Date of publication JUN-2-, 2021, date of current version MAY-17, 2021.
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.20.2.2172

Genetic Algorithm Solution for Transfer Robot Operation

OLEKSANDR TSYMBAL, ARTEM BRONNIKOV, YEVHENIJ GUDKOV
Kharkiv National University of Radio Electronics, Nauki Avenue 14, Kharkiv, Ukraine
(e-mail: oleksandr.tsymbal@nure.ua, artem.bronnikov@nure.ua, xme1ez@gmail.com)

Corresponding author: Oleksandr Tsymbal (e-mail: oleksandr.tsymbal@nure.ua).

ABSTRACT The proposed article presents the research of decision-making processes for control system of transport robot, acting inside warehouse. The analysis shows the growing importance of warehouse systems for flexible integrated system. Such warehouse operations, like loading/uploading pallets and sending goods to distribution center can be fully automated by applying robots and some examples are known. Simultaneously, the wide introduction of transport robots with extended possibilities is still limited. Application of intelligent robot can significantly improve properties of warehouse systems, making them closer to Industry 4.0 standards. The functioning of a transport robot can be described as a Travelling Salesmen Problem with numerous solutions that must be effectively limited. Genetic algorithms are proposed as a basis for path generation. The algorithms and software are implemented to model warehouse transfer robots using an Arduino-robot. The results of testing show the effectiveness of the developed algorithms, software and hardware system.

KEYWORDS robotics; warehouse; genetic algorithm; chromosome; decision-making.

I. INTRODUCTION

THE effective organization of internal logistics and use of warehouses are critical for any modern business. While logistic demands vary every three years, warehouses are laid out for nearly twenty years. In conditions of rising logistic requirements, the effective usage of warehouse space is a necessary condition for successful operation and development of an enterprise [1].

Today's reality shows that standard manual storage systems with stackers and automated crane systems have reached the limits of their capabilities. The modern customer wants to reduce operating costs and obtain the maximum efficiency and safety within warehouse space.

Robotic systems for transportation and storage, invented by leading companies in the field of robotics, have become widespread, because of their higher productivity and lower maintenance costs in comparison to systems with human resources [2]. Such complexes work, as a rule, in the Cartesian coordinate system (rectangular or plane) which allows a robot initially located anywhere to reach any point of warehouse. The workspace of the robotic complex is determined only by its kinematics and location in which it works [3].

Modern robotic systems are descendants of warehouse manufacturing systems (WMS) and correspond to current peak of their development. The only difference is, in fact, that now automation is achieved by mobile robots using autonomous movements within the warehouse for loading objects by making independent decisions. The transition to robotic warehouses is a global trend, driven by need to accelerate logistic processes in huge warehouses, where human capabilities, even technically enhanced, seem to have reached their limits. The potential of such logistics is enormous – today it is one of the best examples of the Internet of things adapted to this vital and necessary business [4].

II. DEMANDS FOR WAREHOUSE MODELS AND THEIR APPLICATION

For successful implementation in a wide range of tasks, robots must have mobility, as well as some sort of intelligence to interpret, plan and automatically execute the task, using an onboard computer system [5, 6]. The necessary features include the ability to reach a specified goal in an unstructured workspace, avoiding collisions with stationary obstacles and moving objects [7].

When working in an unknown dynamic workspace, a mobile robot must be able to adapt to changes in the environment, to respond to unexpected situations and to act on the basis of previous experience. Thus, robot needs control systems with elements of artificial intelligence [8].

The proposed article simulates the Automated Storage and Retrieval System (ASRS) in a warehouse, where the robot moves pallets, from places of loading and unloading, a place for the robot's standby mode, with black lines to indicate possible routes. To select the optimal location of the transport robot in the warehouse, it's possible to use the methods for distribution centers allocation optimization [1].

III. DECISION-MAKING MODELS FOR ASRS USING GENETIC ALGORITHM

The applications of genetic algorithms (GAs) for solving the optimization problems were developed by D. Holland in 1975, and the fundamental impact of GAs was introduced by D. Goldberg. Later, GAs were successfully applied for conducting synthesis, constructing timetables, routing transport vehicles in different kinds of workspaces [8, 9], for manipulation tasks [10, 11], and motion of humans and robots [12, 13].

The route planning problem for a robot's movements in the workspace of a warehouse is a type of Travelling Salesman Problem (TSP), while the methods of GAs are also suitable. Here, avoiding obstacles has become an important research problem [14, 15]. The dynamics of a workspace was also taken into account [16, 17]. To solve TPS with GAs, special methods of solutions were developed involving presentations (coding) with corresponding problem-oriented genetic operators [18, 19]. Practically, there are four basic methods of tour presentation for TSP solving by GA: neighborhood, ordered, route, and matrix.

