

# Genetic Optimization of Multidimensional Technological Process Reliability

Alexander Rotshtein

Department of Management and Industrial Engineering,  
Jerusalem College of Technology – Machon Lev, Jerusalem, Israel

Serhiy Shtovba

Department of Computer-Based Information and Management Systems,  
Vinnitsa National Technical University, Vinnitsa, Ukraine

## 9.1 Introduction

A technological process (TP) is considered multidimensional if a number of defects of diverse types occur, and they are detected and corrected simultaneously within the process execution [4, 5]. Quality of the TP is estimated by the probability of output zero-defects as well as by the probabilities of zero-defects for each of the defect types. The tasks of TP-optimization involve the choice of such a process structure that will provide the necessary output level of product quality given some certain cost limits [5]. The typical example of such optimization tasks is optimal choice of multiplicity of control-retrofit procedures in a TP. This particular optimization problem is studied in this article.

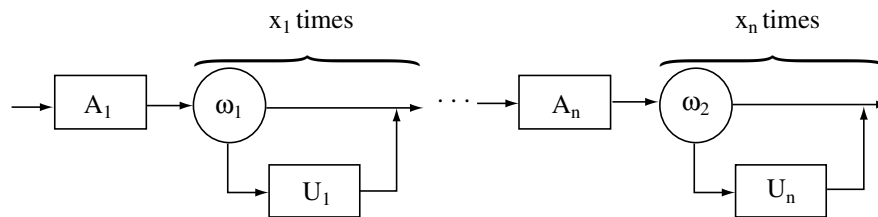
An initial perspective would view this problem as one that may be solved by using known mathematical programming methods. However, taking into account that defects of many diverse types increase the dimensionality of the state space. It means that using classical mathematical programming techniques becomes impractical. Therefore, in this article, the task of TP optimization is solved by using genetic algorithms (GA) [2, 3], which allow to find nearly global optimal solution, and additionally do not require much mathematical backgrounds about optimization. Principal

distinction of GA methods from classical ones is in the fact that they do not use the notion of a derivative while choosing search direction and they are based on crossover, mutation and selection operations.

The first attempt of using a GA in order to solve this nonlinear optimization problem has been proposed in [6]. In that work a simple GA manages to find out the optimal solutions but the whole process proves to be very time-consuming. In this paper we discuss how to improve that GA for catching the optima quickly.

## 9.2 Statements of the Problems

Let's consider a discrete technological process (Fig. 1). The process consists of sequential performing of  $n$  working operation  $A_1, A_2, \dots, A_n$ . Checking procedure  $\omega_i$  for  $x_i$ -times inspects quality of working operation  $A_i$ ,  $i = \overline{1, n}$ . Retrofit procedure  $U_i$  corrects defects detected by checking procedure  $\omega_i$ .



**Fig. 1.** A discrete technological process with multiple checking-retrofit procedures.

Let us introduce the following notations:

- $m$  is the number of diverse types of defects;
- $n$  is the number of working technological operations in TP;
- $x_i \in \{0, 1, 2, \dots\}$  is a number of checking-retrofit procedures after  $i$ -th working operation,  $i = \overline{1, n}$ ;
- $\mathbf{X} = (x_1, x_2, \dots, x_n)$  is a vector denoting controlled variables, used to determine the TP structure;
- $p^1(\mathbf{X})$  is the probability of executing process  $\mathbf{X}$  without any defect;
- $p_j^0(\mathbf{X})$  is the probability of executing process  $\mathbf{X}$  with defect of type  $j$ ;
- $C(\mathbf{X})$  is cost (or other resource) required for process  $\mathbf{X}$  execution.

According to [6], the problem consists of finding that  $\mathbf{X}$ , for which

$$\left. \begin{aligned} & p^1(\mathbf{X}) \rightarrow \max \\ \text{subject to } & C(\mathbf{X}) \leq C^* \text{ and } p_j^0(\mathbf{X}) \leq q_j, \quad j = \overline{1, m} \end{aligned} \right\}, \quad (1)$$

or

$$\left. \begin{aligned} & C(\mathbf{X}) \rightarrow \min \\ \text{subject to } & p^1(\mathbf{X}) \geq P^* \text{ and } p_j^0(\mathbf{X}) \leq q_j, \quad j = \overline{1, m} \end{aligned} \right\}, \quad (2)$$

where  $q_j$  is the admissible probability threshold of the  $j$ -th type defect on process output,  $j = \overline{1, m}$ ;

$C^*$  is the admissible cost threshold of the TP;

$P^*$  is the admissible threshold for zero-defect execution of the TP.

