Parameter meta-optimization of metaheuristics of solving specific NP-hard facility location problem

E S Skakov¹ and V N Malysh²
¹ Novolipetsk Steel, Metallurgov sq. 2, 398040, Lipetsk, Russia
² Department of Humanities and Natural Sciences, Lipetsk Branch of The Russian Presidential Academy of National Economy and Public Administration, Internatsional’naya str. 3, 398050, Lipetsk, Russia
E-mail: wallkirya@mail.ru

Abstract. The aim of the work is to create an evolutionary method for optimizing the values of the control parameters of metaheuristics of solving the NP-hard facility location problem. A system analysis of the tuning process of optimization algorithms parameters is carried out. The problem of finding the parameters of a metaheuristic algorithm is formulated as a meta-optimization problem. Evolutionary metaheuristic has been chosen to perform the task of meta-optimization. Thus, the approach proposed in this work can be called “meta-metaheuristic”. Computational experiment proving the effectiveness of the procedure of tuning the control parameters of metaheuristics has been performed.

1. Introduction
1.1. The problem of placing elements of developing information systems
In this paper the tuning of the parameter values of the metaheuristics of solving the NP-hard facility location problem is carried out in the particular case of NP-hard facility location problem — the problem of placing elements of developing information systems (PPEDIS). PPEDIS is as follows: there is an information system (IS) of some configuration (let’s call it the original IS). There is a need for modernization of the system. There are \(N_{cl}\) customers of information system. Upgraded IS should satisfy the needs of all users by connecting them to the elements (nodes) of information system. There is a predetermined set of potential sites \((N_{ps})\) where IS elements can be installed. There are \(N_{types}\) possible types of IS elements (with different characteristics) available for installation. The task is reduced to minimizing the cost of the transition from the initial state of the IS to the upgraded state with considering a number of restrictions. In [24], [25] and [26] metaheuristic algorithms for solving this problem are described: Evolutionary Algorithm (EA), Bee Colony Algorithm (BCO), Ant Colony Optimization (ACO), Multi-Start Algorithm (MS), Simulated Annealing (SA), Tabu Search (TS). A special case of PPEDIS is solved in this paper: the base stations location problem during planning and optimization of wireless data network.

1.2. Literature overview
The evaluation of the quality of the optimization algorithms is an important issue. Any solution found by some optimization algorithm can be estimated from the point of view of two characteristics:
• the time of finding the solution;
• the value of the objective function of the founded solution.

It is not possible to say exactly which combination of these characteristics is desirable. The ideal case is to find the exact optimal solution in extremely short time interval. However, for NP-hard problems it is impossible. The essence of NP-hardness implies the exponential time of finding the exact solution by the method of exhaustive search. Therefore, finding a solution using metaheuristics involves a compromise between the speed of finding and the accuracy of the obtained solution.

Parameters of metaheuristic algorithm that a decision-maker can specify are called control parameters or free parameters of the algorithm [27]. In [19] the vector (set) of values of free parameters of the algorithm is called the “strategy of the algorithm”.

For each metaheuristic the following statement is true: different sets of control parameters of the algorithm provide solutions of different quality.

We can compare the quality of solutions obtained by the different metaheuristics. Also we can compare the solutions obtained by the same algorithm but with a different set of control parameters values.

By tuning of free parameters values of the optimization algorithm we mean a set of actions and procedures aimed at finding a set of values of free parameters in which the objective function (for the minimization task) reaches the smallest value. Theoretically, it is possible to use an exhaustive search to solve the problem of parameters tuning. However, if we have \( n \) parameters, each of which can take \( k \) values, we will need to perform \( n^k \) experiments. I.e. the task of tuning the free parameters has exponential time complexity.

There are two possible approaches to solving the problem of optimizing the parameters of metaheuristic algorithms:

- offline optimization;
- online optimization.

During offline optimization the values of the various parameters have been fixed before the metaheuristic is executed. The best set of parameter values is selected as the tuning result at the end of the series of experiments. The main disadvantage of offline optimization is high requirements for the time required for the procedure of tuning parameters. The advantage is universality: good method of offline optimization implies its application for several metaheuristic algorithms.