While classic crossing and mutation operators aren't applied for such presentations, each has own strongly different "genetic" operators. Mutation operator is defined by single replacements of neighbor's positions inside a tour. Crossovers can create ancestors with no solution (invalid solution). Invalid solutions can be useful to create additional options for the population, while simultaneously slowing down or even preventing convergence of a GA. One solution to this problem is in restoration of invalid solutions.

The fitness function corresponds to the distance travelled by a salesman on a route, represented by a chromosome. Because this value must be minimized, the formula of fitness function for j-chromosome can be set in the following way:

$$f_j = 1.1 \cdot d_{\max} - d_j, \qquad (1)$$

with $d_{\max}$ equal to the length of maximal route for the current population, d_j the route length presented by j-chromosome. A greater fitness function value indicates a shorter route.

Neighborhood presentation. With neighborhood presentation, the route is stored as a list of *n* locations. Location *j* is on position *i* only if route goes from location *i* to location *j*. For example, the route with a set of ordered location visits is shown in formula (2):

$$1 \to 2 \to 4 \to 3 \to 8 \to 5 \to 9 \to 6 \to 7 \to 1. \qquad (2)$$

This route can be represented as a chromosome by the following vector:

$$P = [2\,4\,8\,3\,9\,7\,1\,5\,6]. \qquad (3)$$

Such view corresponds to neighborhood method, described above – Location *j* is on position i only if route goes from location i to location *j*.

Neighborhood presentation does not support the classic crossover operation because of constant creation of unacceptable routes. To avoid these, three operators are developed: crossover for alternative edges; crossover for subtour chunks; heuristic crossovers.

The basic benefit of ordered presentation is in application of classic crossover. An ordered path realization describes a route as a list of *n* sites, in which *i*-element of the list is a number from 1 to $n-i+1$. The idea of an ordered path realization can be explained as follows. There is the ordered list of sites, which connects routes by their numbers. Suppose that such ordered list of desired routes is given by (4):

$$C = [1\,2\,3\,4\,5\,6\,7\,8\,9]. \qquad (4)$$

In this case the fitness function in formula 1 gives (5) according to formula (1):

$$L = [1\,1\,2\,1\,4\,1\,3\,1\,1]. \qquad (5)$$

It can be explained as follows. Because first element of *L* is equal to 1, the first element of *C* is considered as the starting location of the route (site 1) and excluded from the list giving *C* = [2 3 4 5 6 7 8 9].

The next element in *L* is also 1, so we take the first number from refreshed list *C* = [2 3 4 5 6 7 8 9]. As first site from *C* is excluded, the next site is 2. It is also excluded from list C.

Next number in *L* is 1. We take from *C* = [3 4 5 6 7 8 9] the second site in order. It is site number 4. It is also excluded. The route becomes: 1 → 2 → 4.

Next number in *L* is 1. We take from *C* = [3 5 6 7 8 9] the first site (with num 3). Route: 1→2→4 →3.

Next number in *L* is 4. We take 4-th site from the rest of

C = [5 6 7 8 9] (it is 8). Route: 1→2→ 4→3→8.

Next number in *L* is 1. We take from list C = [5 6 7 9] the first site (site with number 5). The route is: 1 →2 → 4 → 3 → 8 → 5.

Next number in list *L* is 3. We take the third site from list *C*=[6 7 9] (it is site 9) and exclude it from list *C*. The route is: 1 → 2 → 4 → 3 → 8 → 5 → 9.

Next number in list *L* is 1. We take first site form current list *C* = [6 7] (it is site number 6) and remove it from list *C*. The route then becomes: 1 →2 → 4 → 3 → 8→ 5→9→6.

The last number in list *L* is always 1 to return to the initial location. So, we take last site that is from current list *C* = [7], and exclude it from list *C*. The resulting route is then 1→2→4→3→8→ 5→9→6→7→1.

Any two routes in ordered presentation, dissected in any position and joined together, can create two ancestors, each of which corresponds to a valid route.

For mutations of this presentation there are used changes with probability p_m for *i*- element of ordered presentation with numbers from 1 to $n_i - 1$.

Path presentation is natural for describing routes because of its discreteness. For example, tour of (6) will be presented as vector (7).

$$1 \to 2 \to 4 \to 3 \to 8 \to 5 \to 9 \to 6 \to 7 \to 1, \quad (6)$$

$$P = [1\,2\,4\,3\,8\,5\,9\,6\,7]. \quad (7)$$

For possible path realizations it needs to have the crossovers, developed for this realization, including: partially shown crossover; ordered crossover; cycle crossover; crossover of edge recombination; improved crossover of edge recombination; changed crossover.

Matrix presentation. To code the chromosome, the binary matrix $V = [v_{ct}]$ can be used with $v_{ct} = 1$, if site “c” is visited at moment t (which is in route on position t), or $v_{ct} = 0$ in other case.