Tasks (1) and (2) look like as a nonlinear multidimensional knapsack problem. The TP is similar to the knapsack, checking-retrofit procedures correspond to the items, probabilities  $p_j^0(\mathbf{X})$  are analogous to geometrical constrains of the knapsack. The linear knapsack problem is NP-hard [1], hence problems (1) and (2) are also NP-hard.

### 9.3 Models of Multidimensional Technological Process Reliability

The relations connecting reliability figures with TP parameters [4, 5] are described below. The reliability figures of working operation  $A$  and retrofit  $U$  are listed as follows:

$$\mathbf{P}_A = \begin{pmatrix} p_A^1 & p_A^{0_1} & \dots & p_A^{0_m} \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix}, \quad \mathbf{P}_U = \begin{pmatrix} 1 & 0 & \dots & 0 \\ v_U^{1_1} & v_U^{0_1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ v_U^{1_m} & 0 & \dots & v_U^{0_m} \end{pmatrix},$$

where  $p_A^1$  is the probability of zero-defect execution of operation  $A$ ;

$p_A^{0_j}$  is the probability of execution of operation  $A$  with  $j$ -th type of defect,  $j = \overline{1, m}$ ;

$v_U^j (v_U^0)$  is the probability of correcting the  $j$ -th type of defect during the execution of retrofit  $U$ ,  $j = \overline{1, m}$ .

The probabilities of false alarm errors and defect loss errors when a checking procedure  $\omega$  is carried out, are represented by the following matrices:

$$\mathbf{K}_\omega^1 = \begin{pmatrix} k_\omega^{11} & 0 & \dots & 0 \\ 0 & k_\omega^{01_1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & k_\omega^{01_m} \end{pmatrix}, \quad \mathbf{K}_\omega^0 = \begin{pmatrix} k_\omega^{10} & 0 & \dots & 0 \\ 0 & k_\omega^{00_1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & k_\omega^{00_m} \end{pmatrix},$$

where  $k_\omega^{11} (k_\omega^{10})$  is the probability of the right (wrong) decision made about the absence of defects during the checking procedure  $\omega$ ;

$k_\omega^{01_j} (k_\omega^{00_j})$  is the probability of detecting defects of type  $j$  during the checking procedure  $\omega$ ,  $j = \overline{1, m}$ .

The costs (or other resources) required for the execution of the working operation  $A$ , the checking procedure  $\omega$ , and the retrofit  $U$ , are denoted by  $c_A$ ,  $c_\omega$ , and  $c_U$ , respectively.

A model of a working operation with  $x$ -multiple checking is employed in this paper. This operation is defined as follows [5]: a working operation with  $x$ -multiple checking is a working operation  $A$  in which, it is carried out an  $x$  times checking  $\omega$  and retrofit  $U$  (if defects are detected). The output reliability figures on this operation are calculated by the following iterative scheme:

$$\left. \begin{aligned} \mathbf{P}_A^{<x>} &= \mathbf{P}_A^{<x-1>} \cdot (\mathbf{K}_\omega^1 + \mathbf{K}_\omega^0 \cdot \mathbf{P}_U) \\ \mathbf{P}_A^{<0>} &= \mathbf{P}_A \end{aligned} \right\}, \tag{3}$$

$$\left. \begin{aligned} c_A^{<x>} &= c_A^{<x-1>} + c_\omega + \sigma^{<x>} \cdot c_U \\ c_A^{<0>} &= c_A \end{aligned} \right\}, \tag{4}$$

where  $\sigma^{<x>} = \sum_{j=1}^m \left( k_\omega^{01_j} p_A^{<x-1>} + k_\omega^{00_j} p_A^{0_j <x-1>} \right)$  is probability of going

to retrofit  $U$  after the  $x$ -th checking.

