ISSN: 1231-4005
e-ISSN: 2354-0133
DOI: 10.5604/01.3001.0012.4785

THE APPLICATION OF GENETIC ALGORITHM IN THE ASSIGNMENT PROBLEMS IN THE TRANSPORTATION COMPANY

Mariusz Izdebski, Marianna Jacyna

Warsaw University of Technology
Faculty of Transport, Department of Logistics and Transport Systems
Koszykowa Street 75, 00-662 Warsaw, Poland
tel.: +48 22 2341413, +48 48 22 235855, fax: +48 22 2347582, +48 22 2347582
e-mail: mizdeb@wt.pw.edu.pl, maja@wt.pw.edu.pl

Abstract

The article presents the problem of the task assignment of the vehicles for the transportation company, which deals with the transport of the cargo in the full truckload system. The presented problem is a complex decision making issue which has not been analysed in the literature before. There must be passed through two stages in order to solve the task assignment problem of the vehicles for the transportation company. The first stage is to designate the tasks, the other one is to determine the number of the vehicles that perform these tasks. The task in the analysed problem is defined as transporting the cargo from the suppliers to the recipients. The transportation routes of the cargo must be determined. In order to solve the task assignment problem of the vehicles, the genetic algorithm has been developed. The construction stages of this algorithm are presented. The algorithm has been developed to solve the multi-criteria decision problem. What is more, the algorithm is verified by the use of the real input data.

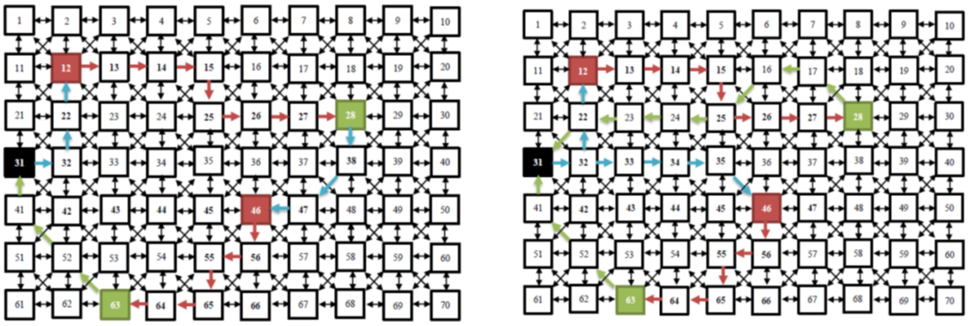
Keywords: *assignment problem, genetic algorithm, multi-criterion optimization, transportation company*

1. Introduction

The transport of the cargo in the full truckload system [15] is defined as transport of one or several shipments directly from one supplier to one recipient, without transhipment at intermediate terminals. The supplier in the full truckload system has the entire cargo area of the vehicle. The issue of the assignment in full truckloads appears when the vehicle leaves the base and go to transport tasks (Fig. 1 – the base 31). Therefore, it is problematic to indicate the tasks performed during the departure of vehicles from the base. The assignment moment also occurs after the current transport task is performed (Fig. 1, e.g. transport task described by elements 12-15, 25-27). The decision problem is to indicate the next task to be implemented (in Fig. 1, the element 46, which is the beginning of the new task 46, 56, 55, 65, 64, 63) or return the vehicle to the base. The return of the vehicle to the base is forced in situations of incompatibility of the capacity of the vehicle with the size of subsequent tasks; completion of all tasks ordered or limited time of completing tasks. Another problem in the issue of the assignment is the designation of transport tasks. The task can be defined as the driving distance of vehicles from the point of picking up the cargo to the ending point. The number of tasks to be carried out is known, while the course of these tasks (routes) should be determined in the first place in order to be able to assign them to vehicles. The course of the route depends on tonnage restrictions, driving times, or travel costs on given sections, as well as travel times through point elements of the transport network, e.g. city, intersections.

In Fig. 1 a) two assignment moments (blue arrows) can be seen, i.e. the assignment between the base 31 and the task beginning with the points 12, and the second assignment moment between the ending point 28 of the task completed by the vehicle and the starting point of the next task 46. This type of assignment can be defined as the base – tasks – the base. These allocations are carried out with one type of vehicle.