During online optimization the values of the control parameters are changing directly in the process of solving. Most algorithms that implement online optimization use dynamic approach or self-adaptive approach [27], [2]. The obvious lack of optimization directly in the search process is the non-universality of this approach. For example, in our case, it would be necessary to come up with modifications for each of the algorithms of PPEDIS solution. Strategies for self-adaptation of parameters applicable, for example, to evolutionary algorithms, are not suitable for a class of local search algorithms, etc. [27]. A classic example is the dynamic meta-optimization of the probability of a mutation for an evolutionary algorithm. It is based on the idea that in the early steps of the algorithm the probability must be large enough (for the latitude of the search space) gradually decreasing in the process of the algorithm functioning (to obtain a sufficiently accurate solution). Thus in order to carry out the dynamic adaptation correctly it is necessary to have at least an approximate idea of how each parameter should change during the functioning of the algorithm.

Evolutionary metaheuristic is a very young approach to solving optimization problems ([15], [16], [17]). Accordingly the idea of applying an evolutionary approach to parametric identification is even younger. Among the most significant works devoted to the identification of parameters of systems of various complexity we note the works of Demidova and Petrova [6],
Huang and Wang [18], Chang [5]. Studies of such scientists as Cachon and Vazquez [4], Ahmad et al. [1] are directed to the application of a genetic algorithm to optimizing the parameters of artificial neural networks.

The questions of tuning the parameters of metaheuristic algorithms by means of evolution metaheuristics are considered in the works of Gaertner and Clark [11], Wun et al. [31]. Real coding EA is used in both works. Note that in these works the algorithm of parameter adjustment is not given. In these articles the approach is simply indicated in words: the using of a genetic algorithm to optimize the parameters of the ant colony optimization [11] and bee colony algorithm [31]. Karpenko also considers the problem of meta-optimization ([19]) but does not consider the meta-metaheuristic approach as a way of solving this problem.

Thus this paper proposes the new numerical method based on an evolutionary approach which permits to optimize the control parameters for any metaheuristic algorithm aimed at solving PPEDIS.

2. Materials and methods

2.1. System analysis of the process of tuning control parameters for metaheuristic optimization algorithms

The problem of finding the parameter values of some metaheuristic algorithm can be formulated as an optimization problem. Since the metaheuristic algorithms are optimization algorithms, the offline optimization of metaheuristic parameters can be called “meta-optimization”. Moreover, since meta-optimization can be performed using metaheuristics, the approach used in this paper can be called “meta-metaheuristic”.

The scheme of meta-optimization is shown in Figure 1 [27]. Meta-optimization includes two levels: the meta-level and the base level. At the meta-level metaheuristic works with solutions that represent the parameters of the base-level metaheuristics, which need to be optimized. Solution \( x_i \) on the meta-level encodes specific parameter values optimized by the base-level metaheuristics (for example, the length of the tabu list, the probabilistic threshold, etc. in the tabu search). On the meta-level the objective function \( F_{\text{meta}}(X) \) is associated with the best of the solutions found on the base level using metaheuristics with parameters encoded by the solutions \( x_i (i = 1, 2, ..., n_{\text{set}}) \). \( X \) is the set of all possible sets of parameters; \( n_{\text{set}} \) is the total number of different sets of parameters, the power of the \( X \). Metaheuristics on the base level work with solutions that are direct solutions of the initial optimization problem (PPEDIS in our case). The objective function \( F_{\text{base}} \) is used by the base-level metaheuristics to designate the objective function of a particular solution with specific set of parameters \( x_i (i = 1, 2, ..., n_{\text{set}}) \).

For the minimization problem we have the formula

\[
F_{\text{meta}}(X) = \min_{i=1,2,...,n_{\text{set}}} F_{\text{base}}(x_i). 
\]  

The final solution to the meta-optimization task is the set of parameters \( x_{\text{best\_set}} \):

\[
\text{best\_set} = \arg \min_{i=1,2,...,n_{\text{set}}} F_{\text{base}}(x_i). 
\]

2.2. Method of tuning control parameters for metaheuristic optimization algorithms

Further in this paper we will describe the evolutionary method of meta-optimizing the values of the parameters of some metaheuristics of solving the problem of placing elements of developing information systems (PPEDIS).

If the method of meta-optimization is considered as a system, then from the point of view of classification it can be called:

- abstract;
Figure 1. Meta-optimization using meta-metaheuristic approach

Figure 2. The decision-making process for meta-optimization

- mathematical;
- multi-element;
- heterogeneous;
- dynamic.

