij}(t_k),\ i,j\in I,\ i\neq j,\ t_k\in T,$ is the adjusted angle of shifting of the *i*-th -AV regarding its target direction for bypassing the nearest *j*-th-vehicle that is an obstacle;
- $\gamma_{ij}(t_k),\ i,j\in I,\ i\neq j,\ t_k\in T,$ is the rebound angle of the *i*-th-AV from the nearest *j*-th-vehicle if their mutual distance is less than the sum of the radius lengths of their personal spaces;
- $r_i(t_k),\ i\in I,\ t_k\in T,$ is the radius length of the personal space of the *i*-th-AV;
- $d_{io}(t_k),\ i\in I,\ t_k\in T,\ o\in O,$ is the distance between the *i*-th-AV and the centre of the *o*-th -CMA, where the AV is located, with coordinates $\{\hat{x}_o,\ \hat{y}_o\}$;
- $\hat{d}_{ij}(t_k),\ i,j\in I,\ j\neq i,\ t_k\in T,$ is the distance between the *i*-th-AV and the nearest *j*-th-neighbour;
- $\{c_1,\ c_2\}$ are coefficients that define the rebound power at motion within the MG and CMA, respectively.

Thus, the number of potential traffic accidents at moment $t_k,\ t_k\in T,$ is

(2) $$\text{TA}(t_k)=\sum_{i\in I} m_i(t_k),$$

where

$$m_i(t_k)=\begin{cases}1 \text{ if } s_i(t_k)=0 \text{ and } s_i(t_{k-1})\neq 0,\\ 0 \text{ else,}\end{cases}$$
$$i\in I.$$

Thus, the spatial dynamics of the *i*-th-AV $(i \in I)$ in AMCRN without taking into account manoeuvring in lane changing is described by the following system of finite difference equations with variable structure at moment t_k, $t_k \in T$:

$$(3)\quad x_i(t_k) = \begin{cases} x_i(t_{k-1}) + v_i(t_k)\cos\alpha_i(t_{k-1}) \text{ if I is true,} \\ \hat{x}_O + d_{io}(t_{k-1})\cos\beta_{io}(t_{k-1}) \text{ if II is true,} \\ x_i(t_{k-1}) + \cos\left(\alpha_i(t_{k-1}) \pm \omega_{ij}(t_{k-1})\right) + \dfrac{c_1}{\hat{d}_{ij}(t_{k-1})}\cos\gamma_{ij}(t_{k-1}) \text{ if III is true,} \\ x_i(t_{k-1}) + \cos\left(\beta_{io}(t_{k-1}) \pm \omega_{ij}(t_{k-1})\right) + \dfrac{c_2}{\hat{d}_{ij}(t_{k-1})}\cos\gamma_{ij}(t_{k-1}) \text{ if IV is true,} \\ x_i(t_{k-1}) \text{ if V is is true,} \end{cases}$$

$$(4)\quad y_i(t_k) = \begin{cases} y_i(t_{k-1}) + v_i(t_k)\sin\alpha_i(t_{k-1}) \text{ if I is true,} \\ \hat{y}_O + d_{io}(t_{k-1})\sin\beta_{io}(t_{k-1}) \text{ if II is true,} \\ y_i(t_{k-1}) + \sin\left(\alpha_i(t_{k-1}) \pm \omega_{ij}(t_{k-1})\right) + \dfrac{c_1}{\hat{d}_{ij}(t_{k-1})}\sin\gamma_{ij}(t_{k-1}) \text{ if III is true,} \\ y_i(t_{k-1}) + \sin\left(\beta_{io}(t_{k-1}) \pm \omega_{ij}(t_{k-1})\right) + \dfrac{c_2}{\hat{d}_{ij}(t_{k-1})}\sin\gamma_{ij}(t_{k-1}) \text{ if IV is true,} \\ y_i(t_{k-1}) \text{ if V is true.} \end{cases}$$

Here:

I. $\hat{d}_{ij}(t_{k-1}) > r_i(t_{k-1}) + r_j(t_{k-1})$ for all $j \in I$, $i \neq j$, and $s_i(t_{k-1}) = 1$,

II. $\hat{d}_{ij}(t_{k-1}) > r_i(t_{k-1}) + r_j(t_{k-1})$ for all $j \in I$, $i \neq j$, and $s_i(t_{k-1}) = 2$,

III. $\hat{d}_{ij}(t_{k-1}) \le r_i(t_{k-1}) + r_j(t_{k-1})$ for the nearest $j \in I$, $i \neq j$, and $s_i(t_{k-1}) = 1$,

IV. $\hat{d}_{ij}(t_{k-1}) \le r_i(t_{k-1}) + r_j(t_{k-1})$ for the nearest $j \in I$, $i \neq j$, and $s_i(t_{k-1}) = 2$,

V. $s_i(t_{k-1}) = 0$.

The conditions of system (3)-(4) can be modified to take into account manoeuvres in lane changing, turning manoeuvres, emergency deceleration and some other features of AVs.

The following control parameters, which affect the AV's dynamics, are used:

- M, $M = |O|$, is the total number of CMAs crossing the MG with the central coordinates of $\{\hat{x}_o, \hat{y}_o\}$, $o \in O$, belonging to the available crossroads of AMRCN;
- L is the edge length of each node in the MG;
- w is the width of each two-lane one-way road in the MG;
- $\{R_1, R_2\}$ are the radius lengths of external and internal circular motion areas in the CMA, respectively;
- $v_i(t_0)$ is the velocity of the AV that is set up at the initial moment t_0 for all $i \in I$;

- a is the fixed number of AVs to arrive to the AMRCN with the frequency of ψ.

Thus, the following main optimisation problem for autonomous transportation systems based on the movement within the AMCRM can be formulated.