The route between positions “ABCDE” is presented by formula (8).

$$A \to B \to E \to D \to C. \quad (8)$$

The matrix presentation of a route can be unwrapped to form the one-hot encoded chromosome vector shown in (9).

$$P = [10000\ 01000\ 00010\ 00100\ 00100] \quad (9)$$

GAs can create some chromosomes, which have no path presentations (invalid solutions). This can happen during process of initial population creation as during implementation of classic genetic operators. For some methods of solution presentation there are developed the tools of crossing and mutation. Because of their complexity, algorithms use the symbol coding of chromosomes, described above.

The matrix presentation for route presented by formula (8) looks as in Fig. 1.

	1	2	3	4	5
A	1	0	0	0	0
B	0	1	0	0	0
C	0	0	0	0	1
D	0	0	0	1	0
E	0	0	1	0	0

Figure 1. Binary matrix representation of route A->B->C->D->E.

After analysis of advantages and disadvantages of GAs, the model of search algorithm for best route of mobile robot must be developed [20].

As a base there is used the hybrid algorithm of minimal route search for TSP with application of GAs on initial stages and with symmetric algorithm for narrowed area of search. The essence of the symmetric algorithm is in fact, that there are considered the alternative routes from starting point to the final. If alternatives do not correspond to symmetry, the asymmetric route is dropped.

Below we have parameters for GA: fitness function (accommodation function) from formula (1); route presentation; changed crossover; greedy mutation; selection of the better of two ancestors.

To make all the found solutions valid, the directions of transitions from every point must be defined. For example, if to limit the conditions, that one point can be source of 4 motion directions, such condition can be set as a matrix in Fig. 2, with 5 – the number of a current point; 3, 7, 10 – possible directions of motion.

0	0	0
3	5	7
0	10	0

Figure 2. Motion directions from point 5

With GA there is created the initial population, the routes with best lengths. It will be the base to form the parts of route, which have no changes under more accurate optimization. For example, the algorithm, making the computations, finds that if parts 1 and 2 in Fig. 3 are presented for the most of solutions, then these parts will be considered as constants and for symmetric method they will be unchanged with predefined names and lengths.

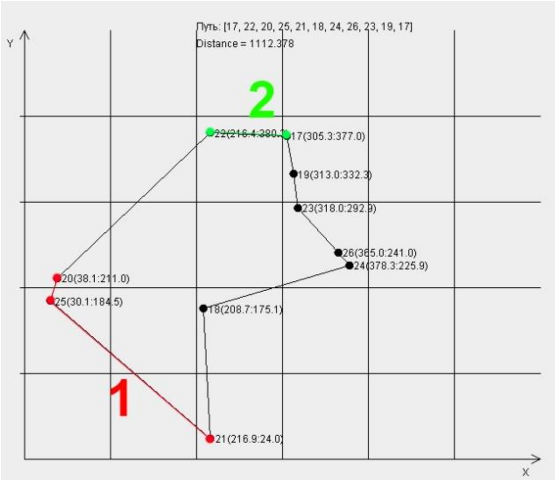

Figure 3. Result of the best route search

The next stage is a search for the best route with application of direct symmetric algorithm. The example of search among the possible routes is shown in Fig. 4. The goal is to find the shortest route from point 0 to point 8. The length of every part is defined for every edge.

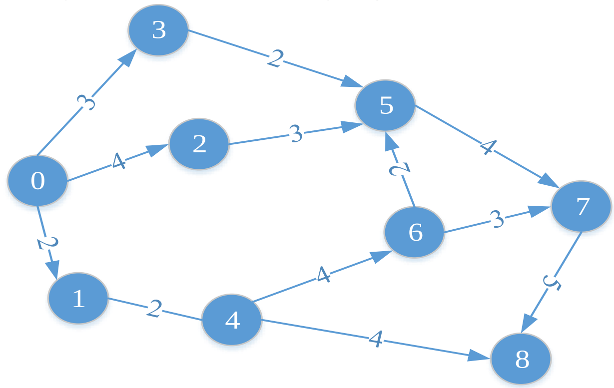

Figure 4. The graph of routes between points

The route length between objects i and j is denoted c_{ij}. The minimal route from starting location 0 to any point of workspace is $Q_0=0$, then the search for minimal distance is provided due to formula (10):

$$Q_j = \min(Q_i + c_{ij}) . \qquad (10)$$

So, route length depends on the length of the previous parts of the route and the cost of the current part, while minimal distance from available variants is selected.