Figure 2 schematically shows the process of meta-optimization of metaheuristic algorithm. The decision-maker (DM) specifies a set of possible values of the free parameters of the original basic algorithm. The method of meta-optimization takes this set as an input and at the end of the calculations returns to the decision-maker the vector of values with which the base algorithm shows the best result. DM on the basis of the analysis of the meta-optimization results has the opportunity to start the calculations again, having transferred to the input another set of possible values of the control parameters.

The method of meta-optimization is a two-component system (see Figure 3). It includes a base-level algorithm for solving PPEDIS and a unit of evolutionary computations that actually produces meta-optimization. The evolutionary computation block passes the vector of parameter
Let’s pose the problem of meta-optimization as follows. We have PPEDIS of \( N_{cl} \times N_{ps} \times N_{types} \) dimension. We are trying to solve it using some optimization algorithm. The functioning of the algorithm is determined by a set of \( npar \) variable control parameters: \( \chi_1, \chi_2, \ldots, \chi_{npar} \). Each of the parameters is an element of some finite set: \( \chi_1 \in A_1, \chi_2 \in A_2, \ldots, \chi_{npar} \in A_{npar}. \) \( A_1, A_2, \ldots, A_{npar} \) — the sets of possible parameter values from which the selection will occur, they are known in advance. Each set of parameters of a particular algorithm can be represented as a chromosome (see example in Figure 4).

Each “square” (bit) in Figure 4 can take the value 0 or 1. The chromosome consists of genes. Genes are not individual bits, but segments of chromosome (in Figure 4 they are separated by the vertical line). Each gene encodes one of the parameters of the algorithm. Each gene representing a sequence of bits is a binary number. We denote these numbers as follows: \( z_1, z_2, \ldots, z_{npar} \). Because each of the sets \( A_1, A_2, \ldots, A_{npar} \) is finite, we can regard any of these sets as a one-dimensional array (vector). Accordingly the genes \( z_1, z_2, \ldots, z_{npar} \) encode the parameters \( \chi_1, \chi_2, \ldots, \chi_{npar} \) position in arrays \( A_1, A_2, \ldots, A_{npar} \).
Example in Figure 4: \( z_1 = 011_2, z_2 = 1101_2, \ldots, z_{\text{par}} = 00101_2 \). We must translate these numbers into a decimal numeral system: \( z_1 = 3, z_2 = 13, \ldots, z_{\text{par}} = 5 \). This means that the chromosome shown in Figure 4 represents the following: parameter \( \chi_1 \) is the 4th element of array \( A_1 \), \( \chi_2 \) is the 14th element of array \( A_2 \), \( \chi_{\text{par}} \) is the 6th element of array \( A_{\text{par}} \) (elements of arrays will be numbered starting from 1).

The use of binary coding of parameters has the following advantages:

- Evolutionary algorithm with binary coding is much simpler than real-coding algorithm.
- Binary coding permits to combine the tuning of both integer and real parameters. Before starting the tuning for each of the parameters a decision-maker simply has to specify a set of possible values for this parameter.
- Binary coding permits to configure non-numerical (symbolic) parameters of metaheuristic algorithms. By non-numerical parameters we mean the parameters of the algorithm which are not numbers but describe “verbally” some of its properties. For example, there are two types of a bee colony algorithm: with and without global memory. According to the approach proposed in this work, we can code the parameter “a type of BCO from the point of view of memory usage” using a gene consisting of one bit, where 0 will correspond to the use of memory, and 1 — not to use (or vice versa, it is absolutely unimportant).

2.3. Pseudocode

Below there is a general scheme of the method for tuning control parameter values for a metaheuristic algorithm of the PPEDIS solution based on an evolutionary algorithm with binary coding.

