

# Optimal Path Planning for Mobile Robot Using Tailored Genetic Algorithm

Fei Liu, Shan Liang\*, and Xiaodong Xian

College of Automation, Chongqing University, Chongqing, China

No. 174 Shazhengjie, Shapingba, Chongqing, 400044, China, 86-23-65112759

\*Corresponding author, e-mail:lightsun@cqu.edu.cn

## Abstract

*During routine inspecting, mobile robot may be requested to visit multiple locations to execute special tasks occasionally. This study aims at optimal path planning for multiple goals visiting task based on tailored genetic algorithm. The proposed algorithm will generate an optimal path that has the least idle time, which is proven to be effective on evaluating a path in our previous work. In proposed algorithm, customized chromosome representing a path and genetic operators including repair and cut are developed and implemented. Afterwards, simulations are carried out to verify the effectiveness and applicability. Finally, analysis of simulation results is conducted and future work is addressed.*

**Keywords:** Mobile Robot, Optimal Path Planning, Multiple Goals Visiting, Genetic Algorithm

**Copyright © 2014 Institute of Advanced Engineering and Science. All rights reserved.**

## 1. Introduction

Mobile robots have been developed for many practical tasks such as automatic patrolling in a transformer substation [1, 2], welding automatically in a production line [3] and guiding in a campus [4]. For all the applications, optimal path planning plays an important role in navigating robot to execute missions [5]. In recent years, the important issue of optimal path planning has attracted enormous attentions.

Generally, path planning is to find a suitable collision-free path from a start point to a designated goal [6]. However, in different applications, there are other four cases of path planning according to the number of robots, start points and goals: (i) One robot starts from a point, and chooses a goal from multiple candidate goals to move to [1]; (ii) One robot moves from a start point and arrives at a destination while during this course, it must visit parts of the specified goals [7]; (iii) Multiple mobile robots leave from the same start point and go towards the same goal [8]; (iv) Multiple robots start from different initial points and move to different goals [9]. In this study, we will solve the problem that a robot starts from a point and traverses the specified goals. Compared with similar research on this problem, a unique speciality is that no priority and order are made for goals to be visited.

The optimal path planning task can be described as an optimization problem in which a single objective or multiple objectives are employed. Among researches on optimal path planning, mainly path length is used for evaluating a path [10]. However, when various features of outdoor environment are considered such as friction and gravity, other criteria including energy consumption, moving time are proposed for determining an optimal path [11, 12]. In previous study on optimal path planning, by considering road attributes including length, road grade, surface roughness and the set of speed control hump, we proposed the decision factor—idle time (non-working time) as the cost of a path, which is proven to be more comprehensive on evaluating a path [1]. In this study, idle time is employed to evaluate the path.

Many researches have concerned on optimal path planning and many conventional techniques including potential field method, visibility graph and Voronoi roadmap are used [13]. Recently, various kinds of artificial intelligence methods like genetic algorithms, neural networks, fuzzy logic method, particle swarm optimization and ant colony optimization have been proposed for optimal path planning [14]. In this work, Genetic algorithm (GA) is adopted and tailored to solve concrete problem. GA is an evolutionary optimization method and is proven to perform well in optimal path planning [15-17]. To use GA, one should first find a pattern to express the feasible solutions, which is called chromosome. Besides, it is necessary to create a

fitness function to evaluate each solution. The most challenging work is developing some appropriate genetic operators acting on the population of each generation that is the set of solutions. After evolving by certain generations, the optimal one will be determined by a criterion. In many occasions, researchers use fixed-length chromosome to represent a path [18]. While in other circumstances, chromosomes with variable length are used [19, 20]. Meanwhile, different forms of fitness functions are created due to the fact that different objectives should be considered in respective application, such as path length [10] and time [11]. The key for evolution are genetic operators. Traditionally, three operators, i.e., *selection*, *crossover* and *mutation* are utilized almost in all applications [20]. Apart from them, customized genetic operators are often established according to different purposes. For example, to make a feasible solution better, operator *improvement* is designed, which will randomly choose a node, and search in neighboring grids of the node, and move it to a better location [19]. Various customized operators have enlarged the field of application of GA-based method vastly.