Computations of route lengths from point 0 to point 8 are shown by (11, 12):

$$\begin{aligned} &Q_0 = 0 ; \\ &Q_1 = Q_0 + c_{01} = 0 + 2 = 2 ; \\ &Q_2 = Q_0 + c_{02} = 0 + 4 = 4 ; \end{aligned} \qquad (11)$$

$$\begin{aligned} &Q_3 = Q_0 + c_{01} = 0 + 3 = 3 ; \\ &Q_4 = Q_0 + c_{04} = 2 + 2 = 2 ; \\ &Q_5 = \min(Q_2 + c_{25}; Q_3 + c_{35}) = \min(4 + 3; 3 + 2) = 5 ; \\ &Q_6 = \min(Q_4 + c_{46}; Q_5 + c_{56}) = \min(4 + 6; 5 + 2) = 7 ; \\ &Q_7 = \min(Q_5 + c_{57}; Q_6 + c_{67}) = \min(5 + 4; 7 + 3) = 9 ; \\ &Q_5 = \min(Q_4 + c_{48}; Q_7 + c_{78}) = \min(4 + 4; 9 + 5) = 8 . \end{aligned} \qquad (12)$$

Therefore, after execution of all required computations there is found the route with minimal distance from point 0 to 8, which is equal to 8 units. The result of computations is shown in Fig. 5.

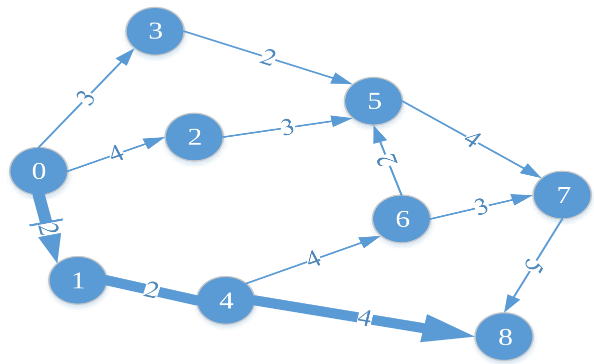

Figure 5. The result of calculations by direct symmetric method

IV. DEVELOPMENT OF HARDWARE AND SOFTWARE FOR ASRS MODELS

The developed platform of mobile robot (MR) is a type of forklift (Fig. 6). To operate the lifting mechanism, two stepper motors 28BYJ-48 5V are used. Lifting is accomplished through the use of a screw gear transmission of sliding – the conversion of torque into translational movement.

MR is built of plexiglass (4 mm), which is sufficient for testing a system. The frame consists of a platform frame and a forklift frame. MR has the ability to navigate in space using line detection sensors. Line detection sensors are based on 4 TCRT5000 sensors, assembled as sensor modules.

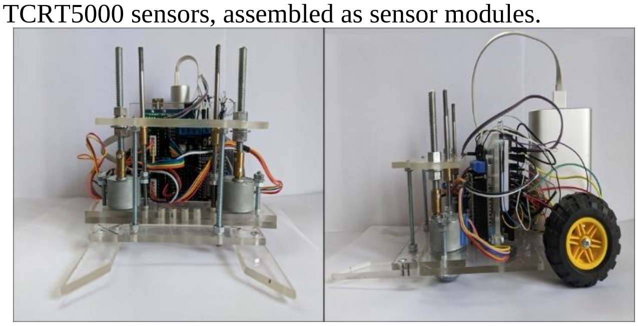

Figure 6. Forklift truck loader

To move on the route, the MR uses 4 sensors to detect special markings on the floor. Two central sensors are used directly for line following movement, and two other sensors for detecting intersections. The position of the sensors on the MP platform during movement is shown schematically in Fig. 7. The movement algorithm is as follows: start moving along a black line, if sensor 2 has lost the line, turn right, if sensor 3 has lost the line, turn left. In case when the sensors 1, 4 detect the line separately, or together, an intersection is found, and it is necessary to update the current position.

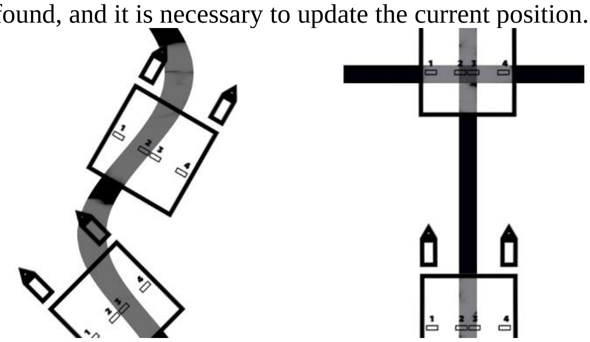

Figure 7. The position of MR sensors during movement

The problem of the effective MR movement within a warehouse or work area is an important part in the design of an intelligent warehouse system [19, 21]. To verify the developed algorithm for MR movement, a forklift robot uses markings on the floor of the working area for orientation in space. The markings are given in the form of black lines to contrast with the white floor. The layout of the configuration for testing contains multiple routes, 8 pallet locations marked by orange color, and 1 parking place for MPs, marked by gray (Fig. 7).

Each pallet location has a size of 140x70 mm, the payload is stored on pallets and has a special design for efficient loading / unloading and for the movement of pallets within the warehouse.