(1) Formulate a test problem based on the solution of which the parameters will be tuned.
(2) Introduce the notation: \( n_{\text{run}} \) — the number of starts of the algorithm of finding the PPEDIS solution to determine the fitness of one individual from the proposed method; \( \text{time}_{\text{alg}} \) — time allocated to one launch of the algorithm for finding the PPEDIS solution; \( \text{time}_{\text{overall}} \) — time allocated to the free parameters tuning method.
(3) Initialize the parameters of the evolutionary algorithm of the control parameters tuning (population size \( N_{\text{pop}} \), probability of mutation \( p_{\text{mut}} \), etc.).
(4) Compose the list of control parameters: determine \( \text{par} \), sets \( A_1, A_2, \ldots, A_{\text{par}} \) (sets can be real, integer or non-numeric).
(5) For each integer \( i \) from 1 to \( \text{par} \): the gene encoding \( \chi_i \) must consist of \( w_i \) bits, where \( w_i \) is the minimum of the numbers for which \( |A_i| \leq 2^{w_i} \) is satisfied.
(6) The chromosome that represents an individual consists of \( W \) bits (\( W = \sum_{i=1}^{\text{par}} w_i \)).
(7) Formation of the first generation. Create \( N_{\text{pop}} \) individuals as follows: each individual is a binary vector, each bit is 0 or 1 (generated randomly).
(8) Evaluate the fitness of the individuals of the population: launch the algorithm of solving PPEDIS \( n_{\text{run}} \) times for \( \text{time}_{\text{alg}} \) seconds each run and averaging the objective function of the results.
(9) Select \( N_{\text{child}} \) pairs of parents using tournament selection.
(10) Crossover. At the end of this step we should have \( N_{\text{child}} \) individuals of descendants.
(11) Apply the mutation operator to the children (the result of the mutation in EA with binary coding is the logical inversion of the bit).
(12) Formation of a new generation. It is carried out by selecting the \( N_{\text{pop}} \) of the best individuals from a population consisting of \( N_{\text{pop}} \) parents and \( N_{\text{child}} \) descendants.
(13) If the current running time is less than \( \text{time}_{\text{overall}} \), then go to step 8.

(14) Return the set of parameter values presented by the best individual in the population as a final solution.

The fitness of each individual in the method of parameter optimization is calculated as follows. Each individual consists of a single chromosome that encodes the complete set of control parameter values of some PPEDIS solution algorithm. For each chromosome we have to start the PPEDIS solution algorithm \( n \_\text{run} \) times (\( \text{time}_{\text{alg}} \) seconds per each run). After that we get \( n \_\text{run} \) values of the objective function \( F_{\text{base}} \). Denote them by \( F_{\text{base},i} \), where \( i \in \{ 1, 2, ..., n \_\text{run} \} \). Denote the target function for the individual in the method of finding the control parameters as \( \text{Target}_\text{func} \). It is calculated by the formula:

\[
\text{Target}_\text{func} = \sum_{i=1}^{n \_\text{run}} \frac{F_{\text{base},i}}{n \_\text{run}}.
\]

(3)

The fitness of the individual is inversely proportional to the value of \( \text{Target}_\text{func} \). The \( \text{Target}_\text{func} \) of the best individual in the last generation of the population is equal \( F_{\text{meta}} \).

It should be noted that the method proposed in this paper is most effective in the situations where the number of control parameters (as well as the number of possible values of each of the parameters) is large. It is obvious that when the number of parameters is small, the simplest and the most effective strategy is to look over all possible combinations of control parameters.

We call the evolutionary algorithm with binary coding, presented above, the first phase of the method for optimizing the values of the control parameters of metaheuristic algorithms for solving PPEDIS. In case of availability of additional computer time it is possible to start the second phase of the parameter tuning method based on EA with real coding. During the second phase those genes that are integer and non-numerical will not be changed. There will be further optimization of only real parameters.

2.4. Parameter tuning for meta-algorithm

In this paper a metaheuristic is used to configure the control parameters of other metaheuristics. This fact raises the question of the necessity of tuning the control parameters directly for the evolutionary algorithm for tuning parameters (so-called meta-algorithm). It is necessary to determine the size of the population, the selection method, the probability of mutation (\( p_{\text{mut}} \)), etc. Theoretically the following infinite procedure is possible:

(1) Meta-algorithm is developed for tuning parameters of metaheuristics that solve PPEDIS.

(2) It is necessary to develop meta-meta-algorithm for tuning the control parameters for the evolutionary meta-algorithm.

(3) It is necessary to develop meta-meta-meta-algorithm for tuning the control parameters for the evolutionary meta-meta-algorithm.

