Optimal Path Planning for Mobile Robot Using Tailored Genetic Algorithm

Fei Liu, Shan Liang*, Xiaodong Xian
Lab of Intelligent Sensing and Control
College of Automation, Chongqing University
Chongqing, China
lightsun@cqu.edu.cn

***Abstract*—During routine inspecting, mobile robot may be requested to leave for certain locations to do special tasks sometimes. This study aims at optimal path planning for this multiple goals visiting task based on tailored genetic algorithm. We adopt idle time (non-working time) as the decision factor for evaluating the path [1, 2]. The proposed algorithm will generate an optimal path that has the least idle time. In the 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

I. INTRODUCTION

Mobile robots have been developed for many real-life tasks such as automatic patrolling in a transformer substation [3], welding automatically in a production line [4] and guiding in a campus [5]. For all the applications, path planning plays an important role in navigating robot to execute missions [6-7]. Further, it is generally occurred that more than one accessible path can be found, which indicates that a strategy is necessary for selecting the optimal or near-optimal path. In recent years, the important issue of optimal path planning has attracted considerable attentions.

Types of Path Planning

Generally, path planning is to find a suitable collision-free path for a robot moving from a start point to a fixed target location [8-9]. In this situation, there have just one start location and one goal. However, in different applications, there are other four types of path planning according to the number of start points and goals: (i) one robot starts from a point, and chooses a goal from multiple candidate goals to move to. For example, when needing to recharge, the robot should select a docking station from all the stations to go 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, for example, to pick up loads and at last carry them to the target location [10]; (iii) multiple mobile robots leave from the same start point and go towards the same goal [11]; (iv) multiple robots start from different initial points and move to different goals [12, 13]. In this study, we will solve the problem that a robot sets off from a point and traverses the specified goals. Compared with other researches, it has two properties: (i) none is designated as the ultimate goal; (ii) no order is set for visiting the goals.

Objectives for Optimal Path Determination

Among researches about optimal path planning, mainly path length is used for evaluating a path [14, 15]. However, when various features of outdoor environment are considered such as friction and gravity, other criteria are proposed for determining an optimal path. For example, Wang et al. intend to plan a time-optimal trajectory for the mobile robot [16]. When finding optimal paths on terrains for a mobile robot, Sun et al. use energy consumed due to friction and gravity as the cost of a path [17]. In [18] researchers considered the energy expended on rotating, since in the environment with many walls and corners, it may cause much energy consumption if rotating frequently. For planning an optimal path to multiple goals, Lobaton et al. took retracing into consideration [19]. In a particular application, the robot is required to pick up loads on the way, so the optimal way is that costing less time and collecting more loads [10].

In previous research on optimal path planning, on consideration of road attributes including length, road grade, surface roughness and the set of speed hump, we have studied optimal path planning based on energy consumption [20]. Further, by taking into consideration influence of vibration on mobile robot induced by motion, 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, 2]. In this study, idle time is employed to evaluate the path.

GA-based Path Planning

Genetic algorithm (GA), based on the mechanism of natural selection and natural genetics, was first developed in the 1970s by Holland [21]. It is an evolutionary optimization method and is proven to perform well in optimal path planning [22]. 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 part 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.

For diverse applications, due to the differentiation of problems, various modifications are made based on basic GA to solve concrete problems. In many occasions, researches use fixed-

Sponsors: the Key Project of Science and Technology Committee of Chongqing (CSTC, 2009AB2139)

length chromosome to represent a path [23, 24]. While in other circumstances, variable length chromosomes are used. For example, in a grid-based environment, authors use string of cells to describe a path whose length is unfixed [25-27]. 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 [14], energy consumption [28], time consumption [16], smoothness and safety [29]. The key for evolution is the genetic operators. Traditionally, two operators, i.e., *crossover* and *mutation* are used nearly in all applications [26]. They play significant pole in adding diversity to the population and therefore are in favor of finding the global optimal solution. 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 [25]. In [27], ***deletion*** is employed to eliminate duplicate nodes existed in a path and adjust an individual in order of search direction. 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.

II. PROBLEM FORMATION

In this section, we expand on the problem including the model of work environment and the task of multiple goals visiting. At last we introduce the properties of a path that are very important for the proposed genetic algorithm.