Problem A. Minimisation of the total number of potential traffic accidents with the set of control parameters:

(5)
$$\min_{\{M,\ L,\ w,\ \{R_1,\ R_2\},\ \{\hat{x}_o,\ \hat{y}_o\},\ v_i(t_0),\ a,\ \psi\}} \sum_{t_k=1}^{|T|} \mathrm{TA}(t_k),$$

s.t.,

$$\underline{M} \le M \le \overline{M}\ ,\ \underline{L} \le L \le \overline{L}\ ,\ \underline{w} \le w \le \overline{w}\ ,$$
$$\underline{R}_1 \le R_1 \le \overline{R}_1\ ,\ \ \underline{R}_2 \le R_2 \le \overline{R}_2\ ,\ \hat{x}_o \in \hat{X}\ ,\ \ \hat{y}_o \in \hat{Y}\ ,\ o \in O\ ,$$
$$\underline{v} \le v(t_0) \le \overline{v}\ ,\ \ \underline{a} \le a \le \overline{a}\ ,\ \ \underline{\psi} \le \psi \le \overline{\psi}\ .$$

Here, $\{\underline{M},\ \underline{L},\ \underline{w},\ \underline{R}_1,\ \underline{R}_2,\ \underline{v},\ \underline{a},\ \underline{\psi}\}$ and $\{\overline{M},\ \overline{L},\ \overline{w},\ \overline{R}_1,\ \overline{R}_2,\ \overline{v},\ \overline{a},\ \overline{\psi}\}$ are the lower and upper limits of the control parameters, respectively; $\{\hat{X},\ \hat{Y}\}$ is the set of coordinates of the AMRCN that can be the centres of the CMAs.

Problem A is related to the simulation-based single-objective optimisation problems that can be solved with the use of the proposed genetic algorithm RCGA-SNUM.

3. Real-Coded Genetic Algorithm with Scalable NonUniform Mutation (RCGA-SNUM)

The proposed algorithm RCGA-SNUM is based on the approach suggested earlier in work [2], where the parallel Multi-Agent Real-Coded Genetic Algorithm (MA-RCGA) has been developed. In particular, MA-RCGA uses a combination of different crossover and mutation operators, such as LX, SBX, MSBX, PM and SUM, etc. Among them, the Scalable Uniform Mutation (SUM) operator should be highlighted. This operator enables the splitting of the feasible ranges of decision variables to small equal subranges to implement multiple mutations in narrow segments and avoid potential jamming in local extremums. The effectivity of SUM directly depends on the total number of agent-processes involved (i.e., process cores) and decreases together with extending boundaries of feasible ranges. On the other hand, the potential decisions have a nonuniform distribution within their feasible ranges, which can be taken into account in the scalable multiple mutation. The growth of density of potential decisions that are localised in some segments rise the probability of new potential decisions with the best characteristics to be found in same areas. Therefore, the scalable nonuniform mutation operator is proposed here. The proposed algorithm is based on the paralleled processes of evolving individuals consisting of the vector of decision-variables and the value of the corresponding objective function (i.e., the fitness function value).

At the first stage, clustering of all individuals with their decision variables is performed to find the cluster with the most evolved potential decisions:

(6) $$\tilde{c} = \arg\min_{c \in C} \hat{f}_c(\mathbf{x}_c), \quad \hat{f}_c(\mathbf{x}_c) = \frac{1}{|J_c|} \sum_{j_c=1}^{|J_c|} f(\mathbf{x}_{j_c}).$$

Here:

- $C = \{1, 2, ..., |C|\}$ is the set of indices of decision variable clusters, where $|C|$ is the total number of decision variable clusters computed with the use of the hierarchical clustering algorithm [21];
- $J_c = \{1, 2, ..., |J_c|\}$, $c \in C$, is the set of indices of individuals (potential decisions) belonging to the c-th-cluster, where $|J_c|$ is the total number of appropriate individuals;
- $\mathbf{x}_{j_c}$, $j_c \in J_c$, is the vector of decision variables belonging to the j_c-th-individual of the c-th-cluster;
- $\hat{f}_c(\mathbf{x}_c)$, $c \in C$, is the value of the fitness function equal to the value of the objective function $f(\mathbf{x}_{j_c})$ computed in result of the simulation modelling;
- $\tilde{c} \in C$ is the index of the cluster with the most evolved decisions.

Next, boundary values of the decision variables for the mutation operator can be computed with the use of cluster characteristics. Further, the new notification of indices of decision variables $i \in I$ and indices of agent-processes $k \in K$ will be used to simplify the description of the genetic algorithm:

(7) $$\tilde{a}_{i\tilde{c}} = \begin{cases} \hat{a}_{i\tilde{c}} & \text{if } a_i \le \hat{a}_{i\tilde{c}} \le b_i, \\ a_i & \text{if } \hat{a}_{i\tilde{c}} < a_i, \\ b_i & \text{if } \hat{a}_{i\tilde{c}} > b_i, \end{cases} \qquad \hat{a}_{i\tilde{c}} = \min_{j_{\tilde{c}} \in J_{\tilde{c}}} x_{ij_{\tilde{c}}} + \frac{\max_{j_{\tilde{c}} \in J_{\tilde{c}}} x_{ij_{\tilde{c}}} - \min_{j_{\tilde{c}} \in J_{\tilde{c}}} x_{ij_{\tilde{c}}}}{|G_k||K|} g_k k,$$