... And so on. Obviously, it is necessary to stop optimizing the parameters and hard-fix values of the algorithm parameters at some of the above steps. It is best to do this immediately in the first step. In this paper it is proposed to use an evolutionary meta-algorithm with the following parameters to tune the optimal values of the control parameters of metaheuristics of PPEDIS solution:

- \( N_{\text{pop}} = 10 \).
- \( N_{\text{child}} = 10 \).
- Selection method = “tournament”.

...
• Method of forming a new population = “creation of an intermediate population of $N_{\text{pop}}$ parents and $N_{\text{child}}$ children with the transfer $N_{\text{pop}}$ best individuals to the next generation”.

• Number of crossover points ($n_{\text{points}}$) = 2;

• $p_{\text{mut}} = 0.05$.

2.5. “Default” values of the parameters

Below there are the “default” values of the control parameters of the metaheuristics that were applied to find PPEDIS solution. “Default” parameter values are the values of free parameters, which are recommended in the classic works on metaheuristics.

(1) Simulated Annealing ([20], [21], [30]):
   (a) cooling factor ($\alpha$) = 0.99;
   (b) number of iterations at each temperature ($N_{\text{iter}}$) = 10
   (c) the probability of the worst solution acceptance at the early steps of the algorithm ($p_0$) = 0.9.

(2) Tabu Search ([13], [14], [12]):
   (a) tabu search length ($l$) = 100;
   (b) probabilistic threshold ($p$) = 0.05;
   (c) diversification usage = “yes”.

(3) Multi-Start ([10], [23], [3]):
   (a) maximum number of iterations in a row without improving the objective function ($N_{\text{iter, max}}$) = 100;
   (b) additional local search usage = “no”;
   (c) search intensification parameter ($\theta$) = 1.25.

(4) Evolutionary Algorithm ([15], [16], [17]):
   (a) selection method = “tournament”;
   (b) $N_{\text{pop}} = 100$;
   (c) $N_{\text{child}} = N_{\text{pop}}$;
   (d) $n_{\text{points}} = 1$;
   (e) $p_{\text{mut}} = 0.05$;
   (f) method of forming a new population = “only children are transferred to the new generation”.

(5) Bee Colony Algorithm ([22], [28], [29]):
   (a) selection method = “roulette”;
   (b) number of bees ($B$) = 5;
   (c) number of forward and backward passes ($NC$) = 15;
   (d) memory usage = “yes”.

(6) Ant Colony Optimization ([7], [9], [8]):
   (a) algorithm type = “ACS”;
   (b) number of ants ($A$) = $N_{ps}$;
   (c) first colony relative importance of the pheromone versus the heuristic information ($\alpha_1$) = 3;
   (d) second colony relative importance of the pheromone versus the heuristic information ($\alpha_2$) = 3;
   (e) first colony evaporation rate ($\rho_1$) = 0.1;
   (f) second colony evaporation rate ($\rho_2$) = 0.1;
   (g) first colony pheromone decay coefficient ($\varphi_1$) = 0.1;
   (h) second colony pheromone decay coefficient ($\varphi_2$) = 0.1;
3. Results of the computational experiment

The developed algorithms have been implemented as a software in the programming environment Embarcadero Delphi 2010. The simulation has been performed on a computer with Intel Core i5-3470 processor.

Below are the values of the control parameters of the metaheuristics of the PPEDIS solution ([26],[25] and [24]) found using the method given above.

1. Simulated Annealing:
   (a) $\alpha = 0.97854$;
   (b) $N_{iter} = 60$
   (c) $p_0 = 0.95712$.

2. Tabu Search:
   (a) $l = N_{ps}$;
   (b) $p = 0.15615$;
   (c) diversification usage = “yes”.

3. Multi-Start:
   (a) $N_{iter,max} = 50$;
   (b) additional local search usage = “yes”;
   (c) $\theta = 1.31773$.

4. Evolutionary Algorithm:
   (a) selection method = “tournament”;
   (b) $N_{pop} = 40$
   (c) $N_{child} = N_{pop} \times 2$;
   (d) $n_{points} = 3$;
   (e) $p_{mut} = 0.03345$;
   (f) method of forming a new population = “creation of an intermediate population of $N_{pop}$ parents and $N_{child}$ children with the transfer $N_{pop}$ best individuals to the next generation”.

