www.czasopisma.pan.pl www.journals.pan.pl

BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 67, No. 5, 2019
DOI: 10.24425/bpas.2019.126630

Artificial immune system in planning deliveries in a short time

B. MRÓWCZYŃSKA*, A. KRÓL, and P. CZECH

Silesian University of Technology, Faculty of Transport, 8 Krasińskiego St., Katowice, Poland

Abstract. In the calculations presented in the article, an artificial immune system (AIS) was used to plan the routes of the fleet of delivery vehicles supplying food products to customers waiting for the delivery within a specified, short time, in such a manner so as to avoid delays and minimize the number of delivery vehicles. This type of task is classified as an open vehicle routing problem with time windows (OVRPWT). It comes down to the task of a traveling salesman, which belongs to NP-hard problems. The use of the AIS to solve this problem proved effective. The paper compares the results of AIS with two other varieties of artificial intelligence: genetic algorithms (GA) and simulated annealing (SA). The presented methods are controlled by sets of parameters, which were adjusted using the Taguchi method. Finally, the results were compared, which allowed for the evaluation of all these methods. The results obtained using AIS proved to be the best.

Key words: artificial immune system, genetic algorithm, simulated annealing, open vehicle routing problem, on-time delivery, Taguchi method.

1. Introduction

The presented work deals with the urgent problem related to the delivery of fresh food on time. The specificity of the presented issue is the timely delivery of very small quantities of the product to multiple recipients scattered over a large area of the entire city in a short time interval in such a manner so as to avoid delays and minimize the number of delivery vehicles. This task boils down to the problem of the traveling salesman, which is an NP-hard task, and for a larger number of recipients it cannot be solved with strict methods. It is classified as an open vehicle routing problem with time windows (OVRPWT) because vehicles are not required to return to the depot and a fleet of vehicles have to deliver goods to customers within fixed time intervals. A similar problem in which the maximum time spent in the vehicle by the driver must be minimized is solved in [1] as a variant of the open vehicle routing problem.

Due to the above-mentioned huge complexity of the tasks, the heuristic and metaheuristic approaches are often used. The advantage of such an approach lies in the ability to obtain an almost optimum solution, without the need for fully understanding or even knowing about all internal dependencies characterizing the problem. The only required feature is the existence of an objective function that describes the quality of each solution variant. Just for this reason, artificial intelligence methods are widely used in various fields of science [2], technique [3] and medicine [4]. In [5], the authors present a hybrid evolution strategy for solving the open vehicle routing problem (OVRP). They use GA to solve the transportation network design problem formulated as a bi-level programming model in [6]. In [7], the authors have proposed a procedure for optimizing the vehicle fleet and for scheduling the delivery of different types of food that require different storage temperatures. In [8], the authors used the tabu search (TS) algorithm to solve a vehicle routing problem (VRP). In [9], the authors also used this algorithm for delivery routing of distribution of perishable food. A genetic algorithm (GA) is used to determine the optimum routes for given time windows and researchers showed the effectiveness of their method for small and medium tasks in [10]. A combination of evolutionary methods and simulated annealing (SA) is used to solve the problem of routing with time-windows in [11]. An artificial immune system (AIS) was used to determine the order of receipt of cars from an automatic garage to minimize service time in [12]. In [13], ant-based clustering time series data are presented.

In the article, an artificial immune system is used, namely the algorithm of clonal selection. It is a relatively new method used for optimization. It was described for the first time in article [14]. In article [15], the authors applied AIS to solve the TSP problem. In order to evaluate the efficiency of the AIS algorithm, the same calculations were carried out using methods more widely known and applied for a longer time now, such as the genetic algorithm, where publication [16] is considered to be the first description of this method while simulated annealing is described as an optimization algorithm for the first time in article [17]. Besides, in paper [18], the authors used the same methods to solve the vehicle routing problem with time windows (VRPWT) for delivery of agri-fresh produce to retailers.

The paper is organized in such a way that Section 2 contains the problem description and introduces the numerical model and solution methods, Section 3 describes the calculations and obtained results, with special attention paid to the selection of suitable values of controlling parameters. The same section also includes the considerations on numerical complexity of the selected algorithm while Section 4 contains the final conclusions.

*e-mail: bogna.mrowczynska@polsl.pl

Manuscript submitted 2018-09-04, revised 2018-12-20, initially accepted for publication 2019-01-13, published in October 2019

The main contribution of the paper is the comparison of three well recognized methods of AI to the important problem of delivery planning. The key finding is the undeniable advantage of AIS in this field, but the importance of proper selection of the values of controlling parameters was shown, too. Efficiency of the algorithms was assured by adjusting their parameters using the Taguchi method [19]. The same method allowed to compare all results. Through the calculations, routes were obtained along with lists of delivery vehicles serving customers on these routes.

2. Problem description and numerical model

Stress, sedentary lifestyle and bad eating habits are factors considered to be a serious threat to human health. People who take care of themselves exercise and try to eat healthy food. It is fashionable to use diets arranged by individual orders by dieticians. The problem may be the preparation of diet meals, but in large cities, food companies prepare meals according to individual needs for the whole day. The "box diet" continues to gain popularity.

Since most orders are delivered to the customer in the morning, for large companies a logistics problem can arise here. At this point, a serious issue arises for the service provider of how to deliver the desired goods to a large number of customers waiting for the delivery at various locations in the city, at a specified time and, obviously, at as low the cost as possible. The presented case study makes use of the data published in [20]. Figure 1 shows the graph presenting a simplified structure of the transportation network with the layout of customers and location of the supplier company. There is a driving time assigned to each edge and a service time assigned to each node.

The presented problem is similar to the travelling salesman problem: the shortest (herein: fastest) path which visits the given set of the graph nodes should be found. If all the nodes to be visited are numbered in a range from 1 to n (where n denotes the total number of such nodes) and the starting point is marked as 0, each series of such numbers is a valid schedule of visiting the nodes:

$$\left(k_1, \ldots, k_i, \ldots, k_n\right) \tag{1}$$

where k_i – number of the node which will be serviced as i-th, $k_i \in N$.

In the considered task, there is a number of vehicles sent out to service the nodes. Each of the vehicles (in predetermined order) passes through successive nodes in the order specified in sequence (1). The nodes through which, for example, vehicle v_i moves, are the subsequence of sequence (1), denoted as follows:

$$\left(k_{i_1}, \ldots, k_{i_{n_i}}\right)_{v_i} \tag{2}$$

where k_{i_j} – number of the node which will be serviced by the v_i-th vehicle in the j-th order, $j = 1, \ldots, n_i$, n_i – the number of nodes serviced by the v_i-th vehicle.

