Probabilistic Roadmap, A*, and GA for Proposed Decoupled Multi-Robot Path Planning

Safaa H. Shwail[1]
Alia Karim[2]

[1] *College of Science, University of Babylon, Babylon, IRAQ*
safaa_shwail@yahoo.com
[2] *Division of Computer Science, University of Technology, Baghdad, IRAQ*

In this paper, two main steps algorithm used to determine the optimal path planning for the multi-robot problem. Firstly, a probabilistic roadmap is developed in the free configuration space of the robots work space which avoids the static obstacles. Then by using the A* algorithm and the probabilistic roadmap created, the near-optimal path for each robot is determined which represents the shortest path between the starting point and the end point. Secondly, we used an improved genetic algorithm to make these paths more optimal to guarantee the shortest paths, which represents an intermediate process lie between roadmap creation and path following. Also, we used a new population production method in this algorithm, and new operators in addition to the selection and crossover in the genetic algorithm also used to ensure the feasibility of the results. The results of the simulation gained from the proposed method show that it is optimal by finding the shortest path for each robot with optimization rate average 18.25% from the original path, and it is fast enough to find the optimal paths in few generations.

Keywords: Path planning; Dynamic path planning; Multi-robot; Probabilistic roadmap
Received: 25 June 2014, **Revised**: 8 August 2014, **Accepted**: 15 August 2014

1. Introduction

The path planning problem is an important topic in robotics, especially in mobile autonomous robots. Any autonomous robot need to plan it path autonomously without the help of human, this done by defining the path that it will follow from the starting point to the goal point without collide with any obstacle in the way [1]. The path planning problems are used in many applications such as mobile robots, video games, and robotic surgery [2]. The definition of the dynamic multi-robot path planning problem is: how the planner can find a path for each robot from its start position to its goal position without collide with static obstacles, movable obstacles, and other robots [1,2]. The multi-robot path planning problem can be categorized either to centralized or decoupled. In the centralized path planning, the planner deal with the robots as one composite robot in a combined workspace to find the path. On the other hand, the decoupled planner finds the path for each robot independently, and then solves any conflicts, or collisions may occur [3,4].

The path planning can be also categorized to online and offline path planning. An online path planning is the path planning that implemented in the real environment depending on the sensor information while the offline path planning is implemented in a model of the environment [5]. The workspace of the robots can be either static or dynamic. The dynamic workspace [6] contains both static and movable obstacles, while static workspace [7] contains only static obstacles. According to [2], the path planning problem solved in one of two philosophies: Sampling-Based approaches like Rapidly Exploring Random Trees RRT [8] and roadmap methods [9] or combinatorial path planning like exact cell decomposition or approximate cell decomposition methods. Genetic algorithm (GA) was firstly presented by Holland [10] in 1975. The Genetic Algorithm, which is the algorithm to simulate the natural evolution, is widely applied to optimization, adaptation and learning problems.

In this paper, two main steps will be used to find the optimal path planning for the multi-robot problem. Firstly, a probabilistic roadmap is built in the free configuration space of the robot's workspace that avoids the static obstacles. Then by using the A* algorithm and the probabilistic roadmap created, the near-optimal path for each robot is found which represents the shortest path between the robot's starting point and ending point. Secondly, we used an improved genetic algorithm to make these paths more optimal to guarantee the shortest paths, which represents a middle process located between roadmap creation and path following. The structure of the paper is as follows: Section 2 will discuss the previous work in the field that combines both path planning and genetic algorithm. Section 3 will state how the paths generated from the free part of the workspace. Section 4 will be the path optimization phase by using the GA and its operators. In Section 5, the simulation results that obtained from the software will be showed. Finally, the conclusion of the work will be given in section 6.

2. Previous Work

The main aim of the robot path planning is how to find a path to the robot that connects its start position and goal position, and when the robot moves through this path, it costs power, time, or both. For this reason, the path of the robot needs to be optimal and feasible. By feasible, it means that the path must be collision free and avoids the collision with obstacles. Moreover, by optimal it means the path must be the shortest among all available paths. For this reason, the need arises to a method for optimization, which is the Genetic Algorithm.