The next type of assignment is assignments defined as the base – task – the base. In such types of assignments, the vehicle leaves the base and performs only one task and then returns back to the base. These types of assignments involve additional vehicles and drivers. Examples of these assignments are shown in Fig. 1, b) (the assignment routes are marked with blue arrows, the route of returning to the base by green arrows).



a) b)

Fig. 1. The assignment problem in the full truck load system a) the base – the tasks – the base b) the base – the task – the base

The number of cells (point elements) in the structures shown in Fig. 1 a), b) depends on the accuracy of the mapping of the transport network. When the urban agglomeration is considered, the elements of this structure may be intersections, in the case of domestic transport these elements may be defined as cities. Each element of the structure has characteristics regarding the travel time through this element, additionally the connections between these elements are characterized by the permissible loads on given sections of the route.

The aim of the assignment is to determine the number of the vehicles, which perform the tasks in such a way that the route in which the tasks are carried out should be minimal in the context of the adapted criteria functions, e.g. the completion time.

The presented problem is considered in the context of multi-criterion optimization. The current multi-criterion optimization algorithms [2, 5] cannot be used in this problem due to the specific character of the problem. The decision variables in the mathematical model are of different types (e.g. the binary form – the connection between the point elements, the number of vehicles), which requires developing a new algorithm suitable for the presented problem. Additionally the complexity of the problem is underlined by the fact that the problem is a multi-stage problem, which means the results of the first stage (the stage in which the tasks are determined) influence the results generated in the second stage (the stage in which the assignments the base – task – base or the base – tasks – base are determined).

In consequence, the genetic algorithm has been developed in order to solve the task assignment of the vehicles for the transportation company. The genetic algorithm generates the results in a quick way and therefore, it was selected for this problem. Although the genetic algorithms, which belong to a group of heuristic algorithms, do not guarantee the optimal solution, they enable to close on the optimal solutions so called suboptimal ones. Despite this inconvenience, the genetic algorithms are a practical tool for optimization and they are used in a variety of complex decision making problems e.g. vehicles routing problems [11, 13, 14], location problems [4, 6, 7, 8], internal transport processes [9].

In the literature, the assignment problem is presented in the context of single or multi-criteria decision problems [1, 12]. The algorithms and methods, which solve the analysed problem, refer to the discrete optimization. The decision variables take the binary form. It emphasizes the fact that the current optimization algorithms cannot be used in the assignment problem of vehicles to tasks in the transportation company in view of specific character of the problem, e.g. different types of the decision variables.

One can underline that the author of this publication has not found the analysed problem in the literature. The presented genetic algorithm is an innovative solution in the field of the task assignment of the vehicles for the transportation company. The aim of the work is to initiate the problem of vehicle assignment in these companies.

2. The genetic algorithm for the assignment problem in the transportation company

The genetic algorithm has been used to solve the multi-criterion assignment problem of the vehicles to the tasks for the transport of the cargo in the full truckload system. Forming the genetic algorithm is advisable to define the chromosome structure, the adaptation function, the cross-linking process, and the mutation [10]. The crossover process and the mutation are reiterated a given number of times, until the stop condition has been achieved. The stop condition for the developed algorithm is the fixed iteration number. In the selection process, the roulette method was adopted, while the cross-linking process and the mutation occur with a defined likelihood set at the beginning of functioning the algorithm. The linear scaling was used in order to prevent the early convergence of the algorithm [3].

The structure of the chromosome was presented as a matrix defining the driving routes of individual vehicles. Based on this structure, the transport tasks, the assignments, and the number of vehicles performing these tasks are designated. Transformation of an exemplary assignment of one vehicle to two tasks (Fig. 2) was saved as a chromosome matrix and shown in Fig. 3 (red arrows – tasks, blue arrows – the assignment, green arrows – the return to the base).