5. Bee Colony Algorithm:
   (a) selection method = “tournament”;
   (b) $B = 5$
   (c) $NC = 12$;
   (d) memory usage = “yes”.

6. Ant Colony Optimization:
   (a) algorithm type = “ACS”;
   (b) $A = N_{ps}$;
   (c) $\alpha_1 = 3$;
   (d) $\alpha_2 = 1$;
   (e) $\rho_1 = 0.09574$;
   (f) $\rho_2 = 0.11055$;
   (g) $\varphi_1 = 0.12560$;
   (h) $\varphi_2 = 0.10042$;
   (i) $q_{10}^1 = 0.14061$;
   (j) $q_{20}^2 = 0.22816$. 
Table 1. The results of comparing different sets of control parameters for the developed metaheuristics (dimension 100×100×3).

| Parameters | SA    | TS    | MS    | EA    | ACO   | BCO   |
|------------|-------|-------|-------|-------|-------|-------|
| δ          | by method | 5.850 % | 5.402 % | 4.498 % | 5.585 % | 3.496 % | 2.233 % |
|            | default  | 6.467 % | 6.364 % | 4.987 % | 6.558 % | 4.483 % | 3.292 % |
| T          | by method | 9.37 sec | 9.26 sec | 8.96 sec | 9.67 sec | 7.07 sec | 6.26 sec |
|            | default  | 13.74 sec | 10.91 sec | 11.84 sec | 12.77 sec | 11.79 sec | 11.48 sec |
| succ       | by method | 97.55 % | 97.99 % | 98.90 % | 97.82 % | 99.90 % | 99.89 % |
|            | default  | 96.93 % | 97.04 % | 98.41 % | 96.84 % | 98.92 % | 98.83 % |

Table 2. The results of comparing different sets of control parameters for the developed metaheuristics (dimension 500×500×3).

| Parameters | SA    | TS    | MS    | EA    | ACO   | BCO   |
|------------|-------|-------|-------|-------|-------|-------|
| δ          | by method | 5.941 % | 4.234 % | 3.789 % | 5.234 % | 3.234 % | 2.272 % |
|            | default  | 6.412 % | 4.859 % | 4.206 % | 6.260 % | 3.842 % | 2.614 % |
| T          | by method | 48.40 sec | 50.27 sec | 38.82 sec | 51.53 sec | 37.00 sec | 36.49 sec |
|            | default  | 67.72 sec | 60.23 sec | 49.39 sec | 57.92 sec | 49.22 sec | 49.26 sec |
| succ       | by method | 97.46 % | 99.17 % | 99.61 % | 98.17 % | 99.83 % | 99.21 % |
|            | default  | 96.99 % | 98.54 % | 99.19 % | 97.14 % | 99.56 % | 98.87 % |

Further for the algorithms proposed in [26], [25] and [24] the efficiency of a set of “default” parameter values and a set of parameter values that found using the method proposed in this article has been compared.

Three criteria to evaluate the quality of algorithms have been used:

- Estimation of the average deviation (δ) of the solutions from the best known solution (record). The same algorithm running time is fixed for all algorithms.
- Estimation of the average time of algorithm convergence (T) — the average time measured from launching the algorithm to the moment when the record is reached with a specified accuracy.
- Estimation of the proportion of successful runnings (succ) — the ratio of the number of successful starts of the algorithm to the total number of starts. A successful launch means a situation when the algorithm succeeds in finding a solution of a predetermined quality within a fixed time.

Table 1 summarizes the results of a series of experiments that were aimed to evaluate and compare algorithms using three quality criteria for a (100×100×3)-dimension problem. Table 1 contains values of the corresponding quality criteria for the corresponding PPEDIS-solving algorithm with a set of control parameters found using the method proposed in this paper and with a set of the “default” control parameters. For correct comparison in the course of experiments all the algorithms were given the same conditions (running time, required accuracy, etc.).

Table 2 summarizes the results of a series of experiments that were aimed to evaluate and compare algorithms using three quality criteria for a (500×500×3)-dimension problem.
We present Figure 5 to illustrate the work of the proposed evolutionary method. It shows for ACO the change of the value of $\delta$ with each EA generation relative to the static value of $\delta$ obtained by the “default” parameters. Note that $time_{alg} = 10$ sec for data represented in Figure 5.