The reliability figures of the whole TP denoted as  $\mathbf{X}$  are calculated as follows:

$$P(\mathbf{X}) = \prod_{i=1,n} P_{A_i}(x_i), \quad C(\mathbf{X}) = \sum_{i=1,n} c_{A_i}(x_i). \quad (5)$$

## 9.4 Basic Notions of Genetic Algorithms

One of the universal approaches for solving NP-hard combinatorial optimization tasks is Genetic Algorithms (GA). GA represents a stochastic method of optimization based on the mechanisms of natural selection acting in living nature [2, 3]. The notions of chromosome, gene, allele, and population constitute the base of GA; and classical optimization theory terms of decision variables vector, decision variable, value of decision variable, and decision set can be brought into correspondence with them.

The basic operations of GA are crossover, mutation and selection:

*Crossover* represents an operation on two parents-chromosomes yielding two offspring-chromosomes, each of which inherits some genes from parents-chromosomes.

*Mutation* is a random gene modification.

*Selection* represents itself as some procedure of population formation from the most adapted chromosomes according to its fitness function values.

Optimization using genetic algorithms consists of performing such a sequence of steps.

- 1<sup>0</sup>. Carrying out genetic coding of decisions variants.
- 2<sup>0</sup>. Generation of an initial population.
- 3<sup>0</sup>. Random way choosing parents and provision of crossover.
- 4<sup>0</sup>. Random way choosing chromosome and provision of mutation.
- 5<sup>0</sup>. Evaluate the values of fitness function for new chromosomes.
- 6<sup>0</sup>. Fixing the best decision.
- 7<sup>0</sup>. Making selection (new population formation) taking into account the fitness function values.
- 8<sup>0</sup>. Repeat steps 3<sup>0</sup>-6<sup>0</sup> as many times as necessary.
- 9<sup>0</sup>. Decoding of the decision.

## 9.5 Genetic Algorithm for Multidimensional Technological Process Optimization

To speed up typical GA we used:

- a smart procedure for the generation of the proper initial population;

- a fast algorithm calculating the reliability figures of the whole TP;
- a specific adaptive fitness function;
- appropriate selection scheme.

The features of the proposed GA are listed below.

### 9.5.1 Genetic Coding of Variants

A TP-variant can be represented by a chromosome that contains  $n$  -genes:  $\mathbf{X} = (x_1, x_2, \dots, x_n)$ , where genes correspond to the controlled variables for optimizing tasks (1) or (2).

### 9.5.2 Initial Population

Typical initialization generates chromosomes with random uniform distributed alleles. In proposed initialization the allele distribution is depends upon efficiency of checking-retrofit procedures. It allows the creation of a population with high quality chromosomes. They are located into the feasible solution area or in its vicinity. Furthermore, the chromosomes have a low value of  $C$ .

For generating good chromosomes we use the so-called gradient of the checking-retrofit procedure. The gradient  $\mathbf{g}_i$  of the checking-retrofit procedure with number  $i$  indicates a relatively efficient factor of impacting this procedure into the TP [5]. The gradient is calculated as follows [7]:

$$\mathbf{g}_i = \frac{\mathbf{P}(\mathbf{X}, x_i = x_i + 1) - \mathbf{P}(\mathbf{X})}{C(\mathbf{X}, x_i = x_i + 1) - C(\mathbf{X})}.$$

Our smart initialization consists of the following 4 steps.