(8) $$\tilde{b}_{i\tilde{c}} = \begin{cases} \hat{b}_{i\tilde{c}} & \text{if } a_i \le \hat{b}_{i\tilde{c}} \le b_i, \\ a_i & \text{if } \hat{b}_{i\tilde{c}} \le a_i, \\ b_i & \text{if } \hat{b}_{i\tilde{c}} > b_i, \end{cases} \qquad \hat{b}_{i\tilde{c}} = \tilde{a}_{i\tilde{c}} + \frac{\max_{j_{\tilde{c}} \in J_{\tilde{c}}} x_{ij_{\tilde{c}}} - \min_{j_{\tilde{c}} \in J_{\tilde{c}}} x_{ij_{\tilde{c}}}}{|G_k||K|},$$

where

- $I = \{1, 2, ..., |I|\}$ is the set of indices of decision variables, where $|I|$ is the total number of decision variables;
- $K = \{1, 2, ..., |K|\}$ is the set of indices of interacting agent-processes (e.g., [1, 2]), where $|K|$ is the total number of agent-processes, $k = 1, 2, ..., |K|$ are agent-process numbers;
- $G_k = \{1, 2, ..., |G_k|\}$ is the set of indices of internal iterations of RCGA-SNUM within each k-th-agent-process ($k \in K$), where $|G_k|$ is the total number of internal iterations, $g_k = 1, 2, ..., |G_k|$ are iteration numbers;
- $\{\tilde{a}_{i\tilde{c}}, \tilde{b}_{i\tilde{c}}\}$, $i \in I$, $\tilde{c} \in C$, are the boundary values of decision variables corresponding to the $\tilde{c}$-th- cluster with the most evolved decisions;
- $\{a_i, b_i\}$, $i \in I$, are lower and upper bounds of the feasible range.

Following this, the main parameters of SNUM such as mean and standard deviation are defined with the use of the distribution of cluster decisions:

(9) $$\tilde{m}_{i\tilde{c}} = \frac{1}{|J_{\tilde{c}}|}\sum_{j_{\tilde{c}}=1}^{|J_{\tilde{c}}|} x_{ij_{\tilde{c}}}, \quad \tilde{\sigma}_{i\tilde{c}} = \sqrt{\frac{1}{|J_{\tilde{c}}|}\sum_{j_{\tilde{c}}=1}^{|J_{\tilde{c}}|}(x_{ij_{\tilde{c}}} - \tilde{m}_{i\tilde{c}})^2},$$

where

- $x_{ij_{\tilde{c}}} \in [a_i,\ b_i],\ i \in I,\ j_{\tilde{c}} \in J_{\tilde{c}},\ \tilde{c} \in C,$ is the value of the *i*-th-decision variable of the $j_{\tilde{c}}$-th-individual belonging to the $\tilde{c}$-th-cluster;
- $\tilde{m}_{i\tilde{c}},\ i \in I,\ \tilde{c} \in C,$ is the mean value of the *i*-th-decision variable for the $\tilde{c}$-th-cluster;
- $\tilde{\sigma}_{i\tilde{c}},\ i \in I,\ \tilde{c} \in C,$ is the standard deviation for the *i*-th-decision variable for the $\tilde{c}$-th-cluster.

Finally, new offspring are generated at each internal iteration of RCGA-SNUM using the set of distribution functions in the range of $[\tilde{a}_{i\tilde{c}},\ \tilde{b}_{i\tilde{c}}]$:

(10) $$\hat{x}_{1i} = \begin{cases} l(\tilde{m}_{i\tilde{c}},\ \tilde{\sigma}_{i\tilde{c}},\ \tilde{a}_{i\tilde{c}},\ \tilde{b}_{i\tilde{c}}) \text{ if } \underline{p} < p_{g_k}(0,\ 1) \le \overline{p}, \\ \tilde{l}(\tilde{a}_{i\tilde{c}},\ \tilde{b}_{i\tilde{c}}),\ \overline{p} < p_{g_k}(0,\ 1) \le \overline{\overline{p}}, \\ \tilde{\tilde{l}}(\tilde{m}_{i\tilde{c}}),\ \overline{\overline{p}} < p_{g_k}(0,\ 1), \end{cases}$$

(11) $$\hat{x}_{2i} = \begin{cases} s(\tilde{m}_{i\tilde{c}},\ \tilde{\sigma}_{i\tilde{c}},\ \tilde{a}_{i\tilde{c}},\ \tilde{b}_{i\tilde{c}}) \text{ if } \underline{p} < p_{g_k}(0,\ 1) \le \overline{p}, \\ \tilde{s}(\tilde{a}_{i\tilde{c}},\ \tilde{b}_{i\tilde{c}}),\ \overline{p} < p_{g_k}(0,\ 1) \le \overline{\overline{p}}, \\ \tilde{\tilde{s}}(\tilde{m}_{i\tilde{c}}),\ \overline{\overline{p}} < p_{g_k}(0,\ 1), \end{cases}$$

$$\tilde{c} \in C,\quad i \in I,\quad j_{\tilde{c}} \in J_{\tilde{c}},\quad k \in K,\quad g_k \in G_k.$$

Here:

- $\{\hat{x}_{1i},\ \hat{x}_{2i}\}$ is the pair of offspring that are formed in the result of the mutation;
- $\{l(\tilde{m}_{i\tilde{c}},\ \tilde{\sigma}_{i\tilde{c}},\ \tilde{a}_{i\tilde{c}},\ \tilde{b}_{i\tilde{c}}),\ s(\tilde{m}_{i\tilde{c}},\ \tilde{\sigma}_{i\tilde{c}},\ \tilde{a}_{i\tilde{c}},\ \tilde{b}_{i\tilde{c}})\}$ are the random values with truncated normal distribution with the expected value of $\tilde{m}_{i\tilde{c}}$, standard deviation of $\tilde{\sigma}_{i\tilde{c}}$ and lower and upper limits of $(\tilde{a}_{i\tilde{c}},\ \tilde{b}_{i\tilde{c}})$;
- $\{\tilde{l}(\tilde{a}_{i\tilde{c}},\ \tilde{b}_{i\tilde{c}}),\ \tilde{s}(\tilde{a}_{i\tilde{c}},\ \tilde{b}_{i\tilde{c}})\}$ are the random values with uniform distribution in the range of $[\tilde{a}_{i\tilde{c}},\ \tilde{b}_{i\tilde{c}}]$;
- $\{\tilde{\tilde{l}}(\tilde{m}_{i\tilde{c}}),\ \tilde{\tilde{s}}(\tilde{m}_{i\tilde{c}})\}$ are the random values with exponential distribution with the expected value of $\tilde{m}_{i\tilde{c}}$;
- are the random values with uniform distribution in the range of $[\tilde{a}_{i\tilde{c}},\ \tilde{b}_{i\tilde{c}}]$ with the expected value of $\tilde{m}_{i\tilde{c}}$ and standard deviation of $\tilde{\sigma}_{i\tilde{c}}$;
- $p_{g_k}(0,\ 1)$ is the random value with uniform distribution in the range of $[0,\ 1]$;