During the optimization process, the route of each delivery van is adjusted. The route of one van is regarded as subsequence (2) of sequence (1). All vehicles always start at the same node – the delivery company (node 0). The route ends at the starting

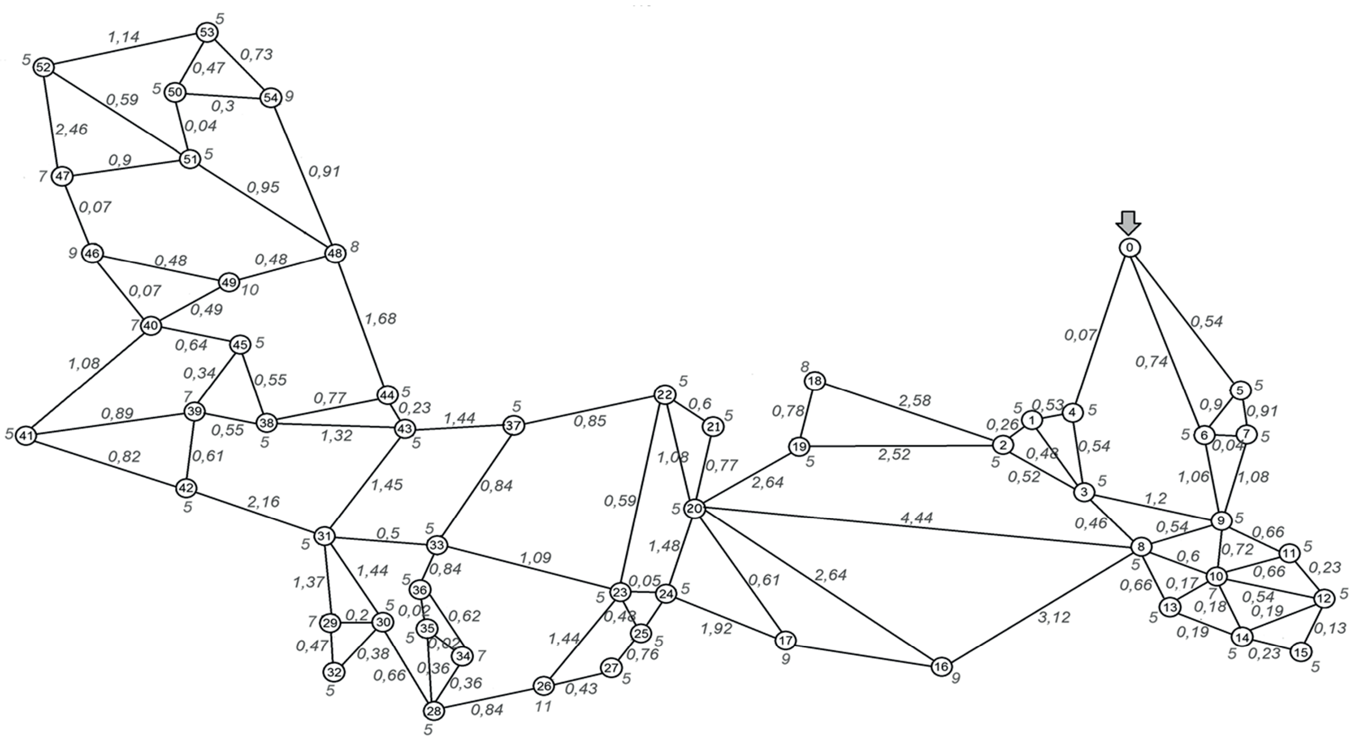

Fig. 1. Graph of the delivery network

node again, but the service time is calculated just to the last serviced point.

The delivery of boxes with food takes place in a narrow time span of 40 minutes.

Working time of the delivery van is defined as the time required for visiting all the planned nodes on the route. It is measured from departure from the starting point to the end of the service of the last node on the route. This time includes also the nodes service time for all planned nodes for each vehicle route. For one vehicle it is described in the following way:

$$t_{wk} = t_{dkij} + t_{skj} \tag{3}$$

where t_{wk} – working time of the k-th delivery van, t_{dkij} – driving time from i to j node, t_{skij} – service time in j node.

Time constraints are included in the AIS algorithm. After servicing the current node, it is checked if the vehicle going to the next node does not exceed the acceptable time. If it does, it does not leave and returns to the base. The next vehicle leaves the base and serves subsequent nodes.

2.1. Solution evaluation. Each solution variant is evaluated, where the assessment takes account of both the duration of the delivery process and the number of vehicles involved.

As the evaluation criterion, the longest of the delivery times (T_{max}) calculated in accordance with formula (3) were adopted. This time is compared with the set time limit (Q) to determine the first rating factor. For this purpose, the following rules were assumed (Equation 4):

1. if T_{max} is less than the limit established, then evaluation (f_T) is equal to 1 (satisfying time),
2. if T_{max} is longer by more than a half of the established limit, the assessment is equal to 0 (time completely unsatisfactory),
3. in other cases, the assessment is linearly dependent on the value of T_{max} (intermediate evaluation).

$$f_T(T_{\max}) = \begin{cases} 1 & T_{\max} \le Q \\ (1.5Q - T_{\max})/0.5Q & Q < T_{\max} < 1.5Q \\ 0 & T_{\max} \ge 1.5Q \end{cases}. \tag{4}$$

The second factor of the assessment (f_p) is associated with the number of vehicles involved (N_P) referred to the expected maximum number of vehicles that are potentially available ($N_{P\max}$):

$$f_P(N_P) = \frac{(N_{P\max} - N_P)}{N_{P\max}}. \tag{5}$$

Final assessment is calculated by multiplying two partial evaluation factors:

$$f = f_T(T_{\max})\, f_P(N_P). \tag{6}$$

In this computation using AIS, the successive customers are being served by a vehicle until the time limit is exceeded. Then the next vehicle serves subsequent customers. Thus the time limit is always met. Under such assumptions, the function determined by expression (4) always takes the value equal to 1.

2.2. Solution methods. As was mentioned in the introductory section, the problem under consideration is NP-hard, therefore artificial intelligence methods were used for solving it, i.e. artificial immune system, genetic algorithm and simulating annealing.

The approach involving artificial intelligence has a fundamental advantage: there is no need to fully understand all inner dependencies of the model or even to know some of them. There is only one demand: the model must allow to calculate the value of the objective function. The objective function is the function defined in the solution domain which determines the solution quality. The objective function does not need to be differentiable and none of its gradients needs to be known.

2.3.1. Artificial immune system. In general, artificial immune systems mimic the immune system of mammals. Interesting attempts at understanding of the immune features in biology were made by [21]. It promptly appeared that the ideas could be applied in science and technology [14]. Commonly, due to the very high complexity of biological immune systems just some of their elements are adopted for optimization. The optimization algorithm used herein is based on the clonal selection paradigm. The primary aim is to recognize the enemy called antigen by the antibodies – the cells of the organism's immune system. For this purpose, the affinity of antibody and antigen cells is rated. The best-matched antibodies neutralize antigens.

Herein the antibody is represented by the sequence described by formula (1). The affinity of antigen-antibody interactions is calculated using formula (4) for every solution. A block diagram of a clonal selection algorithm is presented in Fig. 2.