The previous work that combines both the path planning and the genetic algorithm will be stated here in this section. The authors of [11] adopted decoupled planning approach by independent plans for each robot's path from its start position to its goal position by using a genetic algorithm. They used the reactive strategy as a local collision avoider. In order to improve the genetic algorithm, they used based-knowledge operators, which indicated an increase in the performance of the algorithm to satisfy the real-time re-planning demand. An adaptive motion planning with the genetic algorithm has been suggested in [12], which make it possible to make a change in the path whenever any change occurs in the environment. This adaptation makes the algorithm suitable for both online and offline motion planning. A hybrid method based on A* algorithm and genetic algorithm is proposed in [13] to find the shortest path in a grid environment, by connecting the vertexes of the adjacent obstacles. Then the path is optimized by using a genetic algorithm as a final step. A variable-length chromosome is introduced in [14] as a novel approach, to find the shortest path for a mobile robot in a static environment and a near-optimal obstacle-free path in a dynamic environment. A new genetic algorithm mutation operator improved in [15] and applied to the path planning problem of mobile robots. In simultaneous manner, the mutation operator checks all the free nodes close to mutation node, which is assumed to be better than randomly selecting a node one by one. A hybrid approach consists of Artificial Potential Field and Genetic Algorithm developed in [16] for the mobile robot path planning with a dynamic environment. The grid method is used to represent the environment with orderly numbered, and then Genetic Algorithm is used as global planner and Artificial Potential Field as local planner between intermediate nodes. In addition to the Genetic Algorithm, the Neural Network is used in [7], where the Neural Network is used to establish a relationship between the collision avoidance path and the output by constructing a model of the environment information in the robot's workspace. The concept of selfish planning presented by the authors of [17], states that each robot selfishly chooses the best paths which are either feasible or optimal paths regardless of the other robots. The authors used the Genetic Algorithm two times, first time in the path planning of each robot then in the coordinating between the paths of the robots. The Genetic Path Planner presented in [18] used continuous paths instead of sub-goals, and it is not limited to binary environment maps.

3. Paths Generation

A two-dimensional robot workspace consists of black and white areas represents the environment of the robots, Figure (1). The white areas represent the free space where the robots can move through it without any problem. The black areas represent the obstacles where the robots cannot move through it. The blue dots Rs_n represent the starting points of the robots, and the red dots Rg_n represent the goal points.

The shapes of robots are all the same, and it is a rigid disc that can move in any direction without turn, the planner will deal just with the x and y coordinates of the robots. To make it easy for the planner to plan for a point robot rather than disc robot, the obstacles in the workspace can be grown or dilated [19] by the amount of the robots radius plus a safe factor to prevent the robots from touching the obstacle boundaries and leaving a safe distance as in Equation (1).

$$O_{new} = dil(O_{old}, r) + s \qquad (1)$$

where (O_{new} and O_{old}) are the new and old shapes of the obstacle before and after dilation, *r* is the radius of the disc robot or the structure element that represents the amount of the added size to the obstacles, *s* is the safe factor. Dilating the obstacles and shrinking the robots from disc robots to point robots, this converts the workspace to the configuration space [1]. After the dilation process, the graph generation step starts by randomly generating a number of distributed points in the free space part of the configuration space. The number of points should be balanced between the amounts of the free space to the reserved area by obstacles and the dimensionality of the configuration space. In this paper, Equation (2) is used to determine the number of points to be generated.

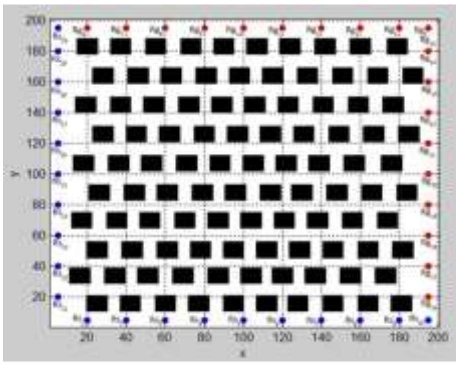


Fig. (1) Robot Configuration space

$$no. Points = \frac{free\ space}{reserved\ pace} * dim(C) \qquad (2)$$

where $dim(C)$ stand for dimension of the configuration space

Then every generated point is connected to its reachable neighbor points according to the distance factor d to finish building the graph, which will be used to find a near-optimal path for each robot, Figure (2). A distance factor is used to make sure a balanced graph is created rather than fully connected or poorly connected graph, and it represents the longest allowed edge between any two points in the graph. In this paper, the value used for d is 30% of the largest dimension.

The start and goal points of every robot are connected to the nearest node in the graph. From the generated graph and by using A* algorithm, the path of each robot is founded which represent the shortest feasible near-optimal path, Figure (3).

Until this step, the paths of the robots are established which means it will be used to reach its goals. However, these paths do not represent optimal paths because those paths are restricted by the edges of the graph. Moreover, there is no flexibility for robots to change its path to make it shorter.