- $\underline{p}$, $\overline{p}$, $\overline{\overline{p}}$ are threshold values used in choosing different distribution functions to form offspring.

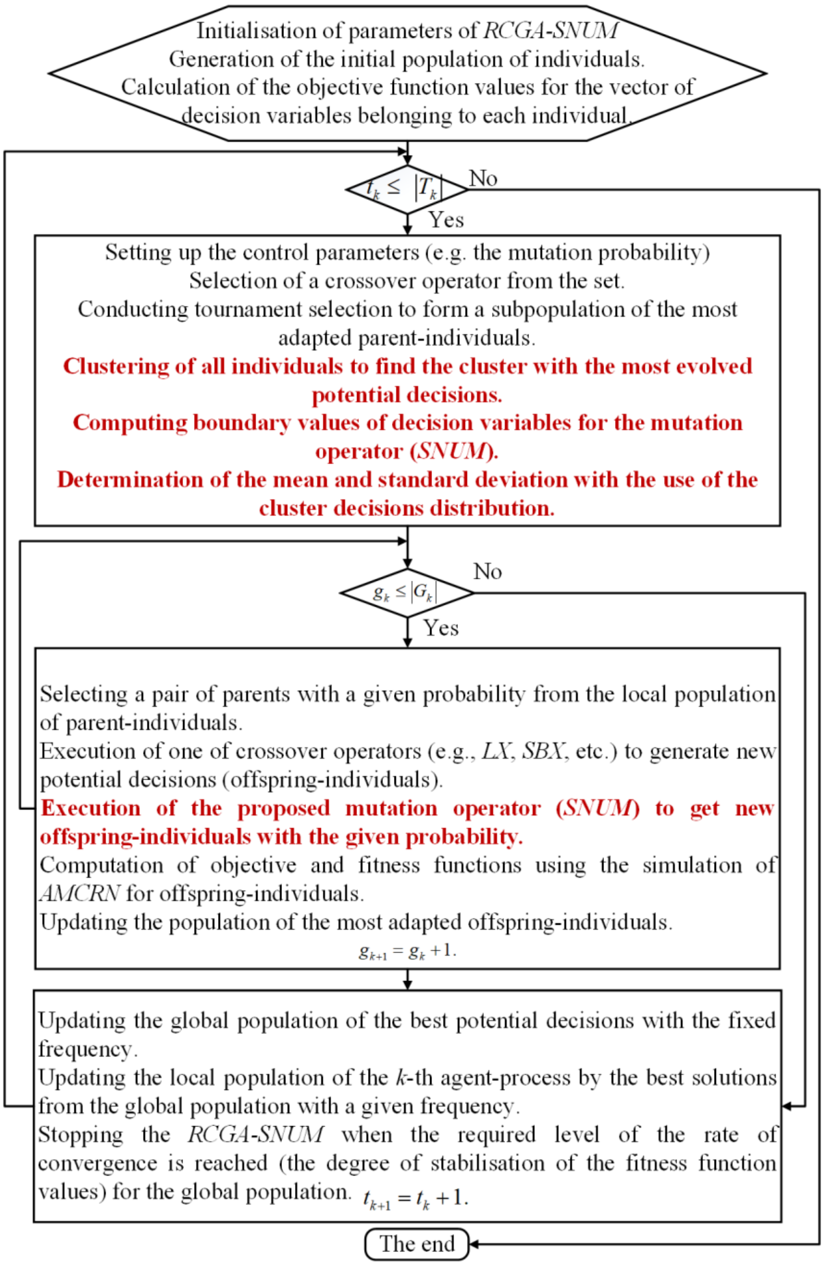

Fig. 5. General scheme of the proposed algorithm (RCGA-SNUM)

The common scheme of RCGA-SNUM implemented at the individual level of each agent-process is shown in Fig. 5. The most important steps of the proposed algorithm are highlighted. In Fig. 5, $T_k = \{1, 2, ..., |T_k|\}$ is the set of indices of external iterations of RCGA-SNUM within each *k*-th-agent-process ($k \in K$), where $|T_k|$ is the total number of external iterations, $t_k = 1, 2, ..., |T_k|$ are iteration numbers. Each agent-process generates new offspring-individuals with the use of one of the crossover operators such as the Laplace Crossover (LX) or the Simulated Binary Crossover (SBX), and the proposed mutation operator SNUM. After that, the best potential decisions are exchanged between agent-processes through the global population.

4. Results and discussion

Table 1 presents the test instances (FT1-FT2) used for examination of RCGA-SNUM and assessment of its efficiency in comparison with single-objective RCGAs based on other mutation operators.

The following parallel RCGAs are used for the validation of five RCGA-SNUM.