$I^0$ . Predict the approximate number ( $N$ ) of checking-retrofit procedures in the TP by the following formulae:

for problem (1):

$$N = \frac{C^* - C(\mathbf{X}_0)}{\tilde{c}_\omega}, \quad (6)$$

for problem (2):

$$N = \left( \max \left( \frac{p^* - p^1(\mathbf{X}_0)}{\tilde{g}_{11} \cdot \tilde{c}_\omega}, \frac{q_1 - p_1^0(\mathbf{X}_0)}{\tilde{g}_{12} \cdot \tilde{c}_\omega}, \dots, \frac{q_m - p_m^0(\mathbf{X}_0)}{\tilde{g}_{1m+1} \cdot \tilde{c}_\omega} \right) \right)^{1 + \frac{m}{10}}, \quad (7)$$

where  $\mathbf{X}_0 = (0, 0, 0, \dots)$  is the initial TP-variant without a checking-retrofit procedure;  $\tilde{c}_\omega = \frac{1}{n} \sum_{i=1, n} c_{\omega_i}$  is the mean checking cost;  $\tilde{\mathbf{g}} = \frac{1}{n} \sum_{i=1, n} \mathbf{g}_i$  is the mean gradient;

Exponential factor  $\left(1 + \frac{m}{10}\right)$  in (7) is roughly takes into account the nonlinear dependence upon the probability of zero-defect and the cost.

2<sup>o</sup>. Calculate the statistical expectation of multiplicity of the  $i$ -th checking-retrofit procedure:

$$M_i = \frac{g_{11i}}{\sum_{j=1, n} g_{11j}} N, \quad i = \overline{1, n}.$$

3<sup>o</sup>. Find the probabilistic distribution for each checking-retrofit multiplicity from the following systems:

$$\begin{cases} 0 \cdot prob_0(i) + 1 \cdot prob_1(i) + 2 \cdot prob_2(i) + \dots = M_i \\ \frac{prob_1(i)}{prob_2(i)} = \frac{prob_2(i)}{prob_3(i)} = \frac{prob_3(i)}{prob_4(i)} = \dots = const, \quad i = \overline{1, n}, \\ prob_0(i) + prob_1(i) + prob_2(i) + \dots = 1 \end{cases}$$

where  $prob_k(i)$  denotes probability that the multiplicity of  $i$ -th checking-retrofit procedure equals  $k \in \{0, 1, 2, \dots, \overline{x_i}\}$ ;

$\overline{x_i}$  is an a priori defined upper bound of the multiplicity of the  $i$ -th checking-retrofit procedure;

$const > 1$  is a heuristically defined constant. In our case it equals 10.

The second equation in the system describes the fact that a double check occurs more seldom than a single check in the industrial TP, and a triple check happens more rarely than a double check etc.

4<sup>o</sup>. Generate *pop\_size* chromosomes according to the found probabilistic distributions of the checking-retrofit multiplicities.

### 9.5.3 Crossover and Mutation

A uniform crossover with one cutting point and single-gene mutation [2, 3] are employed. After the mutation the value of gene  $x_i$  should be put into interval  $[0, \overline{x_i}]$ .

### 9.5.4 Fitness Function

The following adaptive fitness function is used for problem (1):

$$F(\mathbf{X}) = \begin{cases} p^1(\mathbf{X}), & \text{if } \mathbf{X} \text{ is a feasible solution} \\ p^1(\mathbf{X}) \cdot (1 - D(\mathbf{X})), & \text{otherwise} \end{cases},$$

where  $D(\mathbf{X})$  is the following penalty function:

$$D(\mathbf{X}) = \frac{1}{m+1} \cdot \left( \max \left( 0, \frac{C(\mathbf{X}) - C^*}{C^*} \right) + \sum_{j=1}^m \left( \frac{\Delta b_j(\mathbf{X})}{\Delta b_j^{\max}} \right)^\alpha \right),$$

where  $\alpha > 0$  is a factor of importance of avoidance the defects;

$\Delta b_j(\mathbf{X}) = \max(0, p_j^0(\mathbf{X}) - q_j)$  means violation of  $j$ -th constraint by chromosome  $\mathbf{X}$ ,  $j = \overline{1, m}$ ;

$\Delta b_j^{\max} = \frac{\max}{p=1, \text{pop\_size}} (\Delta b_j(\mathbf{X}_p))$  is max-violation of constraint on  $q_j$

in the current population. One can associate  $\Delta b_j^{\max}$  with population inferiority.