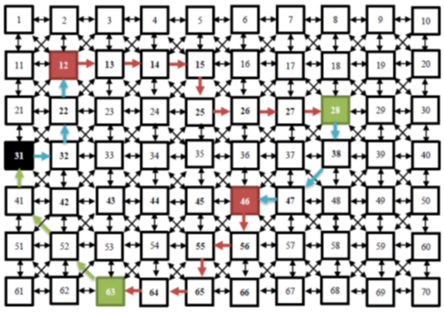


Fig. 2. The assignment the base – the tasks – the base

<table>
<tr><th colspan="10">Task route 1</th><th colspan="10">Task route 2</th><th colspan="10">Assignment</th><th colspan="10">Route to the base</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td colspan="10">Task routes:</td><td colspan="30">Types of vehicles performing tasks</td></tr>
<tr><td colspan="10">Task route 1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td colspan="10">Task route 2</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

Fig. 3. The structure of the chromosome

The matrix structure of the chromosome processed by the genetic algorithm consists of the following substructures: Task route 1, Task route 2, Assignment, Route to the base. For the analysed assignment (Fig. 2), it was assumed that the tasks are carried out with one type of the vehicle (Task route 1 and 2 has values “1” in the window: Types of vehicles performing tasks).

The number of substructures – Task route n – depends on the size of tasks assigned for implementation. Each route is characterized by information about the type of the vehicle that is used to implement it. Types of vehicles performing the tasks are possible to indicate after analysing the amount of the transported cargo in each route. For example, for a task in which 40 pallet units should be transported, two task routes implemented by 30 and 15 pallet trucks can be determined.

The number of substructures Assignment and Route to the base depends on the number of vehicle types performing the designated tasks. In the case of, for example, two types of vehicles, two substructures are introduced, each for a particular type of vehicle, respectively.

It was assumed that in the case of two or more types of vehicles performing the assigned tasks, the first assignment is carried out by the vehicle with the highest load /capacity (the first substructure Assignment); in next substructures' vehicles with smaller capacities are assigned.

Only task routes, which are adapted to the current capacity of the vehicle, can be included in the assignment (in the analysed case – Task route 1 and 2, Fig. 2).

All substructures in the chromosome determine the first type of decision variables, i.e. the binary variables that designate the routes of vehicles. Additionally the number of the returns of vehicles to the base in the structure Route to the base determines the number of vehicle, i.e. the second type of the decision variables.

Cells of the substructure Assignment and Route to the base can assume natural numbers, e.g. a matrix cell with the value of 2 means double vehicle travel via the given element of the transport network.

The routes in the chromosome are designated in the random way. In order to determine routes in the chromosome each cell of the substructure must be marked by the coordinates *x* and *y*, Fig. 4. It should be remembered that the choice of the route between individual elements of the transport network for the substructures Task routes depends on the tonnage restrictions. Additionally for the substructure Assignment, time of task completion must be met.

Task route 1										Coordinates of task route elements (*x*,*y*)									
0	0	0	0	0	0	0	0	0	0	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	(1,7)	(1,8)	(1,9)	(1,10)
0	**1**	**1**	**1**	**1**	0	0	0	0	0	(2,1)	**(2,2)**	**(2,3)**	**(2,4)**	**(2,5)**	(2,6)	(2,7)	(2,8)	(2,9)	(2,10)
0	0	0	0	**1**	**1**	**1**	**1**	0	0	(3,1)	(3,2)	(3,3)	(3,4)	**(3,5)**	**(3,6)**	**(3,7)**	**(3,8)**	(3,9)	(3,10)
0	0	0	0	0	0	0	0	0	0	(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	(4,6)	(4,7)	(4,8)	(4,9)	(4,10)
0	0	0	0	0	0	0	0	0	0	(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)	(5,7)	(5,8)	(5,9)	(5,10)
0	0	0	0	0	0	0	0	0	0	(6,1)	(6,2)	(6,3)	(6,4)	(6,5)	(6,6)	(6,7)	(6,8)	(6,9)	(6,10)
0	0	0	0	0	0	0	0	0	0	(7,1)	(7,2)	(7,3)	(7,4)	(7,5)	(7,6)	(7,7)	(7,8)	(7,9)	(7,10)