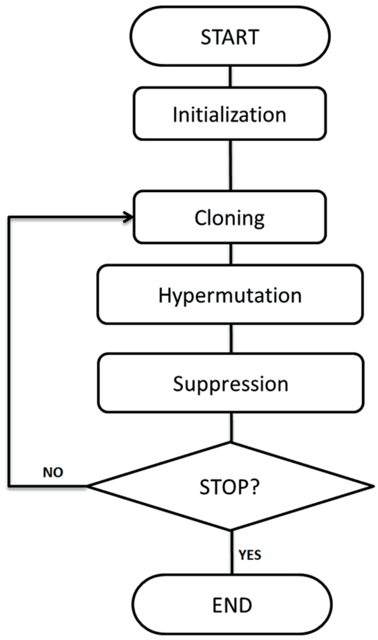

Fig. 2. Clonal selection algorithm

The starting population of antibodies is obtained randomly. The solutions received differ in the order of the elements of sequence (1). The objective function (4) is used as the affinity function.

Clonal selection involves the cloning of cells with a high degree of affinity. Artificial cloning is inspired by cell division. In the algorithm, cloning involves copying solutions. Next, most of these cells will be mutated. Somatic hypermutation is a cellular mechanism by which the immune system adapts to recognize the enemy [22]. In real clonal selection, from a certain moment cloning and mutation of antibodies have a rapid course: the cells are cloning in very large quantities. Then some of them are released into the bloodstream, where they work more efficiently. In artificial hypermutation, a great amount of small changes are introduced in randomly selected solutions. The following mutation operators were used in the algorithm [23]:

- **position based mutation (PBM)** – the randomly chosen element $\boldsymbol{k}$m was put on the randomly chosen position between element $\boldsymbol{k_i}$ and $\boldsymbol{k_j}$:

$$(k_1, \ldots, k_i, k_j, \ldots, \boldsymbol{k_m}, \ldots, k_n) \xrightarrow{PBM} (k_1, \ldots, k_i, \boldsymbol{k_m}, k_j, \ldots, \ldots, k_n), \tag{7}$$

- **order based mutation (OBM)** – the randomly chosen elements $\boldsymbol{k_j}$ and $\boldsymbol{k_m}$ have mutually changed their positions:

$$(k_1, \ldots, \boldsymbol{k_j}, \ldots, \boldsymbol{k_m}, \ldots, k_n) \xrightarrow{PBM} (k_1, \ldots, \boldsymbol{k_{m-1}}, k_j, \ldots, k_n), \tag{8}$$

- **adjacent two-job exchange mutation (ATM)** – the randomly chosen element $\boldsymbol{k_m}$ has changed its position to the place before the previous element $\boldsymbol{k_{m-1}}$

$$(k_1, \ldots, \boldsymbol{k_{m-1}}, \boldsymbol{k_m}, \ldots, k_n) \xrightarrow{ATM} (k_1, \ldots, \boldsymbol{k_m}, \boldsymbol{k_{m-1}}, \ldots, k_n), \tag{9}$$

- drawing a new antibody.

Each mutation operator and drawing the antibody can be used with equal probability.

The advantage of optimization algorithms based on clonal selection lies in their efficiency. The improvement of the solution is based on small changes which are introduced more frequently. The subtlety of changes reduces the probability of omitting the best solutions, and the large number of them results in a wide spectrum of solutions.

2.3.2. Genetic algorithm. Optimization methods using genetic algorithms mimic the process of evolution in the world of living nature and were first described by Holland [16]. Their advantages have allowed for wide application in science and technology [24]. The optimization procedure implementing a genetic algorithm is based on the following assumptions:

- different versions of solutions are individuals, which compete with each other,
- the structure of each individual is determined by the sequence of genes called genotype,
- genotype is subjected to accidental changes – mutations,
- random pairs of individuals can exchange parts of their genotypes – crossover,
- fit function, which is a measure of solution quality (adaptation), determines the probability of transition to the next generation – selective pressure,
- a combination of random mutation and crossover with targeted selection pressure leads towards the optimum solution.

In the presented calculations, the chromosome is represented by sequence (1). The single element of the sequence represents a single gene. The objective function (4) is used as the fit function.

Because of the given structure of an individual, four different mutation operators are designed:

- **swapping the position (SPM)** for a pair of randomly chosen nodes for a randomly selected vehicle:

$$\left(k_{i_1}, k_{i_2}, \ldots, \boldsymbol{k_{i_j}}, \ldots, \boldsymbol{k_{i_m}}, \ldots, k_{i_n}\right)_{v_i} \xrightarrow{SPM} \left(k_{i_1}, k_{i_2}, \ldots, \boldsymbol{k_{i_m}}, \ldots, \boldsymbol{k_{i_j}}, \ldots, k_{i_n}\right)_{v_i}, \tag{10}$$

- **moving the randomly selected subsequence of nodes (MSM)** within a pair of randomly selected vehicles:

$$\begin{aligned}&\Big(\ldots, \left(k_{i_1}, \ldots, \boldsymbol{k_{i_l}}, \boldsymbol{k_{i_{l+1}}}, \ldots, \boldsymbol{k_{i_{l+m}}}, \ldots, k_{i_n}\right)_{v_i}, \ldots,\\ &\left(k_{j_1}, \ldots, k_{j_n}\right)_{v_j}, \ldots\Big) \xrightarrow{MSM} \Big(\ldots, \left(k_{i_1}, \ldots, k_{i_n}\right)_{v_i}, \ldots,\\ &\left(k_{j_1}, \ldots, \boldsymbol{k_{j_l}}, \boldsymbol{k_{j_{l+1}}}, \ldots, \boldsymbol{k_{j_{l+m}}}, \ldots, k_{j_n}\right)_{v_i}, \ldots\Big),\end{aligned} \tag{11}$$

- **changing the order (COM)** of a random nodes subset for a randomly chosen vehicle:

$$\left(k_{i_1}, \boldsymbol{k_{i_j}}, \boldsymbol{k_{i_{j+1}}}, \ldots, \boldsymbol{k_{i_{j+m}}}, \ldots, k_{i_n}\right)_{v_i} \xrightarrow{COM} \left(k_{i_1}, k_{i_2}, \ldots, \boldsymbol{k_{i_{j+m}}}, \ldots, \boldsymbol{k_{i_j}}, \ldots, k_{i_n}\right)_{v_i}, \tag{12}$$

- increasing or decreasing the number of vehicles
  - in the case of decrease, the nodes assigned to the removed vehicle are transferred to a random vehicle,
  - in the case of increase, the randomly selected nodes subset assigned to a randomly selected vehicle is transferred to a new vehicle.

If a mutation is to be done, they are randomly selected with equal probability. The cross-over operator used herein is the OX operator, typical for such permutation genotypes [25]. Generally, for the OX cross-over the parent genotypes are cut at the same randomly selected position, then the children genotypes are created in such a way that the first part remains unchanged and the other is sorted in order of appearance at a partner (Fig. 3).