In this paper, for the multiple goals visiting task, we proposed a tailored genetic algorithm to find an optimal path. This section has summarized related work and introduced our research. The remainder of this paper is organized as follows: in section 2, we will state the problem including the model of work environment, the multiple goals visiting task and properties of a path. In section 3, tailored genetic algorithm is described in detail. Then, simulations and analysis of results are conducted in section 4. Finally, conclusion and future work are addressed.

## 2. Problem Formation

### 2.1. Work Environment

A graph-based topological map is used to describe the work environment [1], which is illustrated in figure 1.

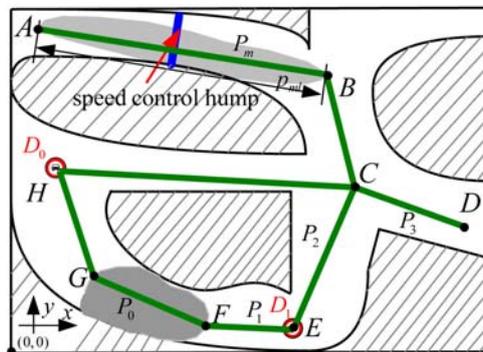


Figure 1. Model of work environment

Let  $V$  be the environmental space, which includes three parts, namely:

$P$ : The set of path segments. In figure 1,  $P_m$  (i.e.,  $m = 0, 1$ ) represents a path segment and for each one, four attributes are considered, i.e., path length  $p_{ml}$ , surface roughness  $p_{mr}$ , road grade  $p_{mg}$  and the set of speed control hump  $p_{mh}$ . Especially, the segment with shadow implies that it is a rough segment.

$N$ : The set of nodes connecting path segments. It involves four types of nodes, i.e., general node, start point, goal and charging station. For example,  $A$  to  $H$  are general nodes connecting two or more path segments respectively. We use  $D_h$  (i.e.  $h = 0, 1$ ) to represent charging stations placed in the environment. In a concrete mission, if any one node is designated to visit, it is called a goal, and the position where the robot starts from is called the start point. Since each segment has two nodes, a segment can be also represented by nodes.

For example, in figure 1, segment  $P_0$  can be also represented as  $P_{GF}$  if in one path the robot moves from node  $G$  to  $F$ , or  $P_{FG}$  from  $F$  to  $G$ .

$O$ : The set of obstacle regions. In figure 1, the regions with oblique lines represent obstacle areas.

Thus we can describe the environment as  $V = \{P, O, N\}$ .

In previous work [1], we have elaborated the cost of a path segment. The cost that the robot will pay for passing each segment includes two parts: energy  $c_e$  and the influence of vibration on robot body  $c_b$ . We use  $C(c_e, c_b)$  to describe the cost of each path segment. Furthermore, the calculation of  $c_e$  and  $c_b$  is

$$c_e = \hat{c}_e + c_s = (r_e + r_s)p_{ml} = (\Delta r_{e-r} p_{mr} \cos(p_{mg}) + \Delta r_{e-g} \sin(p_{mg}) + r_s)p_{ml} \quad (1)$$

$$c_b = (r_{b-\mu_f} / \mu_f) \eta p_{mr} p_{ml} + p_{mh} r_{b-h} \quad (2)$$

where  $\hat{c}_e$  is the energy used for moving and  $c_s$  is the energy that consumed by sensors on robot. By using  $c_e$  and  $c_b$ , the idle time is computed as

$$T_{IDLE} = f(c_e, c_b) = c_e / v_{charge} + c_b T_{MTTR} \quad (3)$$

where  $v_{charge}$  is the charging speed and  $T_{MTTR}$  is used to describe the Mean Time To Repair (MTTR) of the robot. Details of derivation of these formulations and descriptions of other parameters are not expected to shown in detail in this paper since they are available in [1].