- RCGA1 is the parallel real-coded genetic algorithm using the uniform mutation [13] to generate the pair of offspring

$$\hat{x}_{1,i} = q(a_i, b_i), \quad \hat{x}_{2,i} = \tilde{q}(a_i, b_i), \quad i \in I,$$

where $q(a_i, b_i)$, $\tilde{q}(a_i, b_i)$ are the random values with uniform distribution in the range of $[a_i, b_i]$, $\{\hat{x}_{1,i}, \hat{x}_{2,i}\}$ is the pair of offspring generated as a result of the mutation operator;

- RCGA2 is the parallel real-coded genetic algorithm using the Power Mutation (PM) [11]:

$$\hat{x}_{1,i} = \begin{cases} \bar{x}_{1,i} - u(0,\ 1)^{\lambda_{\text{mut}}} (\bar{x}_{1,i} - a_i) \text{ if } (\bar{x}_{1,i} - a_i)/(b_i - a_i) \le h(0,\ 1), \\ \bar{x}_{1,i} - u(0,\ 1)^{\lambda_{\text{mut}}} (b_i - \bar{x}_{1,i}) \text{ if } (\bar{x}_{1,i} - a_i)/(b_i - a_i) > h(0,\ 1), \end{cases}$$

$$\hat{x}_{2,i} = \begin{cases} \bar{x}_{2,i} - u(0,\ 1)^{\lambda_{\text{mut}}} (\bar{x}_{i,2} - a_i) \text{ if } (\bar{x}_{2,i} - a_i)/(b_i - a_i) \le h(0,\ 1), \\ \bar{x}_{2,i} - u(0,\ 1)^{\lambda_{\text{mut}}} (b_i - \bar{x}_{2,i}) \text{ if } (\bar{x}_{2,i} - a_i)/(b_i - a_i) > h(0,\ 1), \end{cases}$$

$$i \in I,$$

where λ_{mut} is the parameter of the mutation operator, $0 < \lambda_{\text{mut}} \le 1$, $h(0,\ 1)$ is the random value with uniform distribution in the range of $[0,\ 1]$, $\{\bar{x}_{1,i}, \bar{x}_{2,i}\}$ is the pair of individuals before the mutation (e.g., it is generated as a result of a crossover);

- RCGA3 is the parallel real-coded genetic algorithm using the Dynamic Mutation Rates (DMR) [27]:

$$\hat{x}_{1,i}(t_k) = \begin{cases} q(a_i, b_i) \text{ if } r(0,\ 1) \le \dfrac{1}{t_k}, \\ \bar{x}_{1,i}(t_k) \text{ if } r(0,\ 1) > \dfrac{1}{t_k}, \end{cases} \qquad \hat{x}_{2,i}(t_k) = \begin{cases} \tilde{q}(a_i, b_i) \text{ if } \tilde{r}(0,\ 1) \le \dfrac{1}{t_k}, \\ \bar{x}_{2,i}(t_k) \text{ if } r(0,\ 1) > \dfrac{1}{t_k}, \end{cases}$$

$$i \in I, \quad t_k \in T_k, \quad k \in K,$$

where $r(0,\ 1)$, $\tilde{r}(0,\ 1)$ are the random values with uniform distribution in the range of $[0,\ 1]$;

Table 1. Test instances for RCGA-SNUM

Name	Title	Objective to be minimised and solution	Feasible ranges
FT1	Rastrigin function	$F(\mathbf{x}) = 10n + \sum_{i=1}^{n}\left(x_i^2 - 10\cos(2\pi x_i)\right)$, $F(0,\ 0,\ ...,\ 0) = 0$	$x_i \in [-5,\ 5]$, $i = 1,\ 2,\ ...,\ n$
FT2	Ackley function	$F(\mathbf{x}) = -20\mathrm{e}^{\frac{1}{5\sqrt{n}}\sum_{i=1}^{n} x_i^2,} - e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)} + e - 20$, $F(0,\ 0,\ ...,\ 0) = 0$	$x_i \in [-10,\ 10]$, $i = 1,\ 2,\ ...,\ n$
FT3	Eggholder function	$F(\mathbf{x}) = -x_1\sin(\sqrt{\lvert x_1 - x_2 - 47\rvert}) -$ $(x_2 + 47)\sin\left(\sqrt{\left\lvert \frac{1}{2}x_1 + x_2 + 47\right\rvert}\right)$, $F(512,\ 404.2319) = -959.6407$	$x_i \in [-512,\ 512]$, $i = 1,\ 2$
FT4	Schaffer’s F6 function	$F(\mathbf{x}) = 0.5 + \frac{\sin^2\left(\sum_{i=1}^{n}\sqrt{x_i^2}\right) - 0.5}{\left(1 + 0.001\sum_{i=1}^{n} x_i^2\right)^2}$, $F(0,\ 0,\ ...,\ 0) = 0$	$x_i \in [-100,\ 100]$, $i = 1,\ 2,\ ...,\ n$
FT5	Lunacek bi-Rastri-gin function	$F(\mathbf{x}) = \min\left[\sum_{i=1}^{n}(x_i - \mu_1)^2,\ d\cdot n + s\sum_{i=1}^{n}(x_i - \mu_2)^2\right] +$ $+10\sum_{i=1}^{n}\left(1 - \cos(2\pi(x_i - \mu_1))\right),$ where $\mu_1 = 2.5$, $\mu_2 = -\sqrt{\frac{\mu_1 - d}{s}}$, $d = 1$, $s = 1 - \frac{1}{2\sqrt{2+20} - 8.2}$, $F(2.5,\ 2.5,\ ...,\ 2.5) = 0$	$x_i \in [-5.12,\ 5.12]$, $i = 1,\ 2,\ ...,\ n$

- RCGA4 is the parallel real-coded genetic algorithm using the directed variation mutation (DVM) [33]:

$$\hat{x}_{1,i} = \mathrm{rand}(\hat{x}_{1,i},\ B_{1,i}),\ \ \hat{x}_{2,i} = \mathrm{rand}(\hat{x}_{2,i},\ B_{2,\ i}),$$

where $\mathrm{rand}(\hat{x}_{1,i},\ B_{1,i}),\ \mathrm{rand}(\hat{x}_{2,i},\ B_{2,i}),\ i \in I,$ are the values randomly generated between the selected individuals $\{\hat{x}_{1,i},\ \hat{x}_{2,i}\}$ and the centres of the first and second neighbouring intervals $(B_{1,i},\ B_{2,i})$ where the potential decisions should be moved. The centres of neighbouring intervals for the *i*-ths-individuals are computed based on the fitness function estimation.