The following adaptive fitness function is used for problem (2):

$$F(\mathbf{X}) = \begin{cases} \frac{1}{C(\mathbf{X})}, & \text{if } \mathbf{X} \text{ is a feasible solution} \\ \frac{1 - D(\mathbf{X})}{C(\mathbf{X})}, & \text{otherwise} \end{cases},$$

where  $D(\mathbf{X})$  is the following penalty function:

$$D(\mathbf{X}) = \frac{1}{m+1} \cdot \left( \max \left( 0, \frac{P^* - p^1(\mathbf{X})}{P^*} \right) + \sum_{j=1}^m \left( \frac{\Delta b_j(\mathbf{X})}{\Delta b_j^{\max}} \right)^\alpha \right).$$

The penalties for the same violations are different for various populations. The dependence of the penalty values upon population quality allows to separate good and poor solutions during the selection.



### 9.5.5 Fast Calculation of the Reliability

A profile of the GA shows that the most time-consuming part is the set of calculations by formulae (3) – (4). To speed up the process, we calculate in advance and store in memory the quantities  $\mathbf{P}_{A_i}(x_i)$  and  $c_{A_i}(x_i)$  for all possible values of  $x_i \in \{0, 1, 2, \dots, \bar{x}_i\}$ ,  $i = \overline{1, n}$ . Accordingly, for calculating the reliability figures of the whole TP we simply apply formula (5) for the corresponding (already computed) values of  $\mathbf{P}_{A_i}(x_i)$  and  $c_{A_i}(x_i)$ .

### 9.5.6 Selecting Schemes

We examined 3 types of selection: 1) roulette wheel with elitism strategy on whole population, 2) roulette wheel with elitism strategy on truncated population, and 3) tournament selection. The first selection acts by the following scheme: 1) find the chromosome with the highest fitness and the chromosome with the highest fitness among the feasible ones and include them into the new population; 2) add the remaining chromosomes via the roulette wheel process [2]. In the second selection only fraction of best chromosomes can be picked up for a new population [3]. After this population truncation the selection acts as the roulette wheel with elitism strategy. In the third selection each individual in new population is defined as a winner of competition among some number of randomly chosen chromosomes [3].

## 9.6 Computational Experiments

The GA for problem (1) was tested on two data sets: A\_4\_1 and A\_4\_2. The GA for problem (2) was tested on two data sets: B\_4\_1 and B\_4\_2. The data sets are available at [www.ksu.vstu.vinnica.ua/shtovba/benchmark](http://www.ksu.vstu.vinnica.ua/shtovba/benchmark). The data sets correspond to a TP with 4 diverse types of defects. The number of potential checking operations varies from 20 to 120. In data sets A\_4\_1 and B\_4\_1 the avoidance of one type of defects is the most important task, according to the very low level of the admissible probability threshold for it. All the types of defects had approximately the same importance in data sets A\_4\_2 and B\_4\_2. We restricted upper bound of checking multiplicity as follows:  $\bar{x}_i = 4$  for tasks A\_4\_1 and A\_4\_2, and  $\bar{x}_i = 5$  for tasks B\_4\_1 and B\_4\_2,  $i = \overline{1, n}$ .

Test results are shown in Table 1 and Table 2. As an alternative optimization routine we used the greedy method [7] with fast gradient computing. Its basic idea is similar to the knapsack problem greedy solving. The greedy method is an iterative algorithm with two epochs. On the first epoch, it increases the number of checking-retrofit procedures with maximal gradient iteratively. On the second epoch, the greedy algorithm tries to backtrack the solution into the feasible area.

**Table 1.** Probabilities  $p^1(\mathbf{X})$  for optimal solutions (boldface font points to unfeasible solutions).

Dimension (n)	Tasks A_4_1		Tasks A_4_2	
	GA	Greedy search	GA	Greedy search
20	0.9693	<b>0.9652</b>	0.9546	0.9539
40	0.9425	0.9423	0.9233	0.9225
60	0.9309	<b>0.9254</b>	0.9040	0.9037
80	0.8697	0.8689	0.8649	0.8642
100	0.8548	<b>0.8524</b>	0.8372	0.8363
120	0.8482	0.8479	0.8091	0.8080