In practice, the genetic algorithm investigating only a negligible portion of the solutions' space is able to find a solution close to the optimum one. Meanwhile, if the population size

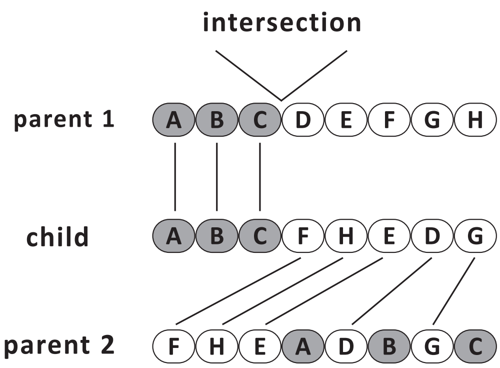

Fig. 3. OX permutation cross-over

and other parameters are properly selected, the probability of the process to remain at a local optimum is very low. Specific for the genetic algorithm, the cross-over operation allows for the creation of a very good solution basing on a pair of average quality solutions, which contain just the promising fragments of the genotype. This determines the generally faster convergence of genetic algorithms in comparison with other methods based on solutions' space exploration.

Selective pressure can be implemented in many ways, but it is always that individuals with a higher value of the fit function are preferred to pass to the next generation. In the present study the method of the roulette wheel was used, in which the probability of passing to the next generation is directly proportional to the value of the fit function. In addition, a certain number of the best individuals pass unconditionally (elite selection). The size of the elite should not be too large to avoid ousting of currently worse individuals from the population, which are candidates to become good solutions.

Additionally, the current leader of each generation is compared with the stored general leader. The better of the two becomes the general leader. This stored solution does not take part in the optimization process, but eventually it contains the result.

2.3.3. Simulated annealing. The principle of simulated annealing is based on an analogy to the physical phenomena occurring during slow cooling and solidification of crystals. The process is characterized by a transition from a high-energy state (hot fluid) into a highly structured state of minimum energy (crystal). At high temperatures the molecule having a plethora of energy can freely jump to any position. Then, as the temperature decreases, transitions towards the states of lower energy are preferred [26].

Simulated annealing algorithm is a modification of simple iterative methods, which are based on the replacement of the current solutions with a randomly generated neighboring solution (in solution space), when this leads to the increase of the objective function value. Simulated annealing also permits for replacement with a worse solution under certain conditions. This modification increases the ability to avoid getting stuck in a local optimum and allows for continuing the search for the global optimum. The probability of replacement with a worse solution is not constant and is slowly decreasing during the optimization process. The probability (p) is determined by a quantity, called, by analogy, the temperature:

$$p = e^{-\frac{f_x - f_0}{T}} \tag{13}$$

where f_0 denotes an objective function of the current solution, f_x denotes an objective function of the modified solution and T denotes temperature. High temperature at the beginning allows for intensive exploitation of the solutions' space, and then it is decreasing with time, so that finally the probability of selecting a worse solution is negligible. Generally, geometric temperature decrease in subsequent iterations is assumed:

$$T_{i+1} = qT_i \qquad q < 1. \tag{14}$$

The q factor may be determined by an empirical relationship, wherein n is the number of iterations:

$$q = 1 - \frac{5}{n}. \tag{15}$$

Simulated annealing can be an effective way to optimize for those problems where the solution space has a low "effective diameter" in all dimensions. This means that the used operator generating the neighboring solution should be able to cross the whole domain of each degree of freedom of the problem in a relatively small number of steps. The number of degrees of freedom (the dimensions of solution space) can however be large. The terms "small" and "large" are of relative importance and their specific values depend on the optimized problem.

Three modifications to the classical simulated annealing algorithm are introduced, which significantly improves its efficiency:

- the algorithm works in parallel mode, which means that the predetermined number of individual processes run simultaneously,
- additionally, the best solution obtained in all processes in the hitherto steps is stored, and this stored solution is not taking part in the optimization, but eventually it contains the result,
- these parallel processes are independent, but sporadic information exchange among them is allowed (similar to cross-over operator in the case of genetic algorithm – parts of the structure between a pair of solutions are exchanged).

The introduction of the parallel mode requires modification of the factor determining the rate of temperature decrease, r_p used herein denotes the number of the parallel processes:

$$q = 1 - \frac{5r_p}{n}. \tag{16}$$

The intensity of information exchange is determined by the cross-over probability. This value should be very low in order to keep the individual processes independent. In each step of the optimization, the current solutions in each of the parallel processes can be crossed-over with a current solution from another randomly selected process.

The simulated annealing requires operators changing the structure of the solution: generation of the neighboring solution. It is "mutation" in the nomenclature for genetic algorithms, and operators described in 2.3.2 were used herein.

2.4. Significant differences in the algorithms used. In the task being considered, there is a number of vehicles sent to service the nodes (customers are located at graph nodes). For GA and SA, the initial solution is generated in the same way: first the number of vehicles in service is randomly drawn, then the nodes to be serviced are sequentially assigned to the vehicles. Thus, the number of nodes assigned to a vehicle is approximately the same. The methods applied differ slightly regarding their inner structure. Genetic algorithm and simulated annealing try to minimize the number of vehicles required and the time needed for service.

For the artificial immune system the method of solution generating assures keeping the time limit, and the number of required vehicles is minimized. In the presented implementation of AIS, successive customers are being assigned to a vehicle until the time limit is exceeded, then a next vehicle is taken into account. Under such assumptions, the function determined by expression (2) always takes the value of 1, and the final assessment (4) is just expression (3). So the task comes down to minimization of the number of required vehicles (the time limit is always met).

3. Calculation

All the calculations were performed on a computer with an Intel® Core™ processor i7–3630QM CPU @ 2.40GHz. The calculations were made using implementations of own programs written in C++.

3.1. Selection of optimum values of parameters. The operation of the optimization methods used is controlled by a number of parameters. These parameters are collected in Table 1. Their values should be adjusted in such a way so as to ensure as good quality of obtained results and as high repeatability as possible. The last feature is very important because all the methods are non-deterministic.

For this purpose, the operation of all optimization methods for different values of these parameters should be examined. It was assumed that each parameter will accept two values in the typical range. Despite this limitation of the level number for each of controlling parameters, the number of possible parameter sets, and thus the number of test series to perform, is still very high. It is $2^5 = 32$ for the genetic algorithm. For simulated annealing and artificial immune system, it would be significantly less because only $2^3 = 8$ test series are necessary. Thus the Taguchi method of experiment planning was used to diminish the number of test series to perform [27].

Taguchi assumed that each process is influenced by two groups of factors: controlled parameters, which can be used to the process management, and uncontrolled, random disturbing factors. The main idea behind Taguchi's approach is to find such

Table 1
Parameters controlling the optimization

AIS

Population size	POP
Mutation variant	VM
Cloning variant	VC

GA

Population size	POP
Cross-over probability	PX
Mutation probability	PM
Number of initial mutations	MUT0
Elite size	EL

SA

Number of processes	POP
Cross-over probability	PX
Number of initial mutations	MUT0

values of the control parameters which allow for minimizing the impact of confounding factors and for achieving the highest result quality.