- RCGA5 is the parallel real-coded genetic algorithm using the Scalable Uniform Mutation (SUM) [2]:

$$\hat{x}_{i,1} = l(\tilde{a}_i,\ \tilde{b}_i)\,, \quad \hat{x}_{i,2} = \tilde{l}\,(\tilde{a}_i,\ \tilde{b}_i)\,,$$

$$\tilde{a}_i = a_i + \frac{(b_i - a_i)}{G_k \times K} g_k k\,, \qquad \tilde{b}_i = \tilde{a}_i + \frac{(b_i - a_i)}{G_k \times K}\,,$$

where $l(a_i,\ b_i)$, $\tilde{l}\,(a_i,\ b_i)$ are the random values with uniform distribution in the range of $[a_i,\ b_i]$.

All RCGAs above use the well-known SBX-crossover [13] in combination with its mutation operator.

The performance metrics values computed for the suggested RCGA-SNUM in comparison with the other RCGAs are presented in Table 2.

Table 2. Comparison of RCGA-SNUM with other RCGAs using well-known test instances

RCGAs	Performance metrics	Test instances				
		FT1	FT2	FT3	FT4	FT5
RCGA1	$F(\mathbf{x})$	16.12	0.558	–953.1121	0.4999	621.12
	CR	0.431	0.556	0.651	0.0001	0.0002
	PT, s	99	91	17	981	1531
RCGA2	$F(\mathbf{x})$	13.21	0.431	–954.6753	0.4968	554.21
	CR	0.565	0.668	0.742	0.0011	0.0012
	PT, s	34	31	15	655	451
RCGA3	$F(\mathbf{x})$	11.54	0.332	–955.4224	0.4891	182.55
	CR	0.398	0.456	0.872	0.0025	0.0022
	PT, s	48	45	11	221	364
RCGA4	$F(\mathbf{x})$	5.64	0.221	–957.4312	0.3212	0.4512
	CR	0.2312	0.1231	0.976	0.1321	0.1211
	PT, s	74	68	12	121	212
RCGA5	$F(\mathbf{x})$	2.21	0.115	–957.8813	0.0099	0.0121
	CR	0.1121	0.1967	0.932	0.8731	0.8212
	PT, s	58	63	14	765	932
RCGA-SNUM	$F(\mathbf{x})$	0	0	–959.6401	0.0001	0.0069
	CR	0.9871	0.8542	0.999	0.9999	0.9991
	PT, s	69	65	19	951	1214
Reference value of objective function		0	0	–959.6407	0	0

In conducting the optimisation experiments, the following control parameters have been used for all RCGAs: $|K| = 4$ is the total number of agent-processes, $|G_k| = 10,\ k \in K$ – the total number of internal iterations, $n = 100$ is the number of decision variables in FT1, FT2, FT4, FT5 and $n = 2$ in FT3.

At the same time, the following performance metrics are used: $F(\mathbf{x})$ is the objective function value; CR is the convergence rate that defines as a ratio the deference between finite and initial values of objective functions to the total number of external iterations of RCGA; and PT is the processing time (in s) defined by a period spent on reaching the extremum (i.e., when the convergence of RCGA is reached).

As shown in Table 2, the proposed algorithm RCGA-SNUM performs better in the objective function values and the convergence rate in comparison with other well-known RCGAs based on appropriate mutation operators. At the same time, some increase in the processing time for RCGA-SNUM is caused by more time needed for clustering the population of potential decisions in mutation (i.e., for the SNUM).

Fig. 6 shows the evolutionary dynamics of cluster centroids for RCGA-SNUM in comparison with the most effective RCGAs (e.g., RCGA4 and RCGA5) using FT5 with two decision variables (i.e., the Lunacek bi-Rastrigin Function, where $n = 2$) as the case study.