2.1 Model of Work Environment

We use a graph-based topological map to describe the work environment [1, 2], which is illustrated in Fig. 1. In the map, $P_m (i.e.\ m = 0,1)$ represents a path segment, and A to H are nodes connecting two or more path segments respectively. For each segment, 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} . Besides, there have charging stations placed in the environment, each of which is named as $D_h (i.e.\ h = 0,1)$. For example, in Fig.1, D_0 is a charging station.

In previous study [1, 2], we have elaborated the cost of a path segment. The cost that the robot will pay for passing each segment includes two parts: energy consumption 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 segment. Furthermore, the calculation of c_e and c_b is

$$\begin{aligned} c_e &= \hat{c}_e + c_s = (r_e + r_s) p_{ml} \\ &= (\Delta r_{e\_r}\, p_{mr} \cos(p_{mg}) + \Delta r_{e\_g} \sin(p_{mg}) + r_s) p_{ml} \end{aligned} \tag{1}$$

$$c_b = (r_{b\_\mu_f} / \mu_f)\eta p_{mr} p_{ml} + p_{mh} r_{b\_h} \tag{2}$$

where $\hat{c}_e$ is the energy used for driving the motor 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

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

where v_{charge} is the charging speed and T_{MTTR} is used to describe the Mean Time To Repair (MTTR) of our 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 previous work [1, 2].

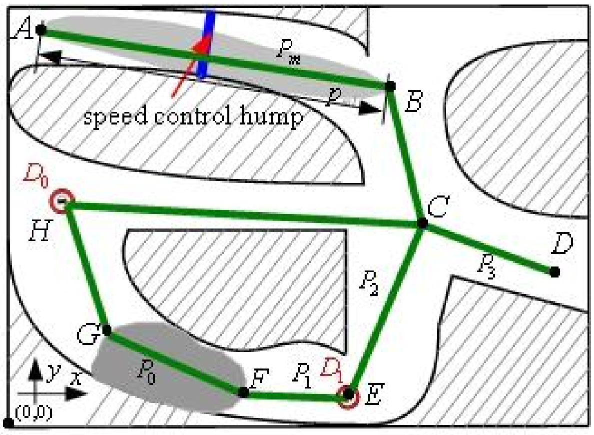

Figure 1. Model of work environment

2.2 Task of Multiple Goals Visiting

Normally when executing regular inspecting task, the robot traverses in accordance with predefined route in the environment. Occasionally, the robot may be commanded to go to certain positions to perform particular mission. At this moment, it should stop and plan an optimal feasible path to visit the goals one by one to accomplish the task. For example, in Fig. 2, when the robot is at point S , it is asked to visit G_1, G_2 and G_3 temporarily.

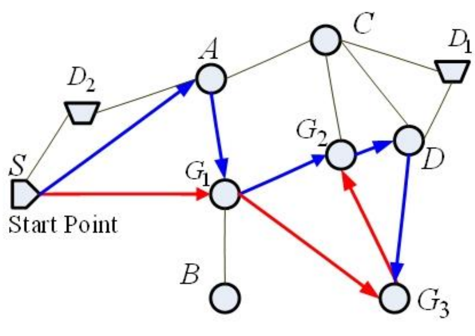

Figure 2. Task of multiple goals visiting.

The robot can choose any goal to visit first, which implies no order is set for the visiting task. Take the situation in Fig. 2 as an example. The robot can select the path colored in blue to visit all the goals. Thus, the sequence of goals visited is

$G_1 \to G_2 \to G_3$, for which we use $\Gamma_1 = \{S, A, G_1, G_2, D, G_3\}$ to describe the whole path for completing the task. However, the robot may choose visiting G_3 before G_2, then the sequence becomes $G_1 \to G_3 \to G_2$, and subsequently we get another feasible path $\Gamma_2 = \{S, G_1, G_3, G_2\}$ that is colored in red in Fig. 2.

2.3 Properties of a Path

Several properties of a path are stated in this subsection. In the topological map used to describe the environment, the road network is constituted by connected nodes, which is illustrated in Fig. 1. Therefore, we use the combination of nodes to represent a path. In this research, four properties of a path are obtained:

(a) A path is constituted of parts of the nodes. For example, the path colored in blue in Fig. 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.

(b) There is no priority or constraint for the sequence of goals to be visited. The ultimate purpose is to visit as more as possible goals. So, whichever is visited first is permitted. For example, in Fig. 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 admissible.

(c) It is permissible for a node appearing in the sequence more than once. For instance, in Fig. 3, one available path is $\Gamma = \{S, G_1, S, G_2, G_4, G_2, G_3\}$, where S and G_2 both are visited 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.