When considering some statistical relationships, Taguchi demonstrated that there is no need to carry out the tests for all potential sets of parameters. He suggested the use of orthogonal arrays containing representative sets of parameters, allowing for formulation of general conclusions by means of some tools of statistical analysis.

In the first step, after determining the number of parameters and the required number of levels values the relevant orthogonal array is chosen (Table 2).

Table 2
Fragment of the selector of orthogonal tables

No. of levels	No. of parameters							
	2	3	4	5	6	7	8	9
2	L4	$\boxed{\text{L4}}$	L8	$\boxed{\text{L8}}$	L8	L8	L12	L12
3	L9	L9	L9	L18	L18	L18	L18	L27
4	L'16	L'16	L'16	L'16	L'32	L'32	L'32	L'32

The test series must therefore be planned using orthogonal arrays L8 (GA) and L4 (SA and AIS). Orthogonal array L8 describes a series of 8 tests by supplying adequate levels of parameter values for each series. As it can be seen, instead of the 32 series of tests it is sufficient to perform just 8 of them. Meanwhile, array L4 allows to reduce the required number of series from 8 to 4. Required parameter values for all series of tests are summarized in Table 3.

To obtain reliable results, single series should contain at least a few tests. Basing on an analysis of similar cases [28], it was assumed that each series will consist of eight runs of optimization procedure. The value of the objective function after

Table 3
Selected parameter values

Series	L4 (AIS)				
	POP	VM	VC		
1	30	1	PROP		
2	20	2	PROP		
3	30	2	MAX		
4	20	1	MAX		
Series	L8 (GA)				
	POP	PX	PM	MUT0	EL
1	100	0.2	0.2	5	1
2	100	0.2	0.2	20	5
3	100	0.8	0.5	5	1
4	100	0.8	0.5	20	5
5	200	0.2	0.5	5	5
6	200	0.2	0.5	20	1
7	200	0.8	0.2	5	5
8	200	0.8	0.2	20	1
Series	L4 (SA)				
	POP	PX	MUT0		
1	10	0.01	5		
2	10	0.05	20		
3	20	0.01	20		
4	20	0.05	5		

Table 4
Values of the fit function for test series

Series	AIS							
	1	2	3	4	5	6	7	8
1	0.50	0.50	0.53	0.50	0.50	0.50	0.53	0.50
2	0.50	0.50	0.50	0.50	0.53	0.50	0.50	0.50
3	0.50	0.50	0.50	0.53	0.50	0.50	0.50	0.53
4	0.47	0.47	0.47	0.47	0.47	0.47	0.47	0.47
Series	GA							
	1	2	3	4	5	6	7	8
1	0.53	0.40	0.43	0.40	0.53	0.50	0.53	0.40
2	0.40	0.40	0.40	0.40	0.43	0.43	0.50	0.43
3	0.51	0.47	0.40	0.53	0.50	0.53	0.40	0.53
4	0.47	0.40	0.43	0.43	0.53	0.47	0.40	0.53
5	0.53	0.47	0.50	0.50	0.51	0.57	0.53	0.53
6	0.47	0.44	0.52	0.45	0.46	0.43	0.53	0.50
7	0.40	0.53	0.40	0.53	0.53	0.53	0.53	0.47
8	0.43	0.40	0.50	0.40	0.53	0.43	0.42	0.53
Series	SA							
	1	2	3	4	5	6	7	8
1	0.46	0.31	0.38	0.40	0.31	0.40	0.32	0.21
2	0.22	0.16	0.23	0.47	0.30	0.35	0.23	0.15
3	0.45	0.40	0.47	0.48	0.37	0.45	0.50	0.47
4	0.43	0.50	0.50	0.51	0.47	0.50	0.47	0.47

200 steps for GA and AIS and 2000 steps for SA was regarded as a reliable result (at such numbers of steps, the fit function stopped to change). All results are summarized in Table 4.

The key concept of the Taguchi method is taken from the signal theory: it is the so-called signal to noise ratio (S/N). Signal corresponds to the desired value while noise represents the undesirable random distortions. For a quantity which is to be maximized, the value of S/N for i-th series of tests is given by [29]:

$$SN_i = -10\log_{10}\left(\frac{1}{n}\sum_{j=1}^{n} - \frac{1}{X_{ij}^2}\right) \tag{17}$$

where: n – series size, X_{ij} – result of j-th test of i-th series.

Table 5 summarized the calculated values of the signal to noise ratio for all series of tests. This table also includes the levels of various parameters in each series, as in orthogonal arrays L4 and L8. The calculated values seem not to differ too much, but it should be remembered that the logarithmic scale is used here. The highest value of the signal to noise ratio SN_i indicates the set of parameters that is most favorable for the operation of the optimization procedure. The next step is to test the impact of each parameter on the quality of the solution. For this purpose, the average value of the signal to noise ratio SN_{mn} for each value level (m) for each parameter (n) is calculated.

This is illustrated by highlighting relevant cells of Table 5 as an example of SN_{52} for GA:

$$SN_{52} = \frac{SN_2 + SN_4 + SN_5 + SN_7}{4}. \tag{18}$$

For each parameter, the maximum and minimum values of averaged signal to noise ratio and the difference (Δ) have to be calculated. The value of this difference describes the relative influence of the parameter on the result quality. The results of this phase of the analysis are summarized in Table 6. As it may be noted, the greatest impact for GA is exerted by the number of initial mutations, the population size and the probability of mutations.

In the case of SA, the population size is most important. In the case of AIS, the mutation variant and cloning variant are most important, although their dominance over the population size is not significant. Other parameters are of less importance.

The Taguchi test result for AIS indicates that the number of clones should be proportional to the value of the affinity function, which is consistent with the literature [15]. The obtained value of the second of the tested parameters confirmed the

Table 5
Values of signal to noise ratio for test series

Series	AIS					
	POP	VM	VC			
1	2	1	1			–5.8871
2	1	2	1			–5.9544
3	2	2	2			–5.8871
4	1	1	2			–6.6199
Series	GA					
	POP	PX	PM	MUT0	EL	SN_i
1	1	1	1	1	1	–6.8376
2	1	1	1	2	2	–7.4989
3	1	2	2	1	1	–6.4711
4	1	2	2	2	2	–6.9137
5	2	1	2	1	2	–5.7567
6	2	1	2	2	1	–6.5184
7	2	2	1	1	2	–6.3773
8	2	2	1	2	1	–6.9922
Series	SA					
	POP	PX	MUT0			
1	1	1	1			–9.8974
2	1	2	2			–13.0897
3	2	1	2			–7.0974
4	2	2	1			–6.4010

Table 6
Analyses of parameters impact on the solution quality

SN_{nm}	AIS				
	POP	VM	VC		
1	–5.9544	–6.2535	–5.9207		
2	–6.1313	–5.9207	–6.2535		
min	–6.1313	–6.2535	–6.2535		
max	–5.9544	–5.9207	–5.9207		
Δ	0.1769	0.3328	0.3328		
SN_{nm}	GA				
	POP	PX	PM	MUT0	EL
1	–6.9303	–6.6529	–6.9265	–6.3607	–6.7048
2	–6.4112	–6.6886	–6.4150	–6.9808	–6.6366
min	–6.9303	–6.6886	–6.9265	–6.9808	–6.7048
max	–6.4112	–6.6529	–6.4150	–6.3607	–6.6366
Δ	0.5192	0.0357	0.5116	0.6201	0.0682
SN_{nm}	SA				
	POP	PX	MUT0		
1	–11.4936	–8.4974	–8.1492		
2	–6.7492	–9.7453	–10.0935		
min	–11.4936	–9.7453	–10.0935		
max	–6.7492	–8.4974	–8.1492		
Δ	4.7444	1.2480	1.9443		

effectiveness of the mutation model, in which three mutation operators described by formulas (7–9) and the generator of a new antibody with the same probability are used.