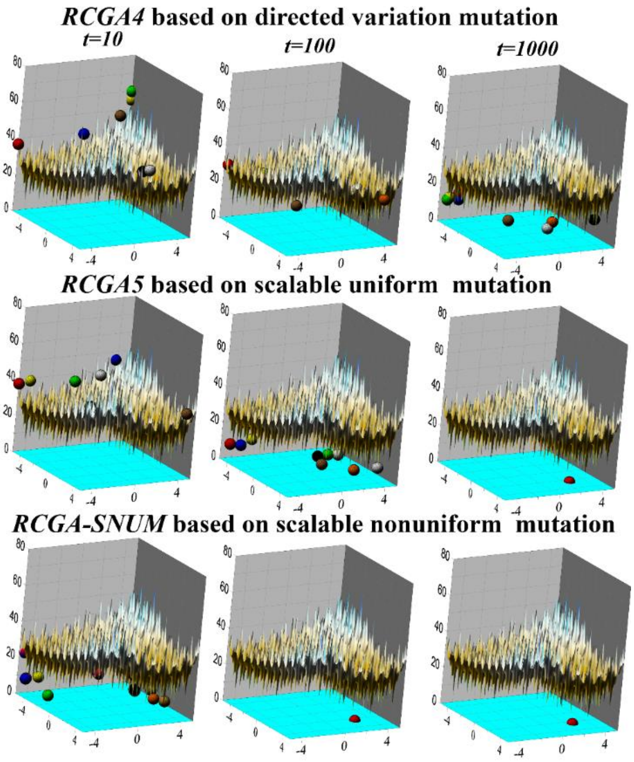

Fig. 6. Evolutionary dynamics of cluster centroids for RCGA-SNUM

As can be seen in Fig. 6, the evolution rate and convergence rate of potential decisions in RCGA-SNUM are significantly better than in other RCGAs, because the Scalable NonUniform Mutation (SNUM) provides an improved shift of new offspring towards clusters with the most evolved solutions, using a combination of different distribution functions (10)-(11). Unlike other well-known heuristic operators such as cluster-based crossover, fuzzy-controlled crossover, directed variation mutation operators, etc., (e.g., [22, 27, 31, 32]), SNUM provides better scaling through embedding the agent-process characteristics to the proposed mutation operator (7)-(8). The operator allows splitting of feasible ranges of decision variables into subranges with unequal lengths, differing by the adaptation level of potential solutions included to appropriate clusters. This leads to improved speed in solving large-scale simulation-based optimisation problems.

After examination of the proposed algorithm RCGA-SNUM, it was applied to solving Problem A and finding the optimal configuration of AMCRN, consisting of 12 rows and columns of the "Manhattan Grid" combined with variable counts of circular motion areas that enable the minimisation of traffic accidents. The computed configuration of AMCRN is presented in Fig. 7.

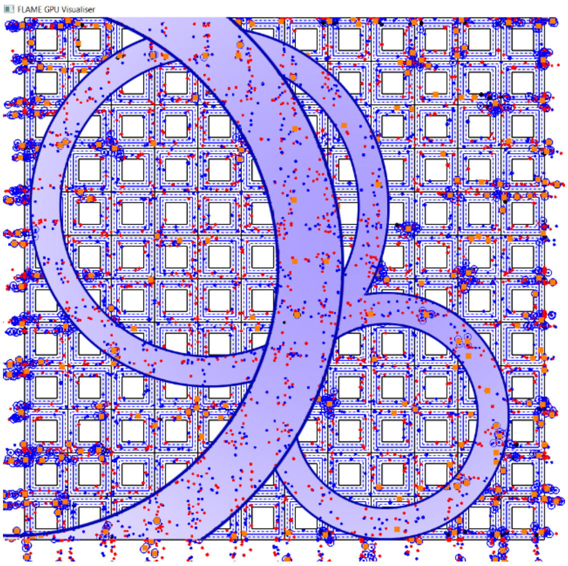

Fig. 7. Optimal configuration of AMCRN obtained with RCGA-SNUM

In Fig. 7, red points are AVs, blue points are MVs and brown rectangles are the centres of car clusters (i.e., traffic jams).

Optimisation experiments have been performed with the use of the CUDA powered supercomputer based on the QUADRO RTX 6000. The simulation model has been implemented with the use of a high-performance agent-based simulation framework FLAME GPU [20], and the AMCRN scheme reconfiguration provides generation of up to 100,000 agent-vehicles as a result of the optimisation achieved with RCGA-SNUM. The total number of potential traffic accidents has been decreased 4 times with the optimisation of AMCRN using RCGA-SNUM.

5. Conclusion

The study aims to develop a novel framework for simulation-based optimisation for autonomous transportation systems using the proposed parallel Real-Coded Genetic Algorithm with Scalable NonUniform Mutation (RCGA-SNUM). The proposed genetic algorithm (Fig. 5) has been examined with the use of known test instances (Table 1) and compared to other parallel RCGAs (Table 2). RCGA-SNUM shows superior behaviour in the accuracy of obtained solutions and the convergence rate

(Fig. 6). After examination of RCGA-SNUM, it was applied to finding the optimal configuration of AMCRN that enables the minimisation of traffic accidents (Fig. 7).

Further research will aim to optimise an artificial multi-connected road network with more complex internal configurations (e.g., multi-layer multi-connected road networks with many heterogeneous agents of the traffic system) to minimise traffic accidents and eliminate traffic congestion.

***Acknowledgments*:** This work was funded by the Russian Foundation for Basic Research (RFBR), Project Number 19-29-06003.