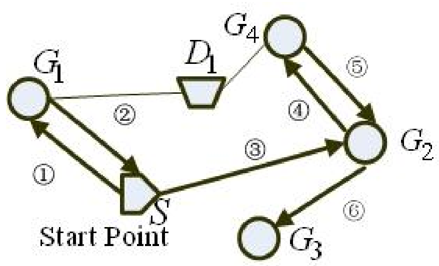

Figure 3. A special situation

III. PROPOSED GENETIC ALGORITHM FOR PATH PLANNING

Based on traditional genetic algorithm, modifications are made to fit our problem. We use the combination of nodes to represent the chromosome. The fitness functions include two parts which are used to calculate energy consumption and idle time respectively. Except the basic three operators, i.e., *selection*, *crossover* and *mutation*, we create two operators: *repair* and *cut*.

3.1 Chromosome

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

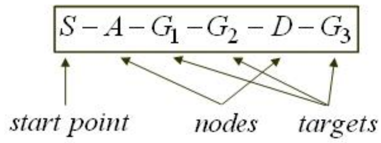

Figure 4. An example of chromosome in tailored genetic algorithm.

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

3.2 Evaluation

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 individual, 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

$$F_{T_{IDLE}} = \sum_{i=1}^{N} T_{IDLE}(i) \tag{4}$$

where N is the number of segments and T_{IDLE}(i) is the idle time of the ith segment.

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 will select the best individual from the population in each generation and keep it in the next generation. The selection is based on the fitness value. Here, some special characteristics of this operator are emphasized in the following.

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.

When executing *crossover* operation, a crossover point will be generated. Since the length of two parents may not be the same, the sequence number of the point will not be bigger than the length of the shorter one. Then, in the other parent, we find the corresponding node and its sequence number of the first appearance. If the other one has the same node, exchange the latter parts of the two parents. If not, quit and restart from choosing parents.

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 Fig. 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 Fig. 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 $S-C-G_1-G_2-D-G_3$. 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. When executing *repair* operation, we first check if this individual is feasible by examine 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 has 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 $S-A-C-G_2-G_1-G_2-D-G_3$, goal 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.

IV. SIMULATIONS AND RESULTS

In this section, simulations are implemented to examine our proposed tailored genetic algorithm. On top of this, with the simulation results, analysis and discussion are addressed in detail.

4.1 Simulations and results

We still use the topological map (noted as C_1) shown in Fig. 5 in simulations, which is built in previous work [1, 2]. In addition, the attributes of each segment are also listed in [1].

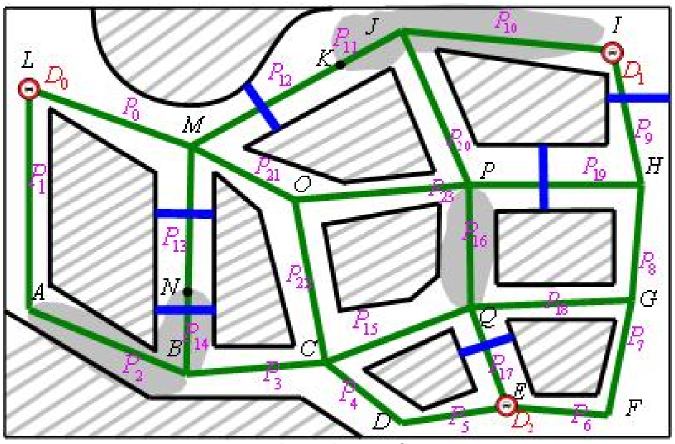

Figure 5. Topological map of environment.

In simulations, parameters in the proposed genetic algorithm are set as follows: *POPULATION_SIZE*=30, and *GENERATION_NUM*=100. Crossover rate is $P_c = 0.9$ and Mutation rate $P_m = 0.001$.

(1) Simulation I

In this test, node A is set as the start point, and the goals are C, H and M. We list out the concrete value of idle time of the best one in each generation in table 1 and show them in Fig. 6. It is obtained from the result that the optimal solution comes out in the 18th generation. The optimal path is $A-B-C-O-M-O-P-H$. Its idle time $F_{T_{IDLE}} = 1022.1330s$ and the order of visiting is C, M, H.