4. Conclusion

Thus the evolutionary method has been developed for tuning control parameters of metaheuristic algorithms for solving the problem of placing elements of developing information systems. The method implies the possibility of simultaneous selection of real, integer and symbolic parameters.

Solution to the problem of planning and optimization of information systems (for example, wireless data networks) using algorithms with free parameter values found by the method, proposed in this paper, can significantly improve the results compared with the “default” free parameter values. This improvement can reduce the cost of network modernization on average by a few percent which means that it increases the efficiency of planning and building wireless information systems.

Let’s note possible further directions of researches on the topic of this article:

- Extension of the list of metaheuristics for which the developed method is applied. Adding algorithms such as particle swarm optimization, fish school search, bat search, etc.
- Using the concepts of parallel programming to improve the performance of metaheuristics for solving PPEDIS. Study of the effectiveness of the proposed method for tuning the parameters for parallel metaheuristics.
- Extension of the list of control parameters for each metaheuristic algorithm.

References

[1] Ahmad F, Isa N A M, Hussain Z, Osman M K and Sulaiman S N 2015 Pattern Analysis and Applications 18 861

[2] Brest J, Greiner S, Boskovic B, Mernik M and Zumer V 2006 IEEE Transactions on Evolutionary Computation 10 646
[3] Bronmo G, Christiansen M, Fagerholt K and Nygreen B 2007 Computers and Operations Research 34 900
[4] Cachon A and Vazquez R A 2015 Neurocomputing 148 187
[5] Chang W D 2007 Applied Mathematical Modelling 31 541
[6] Demidova L A and Petrova N A 2013 Ryazan State Radio Engineering University Bulletin 45 93
[7] Dorigo M, Birattari M and Stutzle T 2006 IEEE Computational Intelligence Magazine 4 28
[8] Dorigo M and Blum C 2005 Theoretical Computer Science 344 243
[9] Dorigo M and Gambardella L M 1997 IEEE Transactions on Evolutionary Computation 1 53
[10] Dreo J, Petrowski A, Siarry P and Taillard E 2006 Metaheuristics for Hard Optimization: Methods and Case Studies (New York: Springer Science & Business Media)
[11] Gaertner D and Clark K 2005 Proc. Int. Conf. on Artificial Intelligence (Las Vegas, USA) vol 1 p 83
[12] Gendreau M, Hertz A and Laporte G 1994 Management Science 40 1276
[13] Glover F 1989 ORSA Journal on Computing 1 190
[14] Glover F 1990 ORSA Journal on Computing 2 4
[15] Holland J H 1973 SIAM Journal on Computing 2 88
[16] Holland J H 1975 Adaptation in Natural and Artificial Systems (Ann Arbor: University of Michigan Press)
[17] Holland J H and Goldberg D E 1988 Machine Learning 3 95
[18] Huang C L and Wang C J 2006 Expert Systems with Applications 31 231
[19] Karpenko A P 2014 Modern Algorithms of Search Engine Optimization. Algorithms Inspired by Nature (Moscow: MGTU Press)
[20] Kirkpatrick S, Gelatt C D and Vecchi M P 1983 Science 4598 671
[21] Kirkpatrick S 1984 Journal of Statistical Physics 34 975
[22] Lucic P and Teodorovic D 2003 International Journal on Artificial Intelligence Tools 12 375
[23] Marti R, Resende M G C and Ribeiro C C 2013 European Journal of Operational Research 226 1
[24] Skakov E S and Malysh V N 2015 Journal of Information Technology and Applications 5 88
[25] Skakov E S and Malysh V N 2015 Information and Control Systems 76 99
[26] Skakov E S and Malysh V N 2015 Proc. Int. Conf. on Stability and Control Processes 348
[27] Talbi E G 2009 Metaheuristics: From Design to Implementation (New York: Wiley-Interscience)
[28] Teodorovic D, Selmic M and Davidovic T 2015 Yugoslav Journal of Operations Research 25 33
[29] Teodorovic D, Selmic M and Davidovic T 2015 Yugoslav Journal of Operations Research 25 185
[30] Van Laarhoven P J M, Aarts E H L and Lenstra J K 1992 Operations Research 40 113
[31] Wun M H, Wong L P, Khader A T and Tan T P 2014 Proc. 4th World Congress on Information and Communication Technologies (Malacca, Malaysia) vol 1 p 314