## 2.2. Path Plannign for Multiple Goals Visiting (PPMGV)

Normally when performing regular inspecting task, the robot moves in accordance with predefined route in the environment. Occasionally, the robot may be asked to go to multiple goals to execute particular missions. For example, in Fig. 2, when the robot is at location  $S$ , it is commanded to visit  $G_1$ ,  $G_2$  and  $G_3$  temporarily. The robot can select the path coloured in blue to visit all the goals. Thus, the sequence of goals visited is  $G_1 \otimes G_2 \otimes G_3$ , for which we use  $\Gamma_1 = \{S, A, G_1, G_2, D, G_3\}$  to describe the path. However, the robot may choose visiting  $G_3$  before  $G_2$ , then the sequence becomes  $G_1 \otimes G_3 \otimes G_2$ , and subsequently we get another feasible path  $\Gamma_2 = \{S, G_1, G_3, G_2\}$  that is coloured in red. The optimal path planning for multiple goals visiting is to find the optimal one in all the accessible paths.

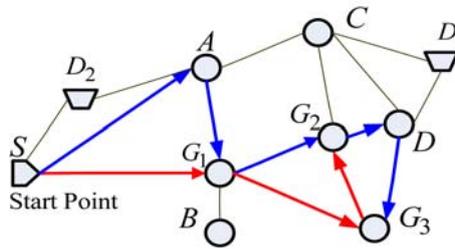


Figure 2. Task of multiple goals visiting

## 2.2. Properties of A Path

In this work, the path is represented by nodes, and four properties of each path are obtained:

(1) A path is constituted of part of the nodes. For example, the path colored in blue in Figure 2 can be described as  $\Gamma_1 = \{S, A, G_1, G_2, D, G_3\}$ . This path is constituted by nodes  $S, A, G_1, G_2, D$  and  $G_3$  in which  $G_1, G_2$  and  $G_3$  are the goals assigned.

(2) There is no priority or constraint for the sequence of goals to be visited. For example, in Figure 2, both paths  $\Gamma_1 = \{S, A, G_1, G_2, D, G_3\}$  and  $\Gamma_2 = \{S, G_1, G_3, G_2\}$  are valid for visiting goals  $G_1, G_2$  and  $G_3$ .

(3) It is permissible for a node appearing in the sequence more than once. For instance, in Figure 3, one available path is  $\Gamma = \{S, G_1, S, G_2, G_4, G_2, G_3\}$ , where  $S$  and  $G_2$  both appeared twice. The purpose of the first arrival at one goal is to perform task, and that of the other times are for going to other goals.

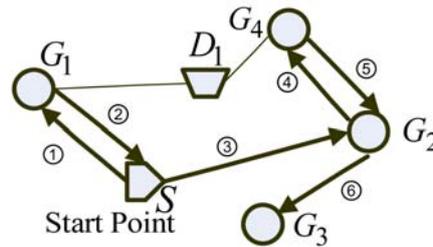


Figure 3. A special situation

### 3. Proposed Tailored Genetic Algorithm

Based on traditional genetic algorithm, modifications are made to fit our problem. We use the combination of nodes to represent the chromosome. We use idle time to evaluate a path. Except the basic three operators, i.e., *Selection*, *Crossover* and *Mutation*, we create two operators: *Repair* and *Cut*.

#### 3.1. Chromosome

The proposed tailored genetic algorithm uses the combination of nodes for path representation. An example of path encoding is shown in Figure 4, which is  $S - A - G_1 - G_2 - D - G_3$ . In this chromosome,  $S$  is the start point,  $A$  and  $D$  are general nodes, and  $G_1, G_2$  and  $G_3$  are three goals.

Two different chromosomes may have different length. For example, the length of the chromosome shown in figure 4 is 6, in which 3 goals are involved. While the length of the chromosome in figure 2,  $S - G_1 - G_3 - G_2$ , is 4, and the same 3 goals are included.