Fig. 4. The coordinates of task route

The next step of the genetic algorithm is to determine an adaptation function. On the basis of the adaptation function, the genetic algorithms designate the final solution. What is more, the genetic algorithms look for the maximal solution. In order to take into account the mentioned aspect and different criterion functions the adaptation function for k – the structure of the matrix $\mathbf{M}(t, k)$ must take the following form ($\boldsymbol{K} = \{1, \ldots, k, \ldots, K\}$-the set of the structures $\mathbf{M}(t, k)$ in the population, t – iteration):

$$\mathrm{F}(k,t) = \frac{\mathrm{F1min}}{\mathrm{F1}(k,t)} + \frac{\mathrm{F2min}}{\mathrm{F2}(k,t)} \longrightarrow \max . \tag{1}$$

In order to be able to add the values of all criterion functions the adaptation function needs to be presented as the sum of quotients where e.g.: F1min determines the structure of the minimum value of the first criterion function from the whole population in a given iteration of algorithm, F1(k,t) determines the value of this criterion function for k – the structure of the matrix M(t,k), F1min and F1(k,t) are calculated with the function which determines the transportation cost, F2min and F2(k,t) determines completion time. The function F(k,t) will reach the maximum value in the case when each function, e.g. F1(k,t) reaches F1min, F2(k,t) reaches F2min and so on.

The reproduction (selection) operation consists in duplicating matrix structures in subsequent

generations (iterations of the algorithm) depending on the adaptation function. The so-called a roulette method has been used which is based on the selection of a new population according to the probability distribution determined on the values of the adaptation function. The principle of the roulette method consists in determining the probability of choosing a single chromosome (matrix structure) from a given population, and then determining the distribution for each chromosome. The next step in the selection algorithm is to draw a number from the range [0, 1]. The *k*-th chromosome is selected based on the value of the probability distribution q_k that meets the formula $q_{k-1} < r \leq q_k$.
The roulette method has the disadvantages:

- it happens that at the beginning of the algorithm's operation there may be above-average solutions and these solutions in the first iterations of the algorithm will dominate the populations, which is an undesirable phenomenon and contributes to the premature convergence of the algorithm,
- at the end of the algorithm's operation, it may happen that the population will maintain diversity, but the average adaptation rate is not much different from the maximum. In this situation, average and best individuals will receive the same number of copies in the next generation.

In order to avoid the disadvantages of the roulette method one should remember about the scaling of the adaptation function F (*k*,*t*), because it prevents the premature convergence of the algorithm to the value of the best individuals in the initial iterations of the algorithm and selecting medium adapted individuals in the final iterations. Linear scaling transforming the adaptation function according to the linear relationship takes the form:

$$F(k,t)' = aF(k,t) + b. \tag{2}$$

The scaling factor coefficients have the following values:

$$a = \frac{(C_{mult} - 1) \cdot f_{sred}}{f_{\max} - f_{sred}},\ b = \frac{f_{sred} \cdot (f_{\max} - C_{mult} \cdot f_{sred})}{f_{\max} - f_{sred}}, \tag{3}$$

where:
C_{mult} – the multiply coefficient, f_{sred}, $f_{\max}$ – the average and maximum population value.

If the negative values are reached by the adaptation function, the coefficients take the form:

$$a = \frac{f_{sred}}{f_{sred} - f_{\min}}, b = \frac{-f_{\min} \cdot f_{sred}}{f_{sred} - f_{\min}}. \tag{4}$$

In crossover process, two chromosomes are selected in a random way. In order to carry out the crossover process, it is necessary to determine the crossing parameter. Crossing parameter determining the probability of how many individuals will cross. The probability of crossing is determined at the beginning of the algorithm implementation. Having designated chromosomes for crossing, a process of randomly joining them into pairs takes place. If an odd number of chromosomes is drawn, a randomly chromosome from the population should be added.

The process of crossing consists in drawing a substructure in which the process will be carried out (e.g. Task route) and then drawing two points cutting this substructure, Fig. 5. Between these points, substructure values are swapped for each pair of chromosomes. If an incorrect chromosome is generated after the crossover process, a repair algorithm should be used. The chromosome is correctly when:

- the starting or ending point must be connected with one point of the route,
- each point of the route must be connected with two different points of the route.