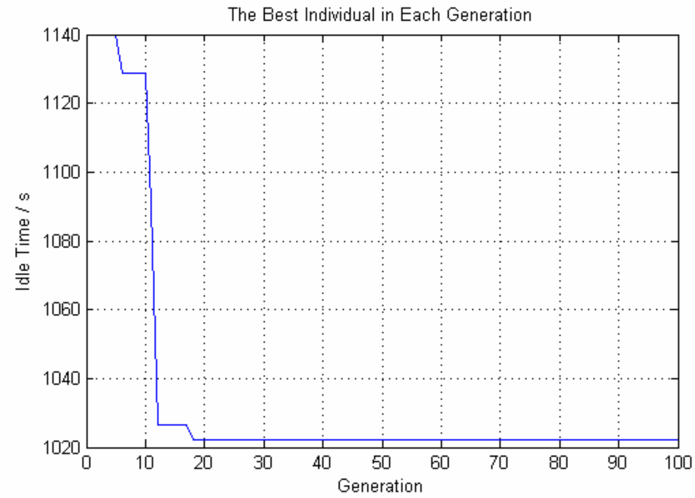

Figure 6. Result of simulation I.

TABLE I. DETAILS OF BEST INDIVIDUALS IN EACH GENERATION

Generation	**Best individual**	$F_{T_{IDLE}}(s)$
1-5	$A-B-C-B-N-M-O-P-H$	1139.6626
6-10	$A-L-M-N-B-C-Q-G-H$	1128.5902
11	$A-L-M-O-C-O-P-H$	1090.5179
12-17	$A-L-M-O-C-Q-G-H$	1026.6969
18-100	$A-B-C-O-M-O-P-H$	1022.1323

(2) Simulation II

In this simulation, we set A as the start point, and the goals are H, N, O and Q. One result is shown in Fig. 7. In the 30th generation, we get the optimal path that is $A-B-N-M-O-C-Q-G-H$. Its idle time is $1123.5112s$. The result shows that the robot can visit the goals in the order of N, O, Q, H.

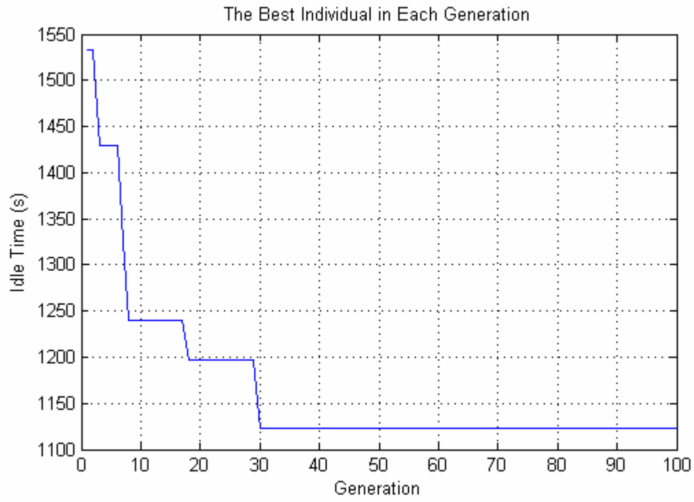

Figure 7. Result of simulation II.

(3) Simulation III

In this simulation, A is the initial point, and goals are E, H, N and O. We show the result in Fig. 8. From Fig. 8, we get the optimal solution in the 92nd generation. The optimal path is $A\text{-}B\text{-}N\text{-}M\text{-}O\text{-}P\text{-}H\text{-}G\text{-}F\text{-}E$ and its idle time is $1193.8413s$. Therefore, the robot will visit these goals in the order of N, O, H, E.

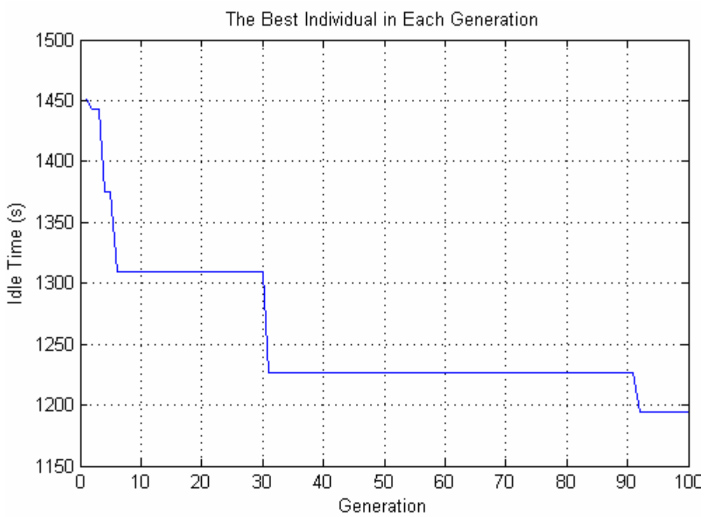

Figure 8. Result of simulation III.