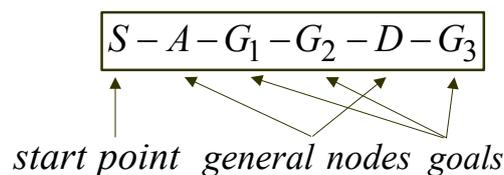


Figure 4. An example of chromosome

#### 3.2. Evaluation of Path

Chromosomes are selected for reproduction through genetic operators based on the fitness function, so it is important to establish a set of criteria to evaluate the quality of a path. For each chromosome, we adopt  $F_{T_{IDLE}}$  to evaluate it, where  $F_{T_{IDLE}}$  indicates the idle time induced by this path. The total energy consumption of a path is the sum of that of each path segment, so

$$F_{T_{IDLE}} = \sum_{i=1}^H T_{IDLE}(P_i) \quad (4)$$

where  $H$  is the number of segments and  $T_{IDLE}(i)$  is the idle time of the  $i$ th segment  $P_i$ . For example, for the individual  $S - A - G_1 - G_2 - D - G_3$  shown in figure 4, we have

$$F_{T_{IDLE}} = T_{IDLE}(P_{SA}) + T_{IDLE}(P_{AG_1}) + T_{IDLE}(P_{G_1G_2}) + T_{IDLE}(P_{G_2D}) + T_{IDLE}(P_{DG_3}) \quad (5)$$

### 3.3. Genetic Operators

In proposed genetic algorithm, except the three basic operators, i.e., *selection*, *crossover* and *mutation*, we create two other operators, i.e., *repair* and *cut*.

**(1) Selection.** The selection operation includes two steps. First, by using the strategy “elitism”, the best chromosome is found out and kept in the population of the next generation. This mechanism is helpful for finding the global optimal solution. The selection process is based on the fitness value. The best one that has the minimal  $F_{T_{IDLE}}$  will be selected to remain in the next generation. This strategy can guarantee that the best one up to now will not be destroyed by other genetic operations and can accelerate the convergence of the algorithm.

**(2) Crossover.** Crossover is an efficient way to add diversity to the population. Firstly, a crossover probability is predefined. In this operation, two parents are selected randomly and a position is selected randomly too. Then, a random probability is generated. If the probability value is less than the predefined value, the operation will go on. Otherwise, the two parents are passed to the next generation directly. The operation will end until certain times of crossing operations are carried out.

The following is an example of *crossover* operation. First, two parents are selected:

Parent 1:  $S - G_1 - G_3 - G_2$

Parent 2:  $S - A - G_1 - G_2 - D - G_3$

If point  $G_1$  is selected as the position for exchanging, then we get the offspring after crossing:

Child 1:  $S - G_1 - G_2 - D - G_3$

Child 2:  $S - A - G_1 - G_3 - G_2$

After crossing, the two children are put into the population of next generation.

**(3) Mutation.** In *mutation* operation, a position is randomly chosen and the node at this position is replaced with a different node. *Mutation* is served as a key role to diversify the solution population. Therefore, it is not necessary that a solution is better after mutating. After mutating, this node may not be connected directly with the two nodes before and after. For example, if node  $A$  in path  $S - A - G_1 - G_2 - D - G_3$  shown in Figure 2 is chosen to mutate, and changes to  $C$ , then, this individual becomes  $S - C - G_1 - G_2 - D - G_3$ . However, as seen in Figure 2, nodes  $S$  and  $C$ , and  $C$  and  $G_1$  are not connected directly, which is to say, the individual after mutation is not a feasible solution. Even so, it has made the population diversified, and the following operator *repair* can make it feasible.

**(4) Repair.** When executing genetic operators, some infeasible paths may appear. For instance, after mutation, individual  $S - A - G_1 - G_2 - D - G_3$  becomes  $S - C - G_1 - G_2 - D - G_3$ , while nodes  $S$  and  $C$ , and  $C$  and  $G_1$  are not connected directly. When this happens, we will use *repair* operator to solve this problem. The practical way is inserting some suitable nodes between the two nodes.

Take string  $S - C - G_1 - G_2 - D - G_3$  as an example. We first check if this individual is feasible by examining every two adjacent nodes. If at a position, the node and the next node are not connected directly, then, this operator will try to add some nodes between them in order to make the two connected reasonably. In the above example, the nodes  $S$  and  $C$  may be inserted by node  $A$ , and then  $C$  and  $G_1$  may be inserted by nodes  $G_2$  or  $A$ , which is decided randomly. If  $A$  is selected, then the individual is repaired to be  $S - A - C - A - G_1 - G_2 - D - G_3$ , and if  $G_2$  is selected, it will become  $S - A - C - G_2 - G_1 - G_2 - D - G_3$ . No matter whichever is chosen, the result is that the path becomes feasible at last.