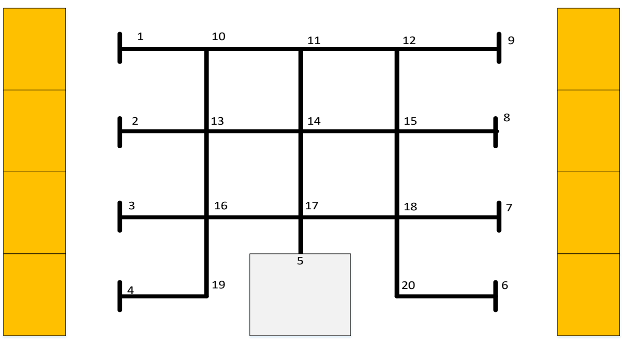

Figure 8. Warehouse layout for algorithm testing

According to Fig. 8, each important point has the form of an intersection, with 2, 3 and 4 possible options for further movement. The intersections under numbers 1, 2, 3, 4, 10, 14, 15, 16, 17 should only be used as stop point for loading / unloading or movement in the opposite direction, fifth point for MR in a standby mode. Through pairs of points 3-9 and 13-17, movement is carried out in only two directions. Through 4, 7 and 11, movement can be performed in 3 directions. Through points 5, 8, 12, 6, 9, 13, movement can be performed in 4 directions.

MR loader is equipped with 4 line detection sensors, in order to “teach” the robot to move between control points, it is necessary to use the following algorithm. Each control point is supplied to the algorithm in the form of a 3x3 matrix, where in the center number is the current position, the points around the center correspond to points from which it is possible to get from the current one.

The route is represented as a sequence of points, starting from the current location to the end point of the route. An example of a route from checkpoint 10 to position 14 is given below:

$$L=[10,9,8,7,11,14].$$

MR operation algorithm consists of automatic movement within the established structure, loading / unloading in accordance with the established task.

When MR is first turned on or in standby mode, it takes its original place as the parking point. When the power is turned on, the robot joins a Wi-Fi network and waits for a task. After MR is connected, the operator sets the tasks and transfers them to MR as a sequence of points from the current location, namely, the first point, where robot must load the pallet, the second point, where robot must unload the pallet, the last point – parking place of robot (defined automatically).

Then MR calculates its optimal route between the points, after which it starts moving along the established route, and performs loading / unloading actions. After completing the tasks, the MR returns to the starting point to wait for a new task.

To efficiently route the MR movements, the developed algorithm was based on the use of GA for TSP. The algorithm has a key difference from the classical TSP, the MR does not need to search for a route using all points, but only those which give the shortest route.

The algorithm consists of two parts:

– genetic algorithm at the beginning of calculations to narrow the area of routes, to search for a more accurate algorithm;

– exact method of “direct symmetric algorithm” to find the best solution to the problem.

The implementation of the genetic algorithm can be divided into several stages, which will be described later. For initial conditions, we take: population size of 10 individuals, the probability of crossing 0.5, the probability of mutation 0.2.

Stage 1 – Determination of the initial population. At the stage of formation of the initial population, knowing the number of individuals in the population and accepting the fact that the approximate location of the optimum is unknown, 10 individuals for the formation of the population will be randomly selected.

Stage 2 – Estimation of individuals of the population. To assess individuals of the GA population, it is necessary to use the fitness function for each individual.

Stage 3 – Selection, Parental pair for crossbreeding is selected as to the panmixia strategy that means equal probability of selection.

Stage 4 – Crossbreeding. A probabilistic crossover operator is applied to two specific decisions of the parents, which forms new 2 successor decisions on their basis. The selected persons are crossed with a given probability Pc.

Stage 5 – “greedy” mutation, based on greedy strategy for travelling salesman problem.

Stage 6 – Formation of a new generation. After crossing and mutation, the formation of a new generation occurs as follows: the population of the previous generation and the population of new individuals are obtained as a result of crossing and mutation. These are sorted in decreasing order of fitness function, and only the best individuals from the previous generation and individuals of the newly formed generation fall into the new population.

Stage 7 – Execution of the algorithm stops after the completion of a specified number of epochs of evolution. If the number of specified epochs is not completed, go to stage 2. After the calculation using the genetic algorithm, it is performed using the direct symmetric method, which is used in a situation where it is needed to find the shortest open route.

V. RESULTS

Application of the proposed GA to calculate the paths of MR within workspace of robotic warehouse was simulated by developed software.

The example tasks and results of simulation are presented by the following figures.