These relationships are shown graphically in Fig. 4 and allow for further conclusions on the values of parameters, which actually have a significant impact on the efficiency of the optimization procedures. This analysis allows for selection of the best sets of parameters for all optimization procedures, which are collected in Table 7.

Table 7
Optimum parameter sets

AIS		GA		SA	
POP	20	POP	200	POP	20
VM	1	PX	0.2	PX	0.1
VC	PROP	PM	0.5	MUT0	5
		MUT0	5		
		EL	5		

3.2. Comparison of optimization results. All the optimization procedures were repeatedly run for the above-shown best parameters sets, then the results were compared. These data are summarized in Table 8.

The results proved that the artificial immune system gave the best effects. It reached the best possible value of the fit

Table 8
Results for multiple runs of AIS, GA and SA

Test No.	AIS	GA	SA
1	0.6	0.533	0.515
2	0.6	0.533	0.492
3	0.6	0.538	0.467
4	0.6	0.500	0.499
5	0.6	0.524	0.467
6	0.6	0.533	0.467
7	0.6	0.533	0.433
8	0.6	0.500	0.413
9	0.6	0.533	0.467
10	0.6	0.533	0.467
f_{avg}	0.6	0.526	0.469
σ	0	0.014	0.030

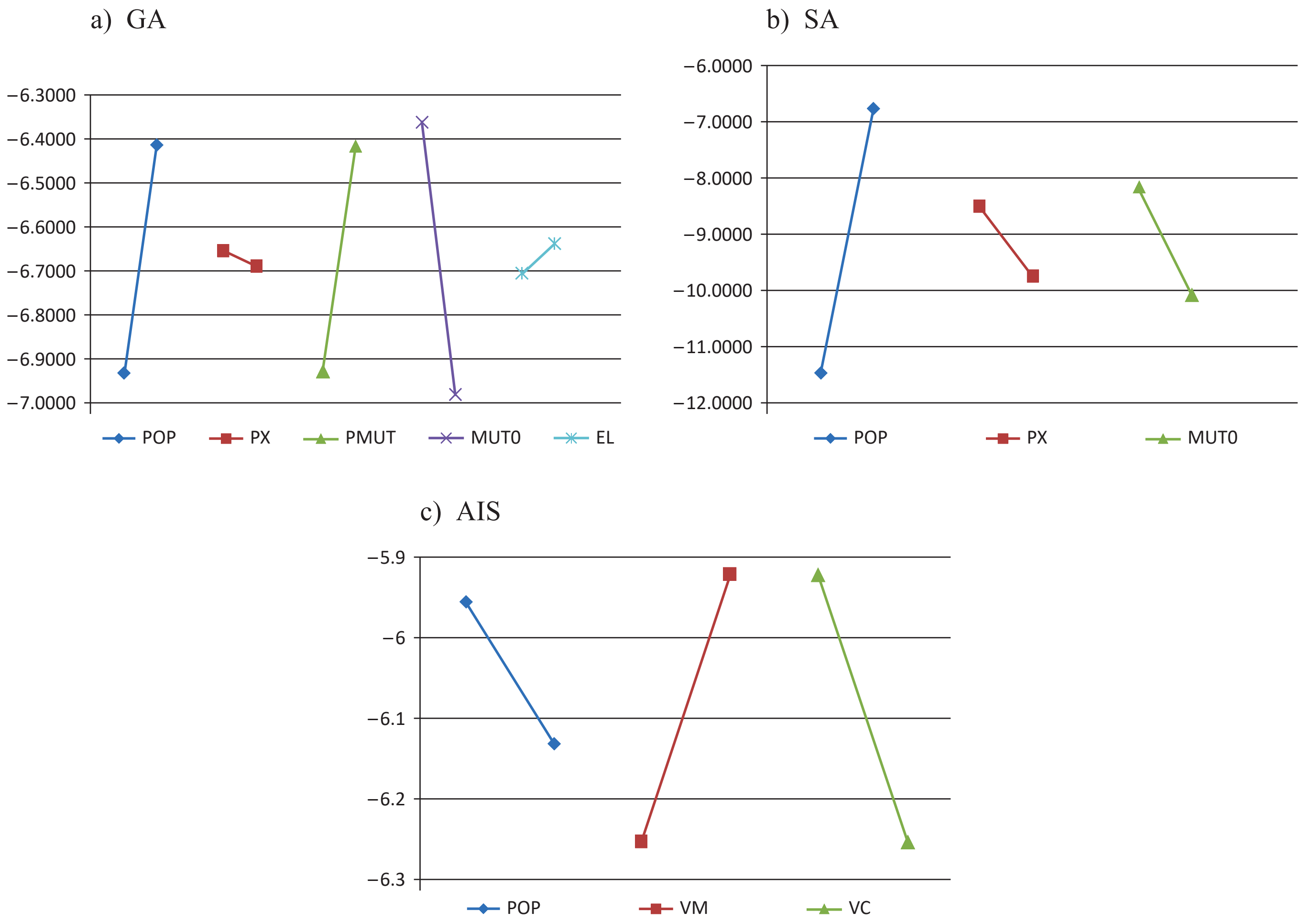

Fig. 4. Relative impact of parameters on optimization efficiency

function in every run. In turn, the genetic algorithm provides much better results than simulated annealing (the average value of the fit function was higher by 12%), and its results are much more consistent (standard deviation was twice smaller). Figure 5 shows the convergence of two best solutions for each examined method. For AIS and GA values of function (6) are shown for each generation from 0 to 200. To make the comparison possible, every tenth result for SA is shown in the range of 0 to 2000.

As it can be noted, AIS immediately reaches high values and in the interval of 50th to 200th generation does not change them. The final result was achieved in the later generations. Genetic algorithm quickly comes to the final solution, thus increasing the number of generations will not significantly affect the quality of the solution. In the case of simulated annealing, getting closer to the final solution requires a much larger number of steps, and in a few cases it was done just at the end of the optimization process. Therefore, for SA another series of tests for the two-fold increased number of steps was carried out. As a result, greater reproducibility of results was obtained, but the standard deviation was still higher than for GA.