References

1. Akopov, A. S., L. A. Beklaryan, A. L. Beklaryan. Cluster-Based Optimization of an Evacuation Process Using a Parallel Bi-Objective Real-Coded Genetic Algorithm. – Cybernetics and Information Technologies, Vol. **20**, 2020, No 3, pp. 45-63.
2. Akopov, A. S., L. A. Beklaryan, M. Thakur, B. D. Verma. Parallel Multi-Agent Real-Coded Genetic Algorithm for Large-Scale Black-Box Single-Objective Optimisation. – Knowledge-Based Systems, Vol. **174**, 2019, pp. 103-122.
3. Akopov, A. S., L. A. Beklaryan, A. K. Saghatelyan. 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.
4. Akopov, A. S., L. A. Beklaryan. An Agent Model of Crowd Behavior in Emergencies. – Automation and Remote Control, Vol. **76**, 2015, No 10, pp. 1817-1827.
5. Akopov, A. S. Parallel Genetic Algorithm with Fading Selection. – International Journal of Computer Applications in Technology, Vol. **49**, 2014, No 3-4, pp. 325-331.
6. Astsatryan, H., A. Kocharyan, D. Hagimont, A. Lalayan. Performance Optimization System for Hadoop and Spark Frameworks. – Cybernetics and Information Technologies, Vol. **20**, 2020, No 6, pp. 5-17.
7. Audet, C., M. Kokkolaras. Blackbox and Derivative-Free Optimization: Theory, Algorithms and Applications. – Optimization and Engineering, Vol. **17**, 2016, No 1, pp. 1-2.
8. Beklaryan, G. L., A. S. Akopov, N. K. Khachatryan. Optimisation of System Dynamics Models Using a Real-Coded Genetic Algorithm with Fuzzy Control. – Cybernetics and Information Technologies, Vol. **19**, 2019, No 2, pp. 87-103.
9. Beklaryan, A. L., A. S. Akopov. Simulation of Agent-Rescuer Behaviour in Emergency Based on Modified Fuzzy Clustering. – In: Proceedings of the International Joint Conference on Autonomous Agents and Multigene Systems, AAMAS, 2016, pp. 1275-1276.
10. Conn, A. R., K. Scheinberg, L. N. Vicente. Introduction to Derivative-Free Optimization. – MPS-SIAM Book Series on Optimization. Philadelphia, SIAM, 2009.
11. Deep, K., M. Thakur. A New Mutation Operator for Real Coded Genetic Algorithms. – Applied Mathematics and Computation, Vol. **193**, 2007, No 1, pp. 211-230.
12. Helbing, D. Traffic and Related Self-Driven Many-Particle Systems. – Review of Modern Physics, Vol. **73**, 2001, No 4, pp. 1067-1141.
13. Herrera, F., M. Lozano. Gradual Distributed Real-Coded Genetic Algorithms. – IEEE Transactions on Evolutionary Computation, Vol. **4**, 2000, No 1, pp. 43-63.
14. Heywood, P., P. Richmond, S. Maddock. Road Network Simulation Using FLAME GPU. – In: S. Hunold et al., Eds. Proc. of Euro-Par 2015: Parallel Processing Workshops. Euro-Par 2015. Lecture Notes in Computer Science. Vol. **9523**. Cham, Springer, 2015, pp. 430-441.
15. Hong, T., H. Wang. A Dynamic Mutation Genetic Algorithm. – In: Proc. of IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems, Vol. **3**, 1996, pp. 2000-2005.
16. Jameel, F., M. A. Javed, D. T. Ngo. Performance Analysis of Cooperative V2V and V2I Communications Under Correlated Fading. – IEEE Transactions on Intelligent Transportation Systems, Vol. **21**, No 8, pp. 3476-3484.

17. J u r g e n, R. V2V/V2I Communications for Improved Road Safety and Efficiency. SAE International, 2012.
18. 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.
19. K i m, K., P. R. K u m a r. An MPC-Based Approach to Provable System-Wide Safety and Liveness of Autonomous Ground Traffic. – IEEE Transactions on Automatic Control, Vol. **59**, 2014, No 12, pp. 3341-3356.
20. K i r a n, M., P. R i c h m o n d, M. H o l c o m b e, C. L. S h a w n, D. W o r t h, C. G r e e n o u g h. FLAME Simulating Large Populations of Agents on Parallel Platforms. – In: Proc. of 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'10), 2010, pp. 1633-1636.
21. M ü l l n e r, D. Fastcluster: Agglomerative Clustering Routines for R and Python. – Journal of Statistical Software, Vol. **53**, 2013, No 9, pp. 1-18.
22. N a q v i, F. B., M. Y. S h a d, M. S. K h a n. A New Logistic Distribution Based Crossover Operator for Real-Coded Genetic Algorithm. – Journal of Statistical Computation and Simulation, Vol. **91**, 2021, No 4, pp. 817-835.
23. P a d e n, B., M. Č á p, S. Z. Y o n g, D. Y e r s h o v, E. F r a z z o l i. A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles. – IEEE Transactions on Intelligent Vehicles, Vol. **1**, 2016, No 1, pp. 33-55.
24. P r e m a l a t h a, M., V. V i s w a n a t h a n. Course Sequence Recommendation with Course Difficulty Index Using Subset Sum Approximation Algorithms. – Cybernetics and Information Technologies, Vol. **19**, 2019, No 3, pp. 25-44.
25. S a r m a, S. S., K. S i n h a, G. C h a k r a b o r t y, P. B h a b a n i, B. P. S i n h a. Distributed Algorithm for Traffic Dissemination in Manhattan Networks with Optimal Routing-Time. – In: Proc. of Symposium on Applied Computing (SAC'17), 2017, pp. 499-505.
26. S h i l l e r, Z., Y. G w o. Dynamic Motion Planning of Autonomous Vehicles. – IEEE Transactions on Robotics and Automation, Vol. **7**, 1991, No 2, pp. 241-249.
27. T h i e r e n s, D. Adaptive Mutation Rate Control Schemes in Genetic Algorithms. – In: Proc. of Congress on Evolutionary Computation. CEC'02 (Cat. No 02TH8600), Honolulu, HI, USA, Vol. **1**, 2002, pp. 980-985.
28. T o m a s-G a b a r r o n, J., E. E g e a-L o p e z, J. G a r c i a-H a r o. Vehicular Trajectory Optimization for Cooperative Collision Avoidance at High Speeds. – IEEE Transactions on Intelligent Transportation Systems, Vol. **14**, 2013, No 4, pp. 1930-1941.
29. T o s h e v, A. Particle Swarm Optimization and Tabu Search Hybrid Algorithm for Flexible Job Shop Scheduling Problem – Analysis of Test Results. – Cybernetics and Information Technologies, Vol. **19**, 2019, No 4, pp. 26-44.
30. Y u r i y, R., L. V i a t c h e s l a v. A Novel Multi-Epoch Particle Swarm Optimization Technique. – Cybernetics and Information Technologies, Vol. **18**, 2018, No 3, pp. 62-74.
31. Z h a n g, J., H. S. C h u n g, W. L o. Clustering-Based Adaptive Crossover and Mutation Probabilities for Genetic Algorithms. – IEEE Transactions on Evolutionary Computation, Vol. **11**, 2007, No 3, pp. 326-335.
32. Z h o u, Q., Y. L i. Directed Variation in Evolutionary Strategies. – IEEE Transactions on Evolutionary Computation, Vol. **7**, 2003, No 4, 356-366.

Received: 11.05.2021; Accepted: 15.06.2021 (fast track)