Symmetrical matrix in Fig. 9 presents possible variants of routes, while different routes can be seen in Figs. 9-12.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0
4	2	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0
5	3	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0
6	4	5	0	0	0	0	6	0	7	0	0	0	0	0	0	0	0	0	0
7	5	0	5	0	0	6	0	6	0	7	0	0	0	0	0	0	0	0	0
8	6	0	0	5	11	0	6	0	0	0	7	0	0	0	0	0	0	0	0
9	7	0	0	0	0	7	0	0	0	6	0	0	7	0	0	0	0	0	0
10	8	0	0	0	0	0	7	0	6	0	6	0	0	7	0	0	0	0	0
11	9	0	0	0	0	0	0	7	0	6	0	2	0	0	7	0	0	0	0
12	10	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
13	11	0	0	0	0	0	0	0	7	0	0	0	0	6	0	5	0	0	0
14	12	0	0	0	0	0	0	0	0	7	0	0	6	0	6	0	5	0	0
15	13	0	0	0	0	0	0	0	0	0	7	0	0	6	0	0	0	5	11
16	14	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0
17	15	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0
18	16	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0
19	17	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0

Figure 9. Presentation of possible route by symmetric matrix

Calculations of route from point 0 to point 17 give different variants.

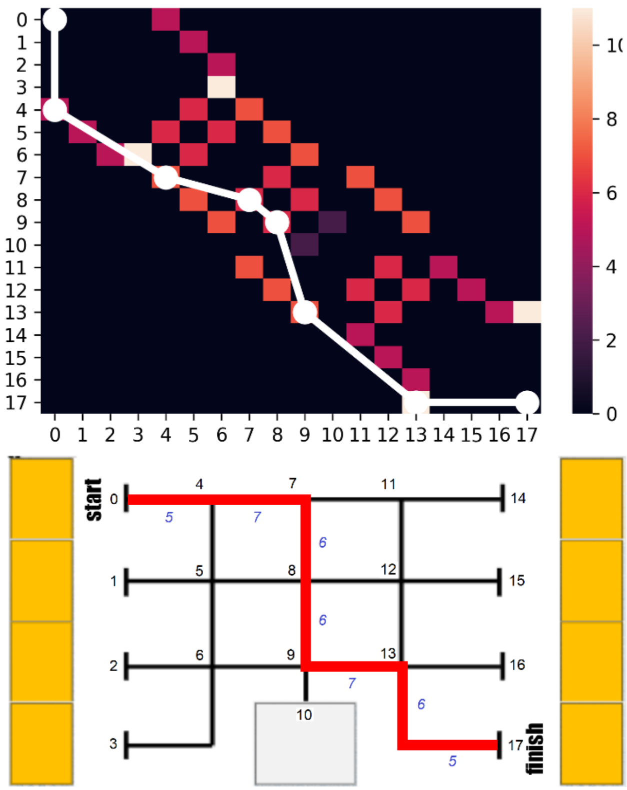

Figure 10. Path [0 4 7 8 9 13 17], length=42

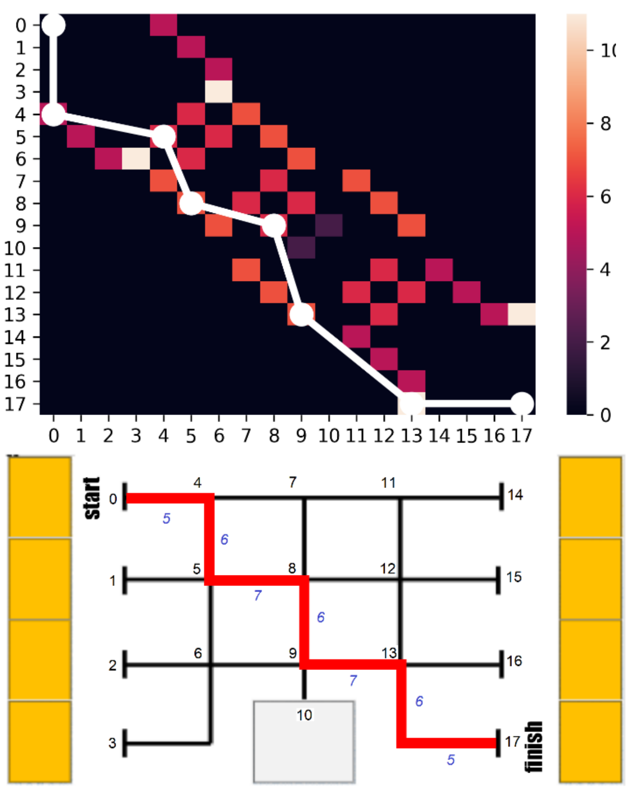

Figure 11. Path [0 4 5 8 9 13 17], length=42

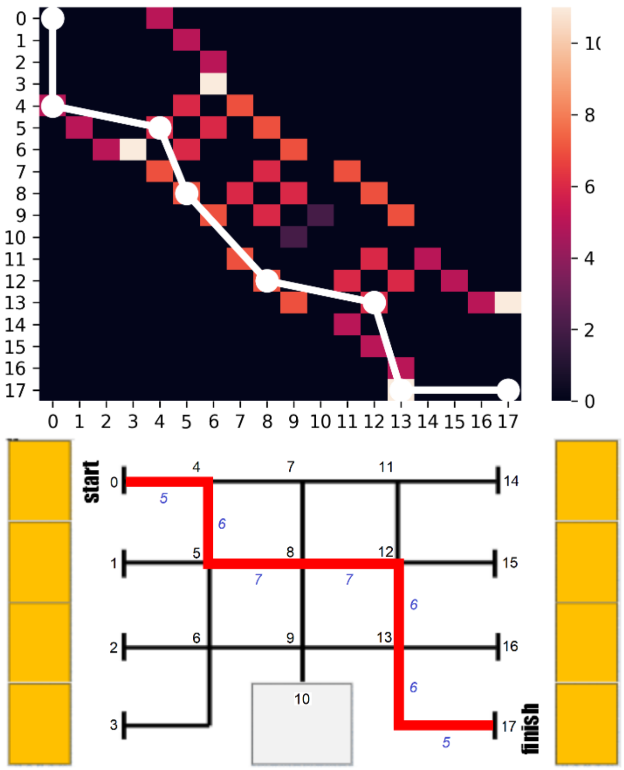

Figure 12. Path [0 4 5 8 12 13 17], length=42

If it is needed, warehouse can be simulated with a larger number of points and scaled up, when all internal distances and centers of cells are fixed.

For the presented initial conditions, presented by figures above, the best result is found by 3 iterations on the average, for case of 40 points – by 6-7 iterations, for case of 100 points – by 12 on the average.

To implement the model of warehouse, there is developed the software, which is used to develop tasks for the forklift. It is Windows-compatible and based on Java language.

To check the route calculation time, a series of test calculations was performed based on the NodeMCU processor with a clock frequency of 80 MHz. The average time of calculations is 0.3 sec.

For comparison, the use of only direct symmetric method for searching the effective route for a similar set of input parameters takes more than 0.5 s.

To test the operation of the algorithm on more points, the size of the composition was increased by 10 times. The time taken to solve the problem by the developed algorithm is about 0.5 s, and when using only the direct symmetric method – 1.8 s.

After execution and comparison of the effectiveness of the developed algorithm, we can conclude that increasing size of the composition leads to the calculation increases linearly in time.