**(5) Cut.** In a chromosome, it is admissible that any node appears more than one time. But the unnecessary reduplication must be avoided. For example, in the string  $S - A - C - A - G_1 - G_2 - D - G_3$  obtained after repairing, node  $A$  appears twice and between them there is no goal. It can be regarded as that between the two times arriving at  $A$ , the intention is not for going to any goal. So, the sequence  $C - A$  is meaningless and it needs to be

cut. Finally, this string becomes  $S- A- G_1- G_2- D- G_3$ . Therefore, the *cut* operator is to do such things that cutting the unmeaning sequences existing in each individual. However, the reduplication does not include the situation that a goal exists between the same two nodes. For instance, in chromosome  $S- A- C- G_2- G_1- G_2- D- G_3$ ,  $G_2$  appears twice. But between them there is another goal  $G_1$  which indicates that the purpose of arriving at  $G_2$  for the second time is for visiting another goal. Thus, the second time passing  $G_2$  is meaningful.

#### 4. Simulation Studies

##### 4.1. Simulations and Results

We use the topological map shown in Figure 5 in simulations, which is built in previous work [1]. There are 23 path segments and 17 nodes in the environment. In addition, the attributes of each segment are also listed out in [1]. In simulations, parameters in the proposed genetic algorithm are set as follows: the population size is 30, and the maximum evolution generation is set to be 100, crossover rate  $P_c = 0.9$  and mutation rate  $P_m = 0.001$ .

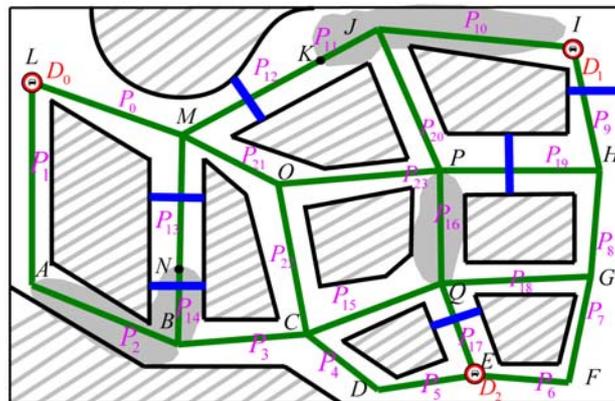


Figure 5. Topological map of environment

**(1) Simulation I.** In this simulation, node  $A$  is set as the start point, and the goals are  $N$ ,  $O$  and  $Q$ . We list out the detailed value of idle time of the best one in each generation in table 1 and show them in Figure 6. It is obtained from the result that the optimal solution comes out in the 23<sup>rd</sup> generation. The optimal path is  $A- B- N- M- O- C- Q$  and its idle time  $F_{T_{IDLE}} = 855.075562s$ . The order of visiting is  $N$ ,  $O$  and  $Q$ . The computational time is measured to be 145ms.

Table 1. Details of Best Individuals in Each Generation

Generation	Best individual	$F_{T_{IDLE}}$ (s)
1-4	$A- B- C- O- C- Q- C- O- M- N$	1293.958740
5-8	$A- L- M- O- C- D- E- Q- C- B- N$	1283.412720
9-13	$A- B- C- O- M- N- B- C- Q$	1092.191162
14-16	$A- B- N- M- O- C- D- E- Q$	1025.964884
17-20	$A- B- N- B- C- Q- C- O$	946.653320
21-31	$A- B- N- M- O- P- Q$	884.662598
32-100	$A- B- N- M- O- C- Q$	855.075562