In mutation process, the chromosome is selected in a random way. The point of the route without the starting and ending point is selected in a random way and its value is changed from 1 to 0, Fig. 6. If an incorrect chromosome is generated after the mutation process, a repair algorithm should be used.

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

a

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0

b

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

c

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	1	0	1	0	0	0
0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0

d

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

e

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0

f

Fig. 5. The crossover process a),b) structures before c), d) structures after e), f) structures after repairing

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

a

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

b

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

c)

Fig. 6. The mutation process a) structures before b) structures after c) structures after repairing

In order to carry out the mutation process, it is necessary to determine the mutation parameter. Mutation parameter determining the probability of how many individuals will mutate. The probability of mutation is determined at the beginning of the algorithm implementation.

3. The verification of the algorithm

The algorithm was implemented by the use of the real input data in the C# programming language. The base was located in the Siedlce, the first task – the starting point – Łuków, the ending point – Lublin, the second task – the starting point Biała Podlaska, the ending Point – Radzyń Podlaski. It is assumed that the weight of the vehicle with the load meets the tonnage limits on each section of the transportation network, time delays for each city – 15 minutes, time delays for junctions – 2 minutes. Distances between transportation network points were designated by Google Maps. The main cities in the transportation network: Stoczek Łukowski, Zbuczyn, Łosice, Leśna Podlaska, Międzyrzec Podlaski, Lubartów, Parczew, Janów Podlaski. The fuel consumption cost – 5 PLN/km, the permissible speed 80 km/h, time of tasks completion – 4h.

The first step of the implementation of the genetic algorithm is to find the set of the best parameters, which characterize this algorithm. The following combinations of the parameters were analysed: p_{cross} – crossover parameter, p_{mut} – mutation parameter (Tab. 1). The number of the iterations was set to 200. The linear scaling factor for the genetic algorithm accordance with the recommendations of the literature [3] assumes the value 2.0. The results of all the tests for the parameters for two variants are presented in Tab. 2.

Tab. 1. The combination of the parameters for the genetic algorithm

Test	p_{cross}	p_{mut}	Test	p_{cross}	p_{mut}	Test	p_{cross}	p_{mut}
1	0.2	0.01	6	0.2	0.03	11	0.2	0.05
2	0.4	0.01	7	0.4	0.03	12	0.4	0.05
3	0.6	0.01	8	0.6	0.03	13	0.6	0.05
4	0.8	0.01	9	0.8	0.03	14	0.8	0.05
5	1	0.01	10	1	0.03	15	1	0.05

Tab. 2. The results of the genetic algorithm

Test	The best value of population	Test	The best value of population	Test	The best value of the structure of population
1	0.43	6	0.43	11	0.3
2	1.5	7	1.4	12	1.27
3	1.23	8	1.32	13	1.4
4	1.6	9	1.76	14	1.77
5	1.54	10	1.7	15	1.51

In order to verify the correctness of the genetic algorithm (AG), its results (for the best parameters – test 14) were compared with the random values (AL). In each case, the genetic algorithm generated a better solution than the random algorithm. The results are shown in Tab. 3. The solution generated by the genetic algorithm for complex decision problems is a sub-optimal solution, which Tab. 3 confirms. However, considering the complexity of the assignment problem, the solution is accepted from a practical point of view.

Tab. 3. The comparison of algorithms

Item	AG	AL	Item	AG	AL	Item	AG	AL
1	1.76	0.55	1	1.64	0.44	1	1.53	1.2
2	1.67	0.61	2	1.51	0.55	2	1.63	0.45
3	1.68	0.72	3	1.58	0.72	3	1.55	0.44
4	1.73	0.65	4	1.63	0.71	4	1.58	0.33
5	1.63	0.57	5	1.53	0.54	5	1.63	0.57
6	1.71	0.8	6	1.61	0.58	6	1.68	0.68
7	1.53	0.7	7	1.63	1.1	7	1.73	0.83
8	1.61	0.34	8	1.71	1.3	8	1.53	0.89
9	1.68	0.45	9	1.65	0.9	9	1.75	0.79
10	1.73	0.34	10	1.73	1.2	10	1.56	1.2

In the analysed example, the assignments were distinguished: the base – task 1 – the base, the base – task 2 – the base.