To improve the MR system, a special function was added that allows storing of calculated routes in the database. This feature makes it possible to reduce the search time for an effective route to fractions of a second. With the presence of this function, the effectiveness of the algorithm becomes doubtful, but testing shows that the proposed algorithm is quite effective when the route of the MR needs to be rebuilt during movement, for example, in the case of impossibility of traveling along the established route because of obstacles.

The effectiveness of the developed algorithm increases with the number of points that the robot must visit in one cycle. For example, if an MR is ordered by sequence of 15 points for minimal time, then the number of route options is calculated as a factor of 15! – the number of possible route combinations will be equal to approximately 10^{14}. Thus, if 1 ms is spent on calculating one route option, then 15 days will be spent on calculation of all possible routes. If the sequence of points is increased to 20, the time required for the calculation will be 77,000 years.

Using GA allows at the initial stage to reduce the list of possible routes so much that it will take about 10-15 seconds to search for a reasonable, although not optimal, route.

Testing of a set of sequences within the designed structure suggests that the developed algorithm in 98% of cases determines the minimal route among many possible.

VI. CONCLUSIONS

The main task of the research is in the development of effective algorithm to find routes between checkpoints of MR loader for simulation of robotic warehouse on transfer operations. To perform route construction, calculations based on the GA and the direct symmetric method are performed. To suggest theoretical results there was developed the model of mobile robotic transfer system, acting within workspace of simulated warehouse. Experiments with robotic system prove their effectiveness for small models and can be the base for more complex configurations of warehouses.

References

[1] E. Mikhailov, B. Remenyuk, “Optimize the placement warehouse transport system,” *Journal of Electrotechnic and Computer systems*, no. 18 (94), pp. 60–64, 2015.

[2] S. Thrun, *Probabilistic Robotics*, The MIT Press, 2005, 667 p.

[3] A. Khor, A. Yong, *Robotics in Supply chain*, in: L. Huan, S. Raman (Eds.), Emerging Trends of Technologies in Supply Chain Management, WOU Press, 2018, pp. 25-38.

[4] I. Nevliudov, O. Tsymbal, A. Bronnikov, “Intelligent means in the system of managing a manufacturing agent,” *Innovative Technologies and Scientific Solutions for Industries*, no. 1 (3), pp. 33–47, 2018, https://doi.org/10.30837/2522-9818.2018.3.033.

[5] O. Tsymbal, A. Bronnikov, “Decision-making information technology for flexible integrated manufacturing,” *Innovative Technologies and Scientific Solutions for Industries*, no. 2 (8), pp. 105–112, 2019, https://doi.org/10.30837/2522-9818.2019.8.105.