**Table 2.** Costs  $C(\mathbf{X})$  for optimal solutions.

Dimension (n)	Tasks B_4_1		Tasks B_4_2	
	GA	Greedy search	GA	Greedy search
20	199.67	203.02	153.27	154.37
40	451.97	460.99	322.35	325.25
60	531.88	540.25	478.45	482.24
80	708.40	719.90	697.51	700.97
100	975.04	992.01	859.29	860.59
120	1105.15	1118.06	958.92	962.69

The GA provides better solutions than the greedy method, especially for tasks A\_4\_1 and B\_4\_1. Fig. 2 and Fig. 3 show a satisfaction of the constraints by solutions found by the GA and by the greedy method. In these figures y-axes correspond to the following factor of relative satisfaction:

$$\text{for } C(\mathbf{X}): \quad \psi(\mathbf{X}) = 100\% \cdot (C^* - C(\mathbf{X})) / C^*,$$

$$\text{for } p_j^0(\mathbf{X}): \quad \psi(\mathbf{X}) = 100\% \cdot (q_j - p_j^0(\mathbf{X})) / q_j, \quad j = \overline{1, m}.$$

A negative value of  $\psi(\mathbf{X})$  means, that solution  $\mathbf{X}$  does not satisfy a constraint. There are 3 such solutions, found by the greedy search for task B\_4\_1 with  $n=20$ ,  $n=60$ , and  $n=100$ .

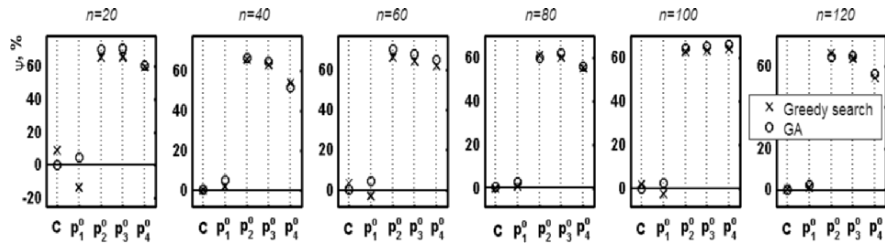


Fig. 2. A satisfaction of constrains for tasks B\_4\_1 by optimal solutions, found by the GA and by the greedy search.

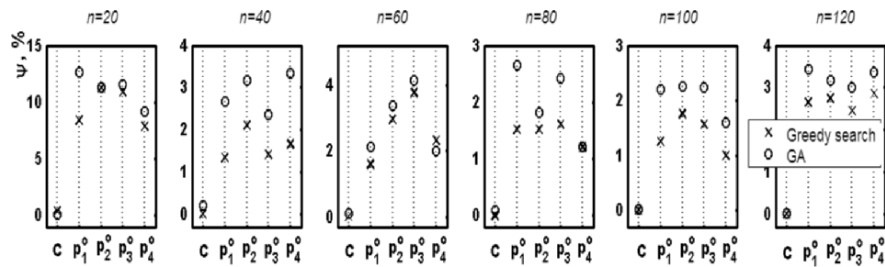


Fig. 3. A satisfaction of constrains for tasks B\_4\_2 by optimal solutions, found by the GA and by the greedy search.

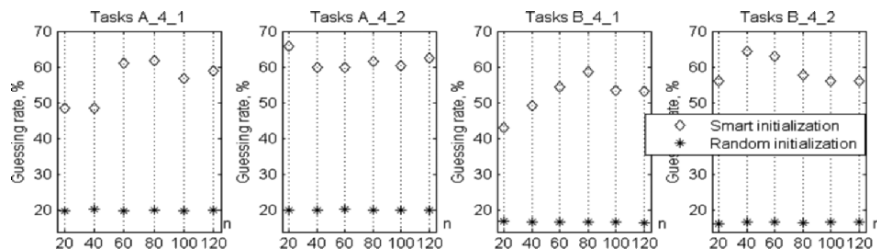


Fig. 4. Matching optimal gene values by random and proposed initializations (average means out of 1000 generations).

Fig. 4 compares random and proposed smart chromosome initializations by the criterion of guessing the optimal gene values. The random initialization produces chromosomes that have about 15-20% genes with the optimal values. The guessing rate of the smart initialization is about 50-60%. Such improvement of the initial population more than doubles the speed of optimization. Fig. 5 shows that the proposed models (6) – (7) predict satisfactorily the number of checking-retrofit procedures.