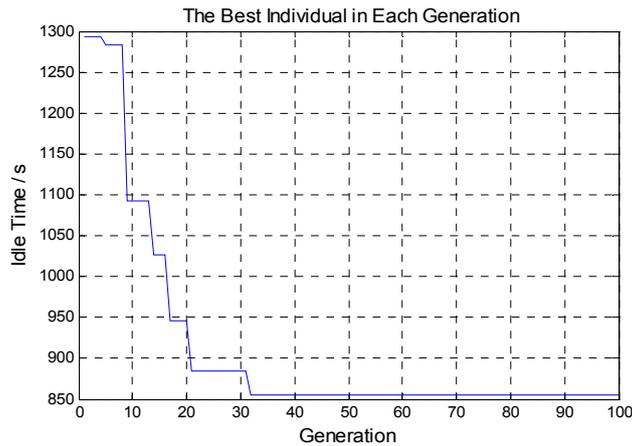


Figure 6. Results of simulation I

(2) **Simulation II.** In this test,  $A$  is the initial point, and goals are  $J, O, P$  and  $Q$ . We conduct the computation twice, and the results are shown as case 1 and case 2 in Figure 7. In case 1, the optimal solution appears in the 86<sup>th</sup> generation, and in case 2 it comes out in the 98<sup>th</sup> generation. At last we get the same optimal path is  $A - B - C - Q - C - O - P - J$  and its idle time is 1043.175903s. Thus the goals are visited in the order of  $Q, O, P$  and  $J$ . The computational time is 157 ms and 153ms respectively.

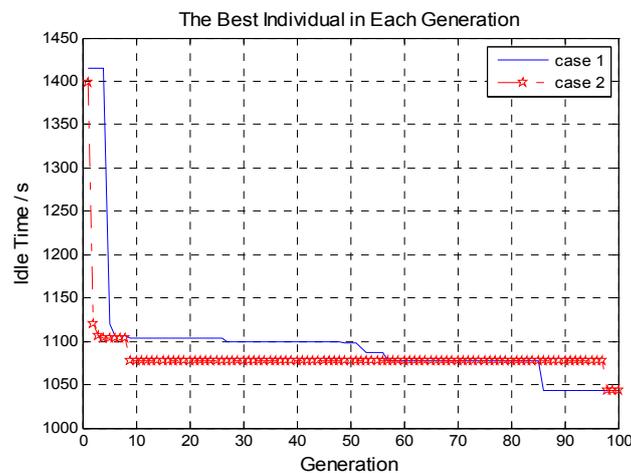


Figure 7. Results of simulation II

#### 4.2. Discussion of Simulation Results

In the two simulations, we implement our proposed tailored genetic algorithm to find the optimal path for multi-goal visiting task and finally optimal solutions are obtained. In the following we will discuss about the similarity and differences between each case and evaluate the proposed genetic algorithm based on simulation results.

(1) As the genetic algorithm is a kind of stochastic, evolutionary search method, the optimal solution obtained at the end may be not the global optimal one truly, but converges to.

(2) In the two cases above, the speed of converging to the optimal solution is different. For example, the optimal one appears in the 32<sup>rd</sup> generation in simulation I, while it is obtained in the 86<sup>th</sup> and 98<sup>th</sup> in two cases in simulation II.

(3) The computational time in two simulations are 145ms, 157ms and 153ms respectively, which shows great efficiency in computation. However, there are only 23 path segments with 17 nodes in the environment. To verify the timeliness and efficiency, a more complicated environment needs to be constructed and more simulations are required.

(4) When using GA method, the stop condition can be either that the best solution keeps unvaried for certain generations, or that the current maximum generation is exceeded [21]. In proposed genetic algorithm, the latter one is adopted. However, in reality, both can not ensure that the final solution is truly the optimal one, and therefore it is uncertain that which one is better absolutely. For instance, in case 2 of simulation II, the solution generated in the 9<sup>th</sup> keeps the best in the following 89 generations. If we use the former criterion, and set the maximum generation to be 80, it will be regarded as the final optimal path. However, it is soon replaced by a better solution. For the latter criterion, if we set the maximum generation as 90, then in case 2 we can not get the optimal one that is obtained in case 1.

## 5. Conclusion

We have proposed a novel tailored genetic algorithm to plan an optimal path for the multi-goal visiting task. According to the particularity of the problem, special form of chromosome is used to represent the path and customized genetic operators are developed. The effectiveness of the method is verified by simulations. Furthermore, through analysis of simulation results, evaluation on our proposed method is addressed, which is useful for wider implementation in various circumstances. Further, we will create more complicated environments to verify and modify the algorithm. In addition, the comparison with other research is considered as an important work to be conducted. Moreover, we will consider the situation that the robot has limited energy, and therefore, both it and idle time should be employed to evaluate the path.