4. Conclusion

The aim of the article is to solve the task assignment of the vehicles for the transportation company. The genetic algorithm has been developed in order to solve this problem. The further step in the context of the used algorithm in the assignment problem is to test other methods in the selection process. The early convergence to the sub-optimal solution is blocked by the use of the linear scaling. The results generated by this algorithm depend on numerous factors, e.g.: the parameters of the algorithm, the number of iterations.

The developed algorithm can be used for transportation companies to develop, e.g. the driver's working schedules. The main advantage of this algorithm is that the results are generated in a quick way, which is very important for the production companies. The processes occurring in the transportation companies are dynamic processes. For this reason, this algorithm must be started a few times depending on the volume of the cargo, the vehicle capacities. In this case, the calculation speed plays a huge role, which underlines the utility of this algorithm for the transportation companies.

The generated results by the genetic algorithm are the basis for further work on the development of new algorithms in the context of the examined problem.

References

[1] Burkrd, R., Dell'Amico, M., Marttelo, S., *Assignment problems*, Society for Industrial and Applied Mathematics, 2009.

[2] Ehrgott, M., *Multicriteria optimization*, Springer 2005.

[3] Goldberg, D.,E., *Genetic Algorithms in Search, Optimization, and Machine Learning 1st Edition*, Addison-Wesley Professional, 1989.

[4] Izdebski, M., Jacyna-Gołda, I., Wasiak M., *The application of genetic algorithm for warehouse location in logistic network*, Journal of KONES Powertrain and Transport 23(3): 201-208, 2016.

[5] Jacyna, M., *Multicriteria evaluation of traffic flow distribution in a multimodal transport corridor, taking into account logistics base service*, Archives of Transport 11(3-4): 43-66, 1999.

[6] Jacyna-Gołda, I, Izdebski, M., *The Multi-criteria Decision Support in Choosing the Efficient Location of Warehouses in the Logistic Network*, in: Procedia Engineering, Elsevier BV, Vol. 187, pp. 635-640, 2017.

[7] Jacyna-Gołda, I., Izdebski, M., Szczepański, E., *Assessment of the Method Effectiveness for Choosing the Location of Warehouses in the Supply Network*, Challenge of Transport Telematics Springer, Vol. 640, pp. 84-97, 2016.

[8] Jacyna-Gołda, I., Izdebski, M., Podviezko, A., *Assessment of efficiency of assignment of vehicles to tasks in supply chains: A case study of a municipal company*, Transport, Vol. 32, Iss. 3, pp. 243-251, 2017.

[9] Lewczuk, K., *The concept of genetic programming in organizing internal transport processes*, Archives of Transport 34(2): 61-74, 2015.

[10] Michalczewski, Z., *Genetic algorithms + data structure = evolutionary programs*, Springer 1996.

[11] Ombuki, B., *Multi-Objective Genetic Algorithms for Vehicle Routing Problem with Time Windows*, Applied Intelligence, Vol. 24, Iss. 1, pp. 17-30, 2006.

[12] Pentico, D., W., *Assignment problems: A golden anniversary survey*, European Journal of Operational Research, Vol. 176, Iss. 2, pp. 774-793, 2007.

[13] Shuguang, L., *A hybrid population heuristic for the heterogeneous vehicle routing problems*, Transportation Research Part E: Logistics and Transportation Review, Vol. 54, 2013, pp. 67-78, 2013.

[14] Szczepański, E., Jacyna-Gołda, I., Murawski J., *Genetic algorithms based approach for transhipment HUB location in urban areas*, Archives of Transport 31(3), pp. 73-82, 2014.

[15] Wasiak, M., Jacyna-Gołda, I., *Transport drogowego w łańcuchach dostaw*, *Wyznaczanie* kosztów, PWN, Warszawa 2016.

Manuscript received 10 July 2018; approved for printing 12 September 2018