[6] V. Vacic, T. Sobh, “Vehicle routing problem with time windows,” *International Journal of Computing*, vol. 3, issue 2, pp. 72–80, 2004, https://doi.org/10.47839/ijc.3.2.289.

[7] V. Red’ko, “Interaction between learning and evolution in population of autonomous agents,” *International Journal of Computing*, vol. 12, issue 1, pp. 42–47, 2013, https://doi.org/10.47839/ijc.12.1.586.

[8] C.E. Thomaz, M.A.C. Pacheco, M.M.B.R. Vellasco, “Mobile robot path planning using genetic algorithms,” In: Mira J., Sánchez-Andrés J.V. (eds) *Foundations and Tools for Neural Modeling. IWANN 1999. Lecture Notes in Computer Science*, vol. 1606. Springer, Berlin, Heidelberg, https://doi.org/10.1007/BFb0098225.

[9] C. Zheng, H. Liu, M. Ge, Y. Liu, "A novel maze representation approach for finding filled path of a mobile robot," *Proceedings of the International Conference on Computer, Network, Communication and Information Systems (CNCI'2019), Advances in Computer Science Research*, vol. 88, 2019, pp. 664-673, https://doi.org/10.2991/cnci-19.2019.92.

[10] S. Števo, I. Sekaj, M. Dekan, "Optimization of robotic arm trajectory using genetic algorithm," *Proceedings of the 19th World Congress the International Federation of Automatic Control Cape Town, South Africa*, August 24-29, 2014, pp. 1748-1753, https://doi.org/10.3182/20140824-6-ZA-1003.01073.

[11] O. Chocrone, P. Bidaud, "Evolutional algorithms in kinematic design of robotic systems," *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Grenoble, France, Sept. 8-12, 1997, pp. 279-286.

[12] B. Eldridge, A. Maciejewski, "Using genetic algorithms to optimize social robot behavior for improved pedestrian flow," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Waikoloa, Hawaii, October 10-12, 2005, pp. 525-529.

[13] J. Xin, J. Zhong, F. Yang, Y. Cui, J. Sheng, "An improved genetic algorithm for path-planning of unmanned surface vehicle," *Sensors*, vol. 19, paper 2640, 2019, https://doi.org/10.3390/s19112640.

[14] G. Nagib, W. Gharieb, "Path planning for a mobile robot using Genetic Algorithms," *Proceedings of the International Conference on Electrical, Electronic and Computer Engineering (ICEEC'04*), 2004, pp. 185-190.

[15] K. Kwaśniewski, Z. Gosiewski, "Genetic algorithm for mobile robot route planning with obstacle avoidance," *Acta Mechanica et Automatica*, vol. 12 no. 2, pp. 151-159, 2018, https://doi.org/10.2478/ama-2018-0024.

[16] M. Fetanat, S. Haghzad, S. Shouraki, "Optimization of dynamic mobile robot path planning based on evolutionary methods," *Proceedings of the AI & Robotics (IRANOPEN 2015)*, 2015, pp. 1–7, https://doi.org/10.1109/RIOS.2015.7270743.

[17] A. Balaji, P. Padmanabhan, "Multi-objective optimization of mobile robot path planning using genetic algorithm," *IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE)*, pp. 17-20, 2011.

[18] K. Sedighi, K. Ashenayi, T. Manikas, R. Wainwright, H. Tai, "Autonomous local path planning for a mobile robot using a genetic algorithm," *Proceedings of the Congress on Evolutionary Computation*, Portland, OR, 19–23 June 2004, pp. 1338–1345.

[19] B. Wu, Y. Wang, "Mobile robot dynamic path planning based on genetic algorithm," *Proceedings of the Annual Conference of China Institute of Communications*, vol. 1864, issue 1, pp. 96-99, 2010.

[20] J. Liu, X. Tian, W. Zhou, K. Guo, "An improved genetic algorithm for path-planning for mobile robot," *Proceedings of the 4th International Conference on Computer, Mechatronics, Control and Electronic Engineering (ICCMCEE 2015*), 2015, pp. 998-1003, https://doi.org/10.2991/iccmcee-15.2015.186.

[21] M. Honjwei, M. Longlong, "Robot path planning based on differential evolution in static environment," *International Journal of Digital Content Technology and its Applications (JDCTA)*, vol. 6, no. 20, pp. 122-129, 2012, https://doi.org/10.4156/jdcta.vol6.issue20.14.

OLEKSANDR TSYMBAL *is graduated from Kharkiv State University, a Professor of Kharkiv National University of Radio Electronics. Areas of interest: robotics, artificial intelligence, decision-making systems.*

ARTEM BRONNIKOV *is graduated from Kharkiv National University of Radio Electronics, Senior Lecturer in this university. Areas of interest: robotics, artificial intelligence, computer vision.*

GENNADIY GUDKOV *is graduated from Kharkiv National University of Radio Electronics; software engineer; areas of interest: robotics, artificial intelligence.*