Profiles of GA with various procedures of calculations of reliability figures are shown on Fig. 6. It is clear that the proposed procedure of calculations of reliability figures accelerates genetic search several fold.

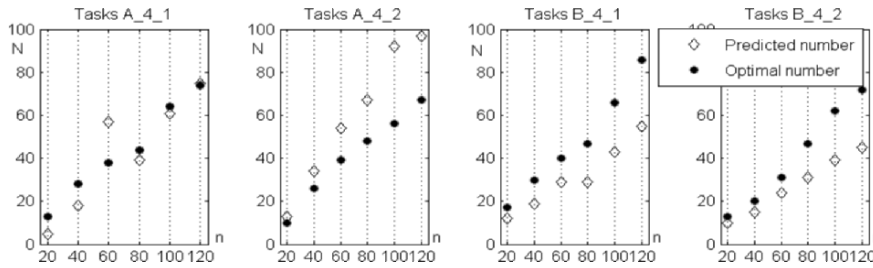


Fig. 5. Predicted and optimal numbers of checking-retrofit procedures.

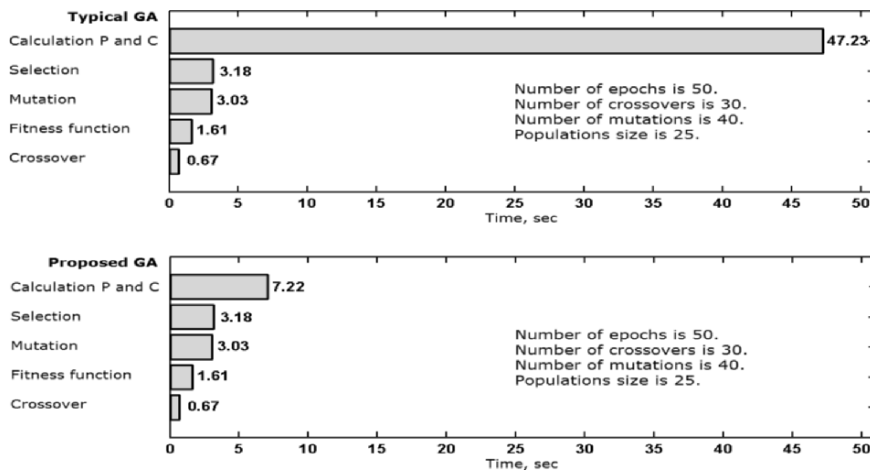


Fig. 6. Profiles of the GA with typical and proposed calculations of reliability figures (task A\_4\_1, n=20).

Fig. 7 compares average dynamics of the GAs with different selection schemes for task B\_4\_2 with n=20. All the selections were tested on 10 various initial populations. For every initial population the GAs run 10 times. The left part of Fig. 7 shows dynamics the optimization in cases of various thresholds  $\tau$  of the truncation selection. The threshold means a percentage of chromosomes, rejected from the population before a roulette wheel spinning. Good dynamics takes place when  $\tau \approx 50\%$ . The right part of Fig. 7 shows dynamics the optimization in cases of various number participants  $t$  in the competition. Good dynamics takes place when  $t \approx 6$ .

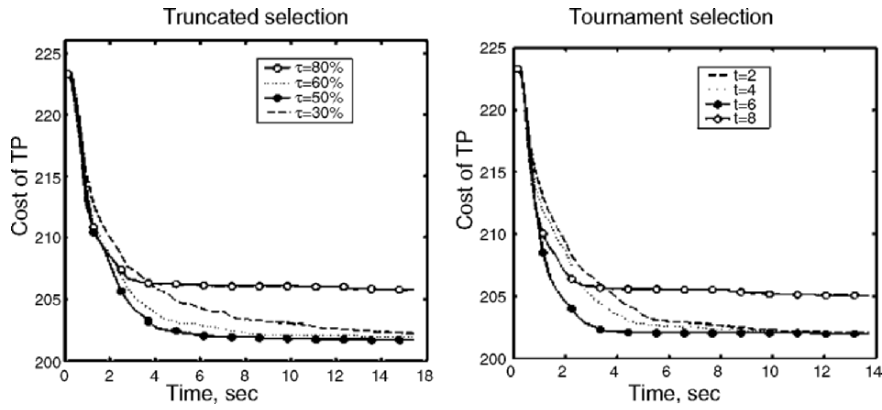


Fig. 7. Average dynamics of the GA with truncation selection and tournament selection (task B\_4\_1, dimension  $n=20$ ).