## References

- [1] Liu F, Liang S, Xian XD. Determination of An Optimal Return-path on Road Attributes for Mobile Robot Recharging. *IJARS*. 2011; 8(5): 83-92.
- [2] Wang KZ, Liang S, Bi HB, Xian XD. Implementation of a robot inspection system for substation equipment based on pioneer 3-at. *ICIC Express Letters*. 2010; 4(5): 1-6.
- [3] Madsen O, Sørensen CB, Larsen RL, Overgaard L, Jacobsen NJ. A system for complex robotic welding. *Industrial Robot: An International Journal*. 2002; 29(2): 127-131.
- [4] Thrapp R, Westbrook C, Subramanian D. *Robust localization algorithms for an autonomous campus tour guide*. Proceedings of IEEE International Conference on Robotics and Automation. 2001: 2065-2071.
- [5] Raja P, Pugazhenth S. Optimal Path Planning of Mobile Robots: A Review. *International Journal of Physical Sciences*. 2012; 7(9): 1314-1320.
- [6] Yuan M, Wang SA, Wu C, Chen N. A Novel Immune Network Strategy for Robot Path Planning in Complicated Environments. *Journal of Intelligent & Robotic Systems*. 2010; 60(1): 111-131.
- [7] Shibata T, Fukuda T. *Intelligent Motion Planning by Genetic Algorithm with Fuzzy Critic*. International Symposium on Intelligent Control. 1993: 5-10.
- [8] Yang SX, Hu Y. *A Knowledge Based GA for Path Planning of Multiple Mobile Robots in Dynamic Environments*. IEEE International Conference on Robotics and Automation. 2007: 71-76.
- [9] Mansouri M, Shoorehdeli MA, Teshnehlab M. *Path Planning of Mobile Robot Using Integer GA with Considering Terrain Conditions*. IEEE International Conference on Systems, Man and Cybernetics. 2008: 208-213.
- [10] Hellstrom T, Ringdahl O. Real-time path planning using a simulator-in-the-loop. *International Journal of Vehicle Autonomous Systems*. 2009; 7(1/2): 56-72.
- [11] Wang HF, Yang YZ. Time-optimal Trajectories for a Car-like Robot. *Automatica*. 2008; 34(4): 445-452.
- [12] Sun Z, Reif JH. On Finding Energy-Minimizing Paths on Terrains. *IEEE Transaction on Robotics*. 2005; 21(1): 102-114.
- [13] Garcia MAP, Montiel O, Castillo O, Sepúlveda R, Melin P. Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation. *Applied Soft Computing*. 2009; 9(3): 1102-1110.
- [14] Ahmed F, Deb K. Multi-objective optimal path planning using elitist non-dominated sorting genetic algorithms. *Soft Computing*. 2012:1-17.
- [15] Wang YH, Chi N. Path planning optimization for teaching and playback welding robot. *Telkomnika*. 2013; 11(2): 960-968.

- [16] Konak A, Coit DW, Smith AE. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering*. 2006; 91(9): 992-1007.
- [17] Yan XS, Wu QH, Liu HM. An improved robot path planning algorithm. *Telkomnika*. 2012; 10(8): 1948-1955.
- [18] AL-Taharwa I, Sheta A, Al-Weshah M. A Mobile Robot Path Planning Using Genetic Algorithm in Static Environment. *Journal of Computer Science*. 2008; 4(4): 341-344.
- [19] Yang SX, Hu Y, Meng MQ. *A Knowledge Based GA for Path Planning of Multiple Mobile Robots in Dynamic Environments*. IEEE International Conference on Robotics and Automation. 2007: 71-76.
- [20] Mansouri M, Shoorehdeli MA, Teshnehlab M. *Integer GA for Mobile Robot Path Planning with using another GA as repairing function*. Proceedings of the IEEE International Conference on Automation and Logistics. 2008:135-140.
- [21] Zitzler E, Laumanns M, Bleuler S. A Tutorial on Evolutionary Multi-objective Optimization. *Evolutionary Computation*. 2004; 535(5): 3-37.