4.2 *Analysis of the simulation results*

In the three simulations above, 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 itself is a kind of stochastic, evolutionary search method, the optimal solution obtained at the end may not be the global optimal one truly, but converges to.

(2) In the three cases above, the speed of converging to the optimal solution is different. For example, the optimal one appears in the 18th generation in simulation I, while it is obtained in the 30th generation in simulation II, and 92nd in simulation III.

(3) Generally, when using GA method, the stop condition can be either that the best solution keeps unvaried for certain number of generations, or that the current maximum generation is exceeded [30]. In proposed genetic algorithm, the latter is adopted. However, in reality, both can not ensure the final solution is one hundred percent the optimal one, and therefore it is uncertain that which one is better absolutely. As an example, in simulation III, the solution generated firstly in the 30th remains the best one in the following 62 generations. If we use the former stop criterion, and set the maximum generation is 50 or 60, this solution will be regarded as the final optimal path. However, it is soon replaced by a better solution. Furthermore, if we set the maximum generation is 90, we also can not get the better solution that comes out soon.

(4) In the topological map, there have just 17 path nodes. Since it is not a big number, the advantage of our method cannot be reflected on time complexity. In relevant studies on analogous problems, generally there are two kinds of solutions, which are stochastic search algorithm and exhaustive search method. The second method can guarantee that the optimal solution is sure to be found. However, when searching space grows exponentially as the nodes increases, time complexity will grow enormously too. Even in extreme cases, it seems impossible to complete in acceptable time. Under this circumstance, stochastic evolutionary search such as the proposed genetic algorithm will show great advantage because it can quickly locate high performance regions in extremely large and complex search space [10].

V. CONCLUSION

We have proposed a tailored genetic algorithm to plan an optimal path for the multi-goal visiting task. Aiming at the particularity of the problem, special form of chromosome is used to represent the path and customized genetic operators are development. 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. Future work will be carried out to consider the situation that energy is limited, and therefore, both it and idle time should be employed to evaluate the path.

ACKNOWLEDGMENT

This research was sponsored by the Key Project of Science and Technology Committee of Chongqing (CSTC, 2009AB2139).

REFERENCES

[1] F. Liu, S. Liang, and X. D. Xian, Determination of An Optimal Return-path on Road Attributes for Mobile Robot Recharging, *International Journal of Advanced Robotic Systems*, vol.8, no.5, pp.83-92, 2011.

[2] F. Liu, S. Liang, and X. D. Xian, Corrigendum to Determination of An Optimal Return-path on Road Attributes for Mobile Robot Recharging, *International Journal of Advanced Robotic Systems*, vol. 9, no.5, pp.1-3, 2012.

[3] K. Z. Wang, S. Liang, H. B. Bi, and X. D. Xian, Implementation of a robot inspection system for substation equipment based on pioneer 3-at, Science, vol. 4, no. 5, pp.1-6, 2010.

[4] O. Madsen, C. B. Sørensen, R. Larsen, L. Overgaard, and N. J. Jacobsen, A system for complex robotic welding, Industrial Robot: An International Journal, vol. 29, no. 2, pp.127-131, 2002.

[5] R. Thrapp, C. Westbrook, and D. Subramanian, Robust localization algorithms for an autonomous campus tour guide, *Proceedings of IEEE International Conference on Robotics and Automation*, 2001, pp.2065-2071.

[6] P. Raja and S. Pugazhenthi, Optimal Path Planning of Mobile Robots: A Review, *International Journal of Physical Sciences*, vol. 7, no. 9, pp.1314-1320, 2012.

[7] N. Sariff and N. Buniyamin, An Overview of Autonomous Mobile Robot Path Planning Algorithms, *4th Student Conference on Research and Development*, 2006, pp.183-188.

[8] M. Yuan, S. A. Wang, C. Wu, and N. Chen, A Novel Immune Network Strategy for Robot Path Planning in Complicated Environments, *Journal of Intelligent & Robotic Systems*, vol. 60, no. 1, pp.111-131, 2010.

[9] M. McNaughton and C. Urmson, FAHR: Focused A* Heuristic Recomputation, *International Conference on Intelligent Robots and Systems*, 2009, pp.4893-4898.