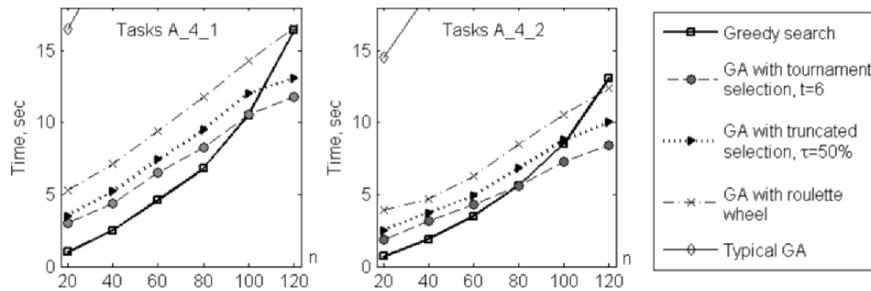


Fig. 8. Average time for solving testing tasks by the greedy method and by various GAs (the GAs run 100 times).

Fig. 8 shows that roulette wheel is out of competition among 3 selections. Timing of truncation selection and timing of tournament selection are comparative, but GA with tournament selection produces solutions near to the optima faster. Fig. 8 also compares the proposed fast GAs, typical GA and greedy method timing on tasks A\_4\_1 and A\_4\_2. This figure shows that the proposed GAs are faster than the typical GA. For large-scale tasks the GAs find out the optimal solutions even quicker the greedy method. Note that the GA with the tournament selection provides the best performance level.

## 9.7 Conclusions

A fast GA of optimization for the multiplicity of checking – retrofit procedures in multidimensional TP is proposed. The acceleration of the GA is achieved through: (a) a smart procedure for the generation of a high-quality initial population, (b) a fast algorithm-calculation of the reliability figures of the whole TP, (c) a specific adaptive fitness function proposed, and (d) appropriate tournament selection. Computational experiments carried out, show that the GA finds out better solutions, and proceeds faster than greedy method for large-scale instances. Proposed schema of GA may be useful for solving hard optimization problems in design of reliable algorithms, checking point allocation in technological processes, diagnosis of defects in complex systems etc.

## Acknowledgement

The authors thank MS Olexiy Kozachko for conducting the computational experiments.

## References

- [1] Garey, M., Johnson, D. (2000) Computers and Intractability. A Guide to the Theory of NP-Completeness, *W.H. Freeman and Company*, New York.
- [2] Gen, M., Cheng, R. (1997) Genetic Algorithms and Engineering Design, *John Wiley & Sons*.
- [3] Mitchell, M. (1996) An Introduction to Genetic Algorithms, *The MIT Press*, Cambridge.
- [4] Rotshtein, A. (1987) Probabilistic-Algorithmic Models of Man-Machine Systems, *Soviet Journal Automation and Information Science*, No 5: 81-86.
- [5] Rotshtein, A., Kuznetsov, P. (1992) Design of Zero-Defect Man-Machine Technologies, Technika, Kiev, [In Russian].
- [6] Rotshtein, A., Shtovba, S. (1999) Optimization of Multidimensional Technological Processes by Genetic Algorithms, *Measuring and Computing Devices in Technological Processes*, No. 2: 7-13, [In Ukrainian].
- [7] Rotshtein, A., Shtovba, S., Dubinenko, S., Kozachko O. (2004) A Heuristic Optimization of Checking Procedure Allocation in Technological Processes for Multidimensional Space of Defects, *Journal of Vinnitsa State Technical University*, No. 1: 54-62, [In Ukrainian].