The average value of the objective function in this series has not increased significantly. It is therefore concluded that

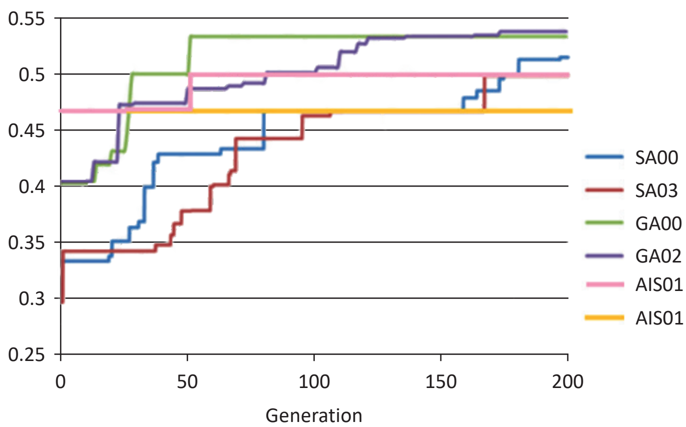

Fig. 5. Comparison of optimization convergence for all methods

SA is not an appropriate optimization procedure for the examined issue.

As was mentioned above, both AIS and GA ensure high reproducibility of the obtained value of the fit function (it concerns GA not so clearly), but it does not mean an identity of the

Table 9
Best solutions for AIS and GA

AIS

Value of affinity function	0.6
Number of vehicles	12
Delivery duration	39’35”
Vehicle	Nodes
1	7 25 24 28 34
2	16 23 33 35 36
3	44 43 38 45 41
4	6 12 11 15 14 13 9
5	410 2 815
6	53 51 50 54
7	21 37 20 18
8	17 30 32 29
9	27 31 39 52
10	42 40 46
11	3 19 26 22
12	48 47 49

GA

Value of fit function	0.533
Number of vehicles	14
Delivery duration	39’ 46”
Vehicle	Nodes
1	1 2 49 47
2	7 6 8 5
3	10 9 11 12
4	14 13 15 16
5	22 23 24 20 19
6	21 17 18
7	28 27 26 25
8	31 29 32 30
9	34 33 35 36
10	40 38 39 37
11	41 42 43 44
12	46 48 45
13	3 4 50 51 52
14	53 54

Table 10
Results of calculations using AIS for graphs of various sizes

graph		average time to obtain the best solution	standard deviation	variant	time to get the best result [s]	the shortest route [km]	value of the affinity function	number of changes for the better	minimum number of vehicles
number of vertices	number of edges								
55	97	11.2	8.26	1	25	199500	50.125313	186	13
				2	9	186600	53.590568	189	12
				3	5	197720	50.576573	194	13
				4	12	193260	51.743765	165	12
				5	5	196500	50.890585	178	13
110	198	128	172.52	1	58	464070	21.548473	390	28
				2	37	450670	22.189185	433	27
				3	435	430290	23.24014	456	25
				4	33	469920	21.280218	405	28
				5	77	435390	22.967914	456	26
165	299	519.8	264.83	1	847	746950	13.387777	768	44
				2	643	721120	13.867318	794	40
				3	233	753890	13.264535	777	43
				4	261	725290	13.787588	754	41
				5	615	720350	13.882141	695	39
220	400	1499.4	1103.00	1	853	1458130	6.858099	1157	58
				2	1076	1451590	6.888998	1254	59
				3	3325	1472830	6.78965	1163	59
				4	551	1491460	6.70484	1048	59
				5	1692	1496210	6.683554	1150	60
275	531	4293.2	2267.81	1	3775	1553170	6.438445	1557	73
				2	7876	1583040	6.31696	1497	75
				3	2174	1597300	6.260565	1397	75
				4	2696	1594150	6.272935	1423	74
				5	4945	1584710	6.310303	1330	74

generated solutions. Table 9 shows the best solutions for both methods, the assumed time limit was 40 minutes. For the AIS method, the value of affinity function reached 0.6 and only 12 vehicles were required.

3.3. Numerical efficiency of AIS. Numerical efficiency was also tested for AIS. For this purpose, at least 5 tests were performed for each of 5 different size graphs with vertices from 55 to 275. The results of the calculations are presented in Table 10.

Figure 6 shows the dependence of the average time of obtaining the best solution as a function of the number of vertices. The calculation time is a function of a higher order. The trend has been expressed by function (19):

$$y = 7.34 \cdot 10^{-7} x^4. \tag{19}$$

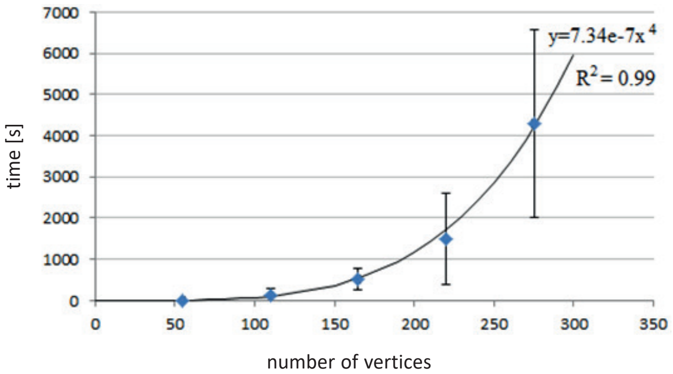

Fig. 6. Average time required to obtain the best solution as a function of number of vertices

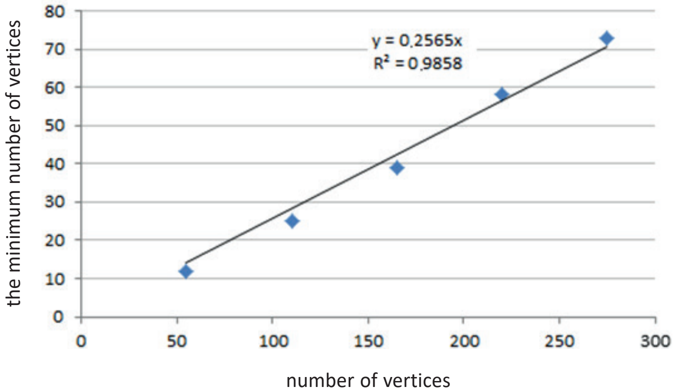

Fig. 7. Minimum number of vehicles which serve customers as a function of the number of vertices

Figure 7 shows the dependence of the minimum number of vehicles which serve customers as a function of the number of vertices. Its trend is linear.

The polynomial computational complexity of the order of four may seem to be a drawback of the presented approach. However, the factor $7.34 \cdot 10^{-7}$ guarantees the acceptable calculation time for reasonable data.

4. Conclusions

In the article, the task classified as OVRPWT was solved. Such problem comes down to the task of a traveling salesman, which belongs to NP-hard problems.

The case study: the scheduling of routes of a fleet of delivery vehicles supplying food products to customers waiting for delivery within a specified, short time, in such a manner so as as to avoid delays and minimize the number of delivery vehicles, was determined.