[10] T. Shibata and T. Fukuda, Intelligent Motion Planning by Genetic Algorithm with Fuzzy Critic, *International Symposium on Intelligent Control*, 1993, pp.5-10.

[11] S. X. Yang and Y. Hu, A Knowledge Based GA for Path Planning of Multiple Mobile Robots in Dynamic Environments, *IEEE International Conference on Robotics and Automation*, 2007, pp.71-76.

[12] M. Bennewitz and S. Thrun, Optimizing Schedules for Prioritized Path Planning of Multi-Robot Systems, *IEEE International Conference on Robotics and Automation*, 2001, pp.271-276.

[13] M. Mansouri, M. A. Shoorehdeli, and M. Teshnehlab, Path Planning of Mobile Robot Using Integer GA with Considering Terrain Conditions, *IEEE International Conference on Systems, Man and Cybernetics*, 2008, pp.208-213.

[14] T. Hellstrom and O. Ringdahl, Real-time path planning using a simulator-in-the-loop, I*nternational Journal of Vehicle Autonomous Systems*, vol. 7, no. 1/2, pp.56-72, 2009.

[15] R. Iraji, Robot path planning using wavefront approach with wall-following, *2nd IEEE International Conference on Computer Science and Information Technology*, IEEE, 2009, pp. 417-420.

[16] H. F. Wang. and Y. Z. Yang, Time-optimal Trajectories for a Car-like Robot, *Automatica*, vol. 34, no. 4, pp.445-452, 2008.

[17] Z. Sun and J. H. Reif, On Finding Energy-Minimizing Paths on Terrains, *IEEE Transaction on Robotics*, vol. 21, no. 1, pp.102-114, 2005.

[18] Y. Mei, Y.-hsiang Lu, Y. C. Hu, and C. S.G. Lee, Energy-Efficient Motion Planning for Mobile Robots, *International Conference on Robotics and Automation*, 2004, pp.4344–4349.

[19] E. Lobaton, J. Zhang, S. Patil, and R. Alterovitz, Planning Curvature-Constrained Paths to Multiple Goals Using Circle Sampling, *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp.1463-1469.

[20] F. Liu, S. Liang, X. D. Xian, and H. B. Bi, Optimal Path Planning for Mobile Robot in Consideration of Road Attributes, *ICIC Express Letters*, vol. 6, no. 1, pp.281-287, 2012.

[21] Holland, Adaptation in Natural and Artificial Systems, *Ann Arbor: University of Michigan Press*, 1975.

[22] A. Konak, D. W. Coit, and A.E. Smith, Multi-objective optimization using genetic algorithms: A tutorial, *Reliability Engineering*, vol. 91, no. 9, pp.992-1007 2006.

[23] I. AL-Taharwa, A. Sheta, and M. Al-Weshah, A Mobile Robot Path Planning Using Genetic Algorithm in Static Environment, *Journal of Computer Science*, vol. 4, no. 4, pp. 341-344, 2008.

[24] G. Nagib and W. Gharieb, Path Planning for a Mobile Robot Using genetic Algorithms, *Proceedings of the International Conference on Electrical, Electronic and Computer Engineering (ICEEC'04), Cairo, Egypt*, 2004, pp.185-189.

[25] S. X. Yang, Y. Hu, and M. Q. Meng, A Knowledge Based GA for Path Planning of Multiple Mobile Robots in Dynamic Environments, *IEEE*

International Conference on Robotics and Automation, 2007, pp.71-76.
[26] M. Mansouri, M. A. Shoorehdeli, and M. Teshnehlab, 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, pp.135-140.
[27] Z. Yao and L. Ma, A Static Environment-Based Path Planning Method by Using Genetic Algorithm, *International Conference on Computing, Control and Industrial Engineering,* 2010, pp.405-407.
[28] S. Liu and D. Sun, Optimal Motion Planning of a Mobile Robot with Minimum Energy Consumption, *International Conference on Advanced Intelligent Mechatronics (AIM2011)*, Budapest, Hungary, 2011, 43-48.
[29] F. Ahmed and K. Deb, Multi-Objective Path Planning using Spline Representation, *IEEE International Conference on Robotics and Biomimetics (ROBIO)* , 2011, pp.1047-1052.
[30] E. Zitzler, M. Laumanns, and S. Bleuler, A Tutorial on Evolutionary Multiobjective Optimization, *Evolutionary Computation*, vol. 535, no. 5, pp.3-37, 2004.