Scheduling deliveries in a large catering company may require the use of quick methods for determining the routes of vehicles, which results in the implementation of artificial intelligence algorithms for solving the problem.

Herein, the artificial immune system was compared with two other methods: genetic algorithms and simulated annealing.

- The high efficiency of the calculations was assured by application of the Taguchi method of experimental design to determine the optimum values of control parameters. The main calculations were carried out using such adjusted parameters.
- The best results were obtained for the artificial immune system, which appeared to be the most effective. Genetic algorithm achieved the value of the fit function that was worse by 12%, and simulated annealing was worse by as much as 23%.
- The advantage of the artificial immune system was also expressed in the dispersion of the fit function values – although it generated different solutions, their quality was always the same. Also here genetic algorithm is rated worse, and again the worst is simulated annealing.
- The best results of AIS confirm the legitimacy of developing the presented method towards large, dynamic VRP tasks. This method can also be used for planning routes of electric cars, and the location of the recharging points of vehicles [29, 30]. The introduction of electric delivery vehicles, and in the future, autonomous vehicles, will bring ecological benefits to the centers of big cities [31].

Acknowledgements. Publication supported as part of the post-doctoral habilitation grant, Silesian University of Technology, 12/030 / RGH18 / 0024.

References

[1] P.P. Repoussis and C.D. Tarantilis, "Solving the fleet size and mix vehicle routing problem with time windows via adaptive memory programming", *Transportation Research Part C: Emerging Technologies* 18(5) 695–71 (2010).

[2] A.D. López-Sánchez, A.G. Hernández-Díaz, D. Vigo, R. Caballero, and J. Molina, "A multi-start algorithm for a balanced real-world open vehicle routing problem", *European Journal of Operational Research* 238(1) 104–113 (2014).

[3] R. Tadeusiewicz, L. Ogiela, and M.R. Ogiela, "The automatic understanding approach to systems analysis and design", *International Journal of Information Management* 28(1) 38–48 (2008).

[4] J. Ferdyn-Grygierek and K. Grygierek, "Multi-variable optimization of building thermal design using genetic algorithms", *Energies* 10(10) 1570 (2017).

[5] K. Pancerz, A. Lewicki, and R. Tadeusiewicz, "Ant-based extraction of rules in simple decision systems over ontological graphs", *International Journal of Applied Mathematics and Computer Science* 25(2) 377–387 (2015).

[6] S. Dinu and G. Bordea, "A new genetic approach for transport network design and optimization", *Bull. Pol. Ac.: Tech.* 59(3) 263–272 (2011).

[7] C.-I. Hsu and W.-T. Chen, "Optimizing fleet size and delivery scheduling for multi – temperature food distribution", *Applied Mathematical Modelling* 38(3) 1077–1091 (2014).

[8] R.T. Berger, C.R. Coullard, and M.S. Daskin, "Location-routing problems with distance constraints", *Transportation Science* 41(1) 29–43 (2007).

[9] A. Osvald and L.Z. Stirn, "A vehicle routing algorithm for the distribution of fresh vegetables and similar perishable food", *Journal of FoodEngineering* 85(2) 285–295 (2008).
[10] N.H. Moin, S. Salhi, and N.A.B. Aziz, "An efficient hybrid genetic algorithm for the multi-product multi-period inventory routing problem", *International Journal of Production Economics* 133(1) 334–343 (2011).
[11] R. Baños, J. Ortega, C. Gil, A.L. Márquez, and F. de Toro, "A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows", *Computers & Industrial Engineering* 65(2) 286–296 (2013).
[12] C. Pypno and G. Sierpiński, G. Automated large capacity multi-story garage concept and modeling of client service processes, *Automation in Construction* 81: 422–433 (2017).
[13] K. Pancerz, A. Lewicki, R. Tadeusiewicz, and J. Warchoł, "Ant-based clustering in delta episode information systems based on temporal rough set flow graphs", *Fundamenta Informaticae* 1–2, 143–158 (2013).
[14] L.N. De Castro and J. Timmis, *Artificial immune systems: a new computational intelligence approach*. London; New York: Springer (2002).
[15] L.N. De Castro and F.J. Von Zuben, The Clonal Selection Algorithm with Engineering Applications, In GECCO 2002 – Workshop Proceedings, Morgan Kaufmann, 36–37.
[16] J.H. Holland, "Adaptation in natural and artificial systems", The University of Michigan Press, Ann Arbor, MI, 1975.
[17] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, "Equation of state calculations by fast computing machines", *The Journal of Chemical Physics* 21(6) 1087–1092 (1953).
[18] M. Shukla and S.J. Harkharia, Artificial Immune System-based algorithm for vehicle routing problem with time window constraint for the delivery of agri-fresh produce, *Journal of Decision Systems* 22(3) 224–247 (2013).
[19] A. Azadeh, S. Elahi, M.H. Farahani, and B.Nasirian, "A genetic algorithm Taguchi based approach to inventory routing problem of a single perishable product with transshipment", *Computers & Industrial Engineering* 104, 124–133 (2017).
[20] B. Mrówczyńska and M. Cieśla, "Planning routes of vans in a catering company", *ICLEEE 2017 International Conference of Logistic*, *Economics and Environmental Engineering. Maribor*, *Slovenia*, 66–70 (2017).
[21] A.S. Perelson, Applications of optimal control theory to immunology. Recent Developments in Variable Structure Systems, Economics and Biology. Springer Verlag, New-York: 272–287D (1978).
[22] C. Janeway, *Immunobiology: the immune system in health and disease*, 6th ed. New York: Garland Science (2005).
[23] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Springer – Verlag Berlin Heidelberg (1996).
[24] E. Goldberg, *Genetic Algorithms in Search*, *Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc. (1989).
[25] L. Davis, "Applying adaptive algorithms to epistatic domains", in *Proceedings of the 9th International Joint Conference on Artificial Intelligence – Volume 1*, San Francisco, CA, USA, 162–164 (1985).
[26] V. Černý, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm", *Journal of Optimization Theory and Applications* 45(1) 41–51 (1985).
[27] R.K. Roy, *A primer on the Taguchi method*. Dearborn, Mich: Society of Manufacturing Engineers (1990).
[28] M. Hajiaghaei-Keshteli, "The allocation of customers to potential distribution centers in supply chain networks: GA and AIA approaches", *Applied Soft Computing*, 11(2) 2069–2078 (2011).
[29] M. Schiffer and G. Walther, "Strategic planning of electric logistics fleet networks: A robust location-routing approach", *Omega* 80, 31–42 (2018).
[30] A. Montoya, C. Guéret, J.E. Mendoza, and J. G. Villegas, "The electric vehicle routing problem with nonlinear charging function", *Transportation Research Part B: Methodological* 103, 87–110 (2017).
[31] V. Leggieri and M. Haouari, "A practical solution approach for the green vehicle routing problem", *Transportation Research Part E: Logistics and Transportation Review* 104, 97–112 (2017.