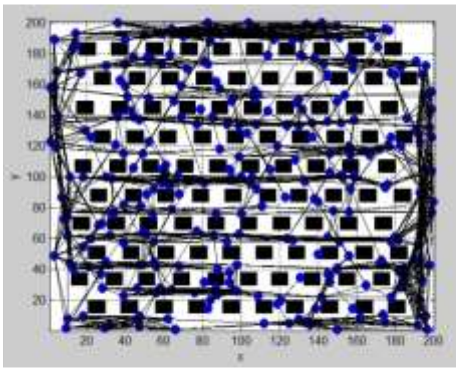


Fig. (2) Graph Generation

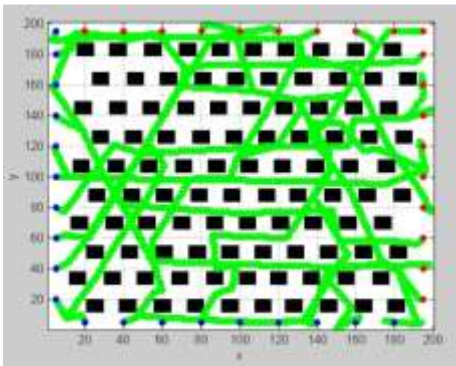


Fig. (3) Paths Generation

4. Path Optimization

One of the most powerful algorithms in search and optimization is the Genetic Algorithm, because it works in a parallel manner and simultaneously estimates many points at the same time in the solution space. The central principle of Genetic algorithm is survival of the fittest; this is done by looking for the best solution in a particular set of solutions. This set of solutions is called the search space which is the space of all feasible solutions (also called population). Every point in this search space represents a possible solution (chromosome). A chromosome consists of a fixed number of genes which represent the behavior of the chromosome, and every gene has a value called allele. The chromosome marked with a value called fitness value that is computed by the fitness function. The fitness value represents the criterion of survival among other chromosomes in the population. A number of operators control the process such as selection, crossover, and mutation. To evolve the population, the Genetic Algorithm loops over an iteration process [20, 21].

The iteration process of a traditional Genetic Algorithm consists of the following steps [20]:

- **Selection**: selecting chromosomes for reproduction, the selection is done randomly with a probability value depending on the fitness of the chromosomes.
- **Reproduction**: the selected chromosomes breed new offspring. By using crossover and/or mutation, new chromosomes are generated.
- **Evaluation**: the fitness of the new chromosomes is evaluated.
- **Replacement**: in the last step, chromosomes from the old population are removed and replaced by the new ones.

When the population converges toward optimal solution; the algorithm stops searching. The paths obtained from the previous section are optimized in this section with the same explained genetic algorithm but with some changes in the encoding of the chromosomes, how to generate the initial population and adding two operators to the algorithm that are correct and refined.

4.1 Chromosome encoding

The success of finding a solution is mainly depending on the success in encoding or representing the chromosomes in a proper way. Most of those problems use binary encoding, but with the path planning problem the numeric encoding is more proper. The environment is represented by a grid-based map with x and y coordinates. So the chromosome will be represented by a number of genes; every gene contains two values that are the x and y coordinates of the grid, Fig (4).

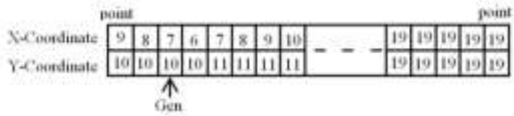


Fig. (4) Chromosome

Initial population generation

The way that is used in this paper to generate the initial population differs from the classic method by generating a random population. The population is created by translating the original path to the right and left if it is moving vertically, and up and bottom if it is moving horizontally, Figure (5). By translating, we mean that a copy from the original path is made at one-unit position distance, then at two-unit position distance, etc. For the points of the path that may cross the obstacle should be eliminated to prevent from generating infeasible paths. The gaps that generated from eliminating the points are treated by the Correction operation which will explain later in this section. After the operation finished, it can be noticed that every path have new start and goal points, so it need to be changed to the original start and goal points. Again, it is the job of the Correction operation, and this done by connecting the first point in the path with the starting point and the last point in the path with the goal point. The size of the population should be large enough to make it_possible to give a chance to go around the near obstacles which may lead to a shorter path; in the same time it should not be vast which may affect the time of execution. The population size used here in this paper is 40 individuals. After the correction operation, 50% of the population size which is 20 individuals used for the reproduction process.

4.2 Fitness function

The fitness function in the genetic algorithm is strongly related to the objective function of the problem under study, and it represents the metric that determine which chromosome or individual will survive and breed. In the multi-robot path planning, the objective function of the problem is to find the feasible shortest path for each robot from the start position to the goal position. Therefore, the fitness function of the problem will be evaluating the length of the path, and the fitness value will be its path, as in Equation (3). In this paper, the unit of measurement that will be used to compute the robot's path is the pixel, and the length of the path will be the number of pixels the robot may follow as in Equation (4). The goal of the work will be minimizing the fitness value using the fitness function.

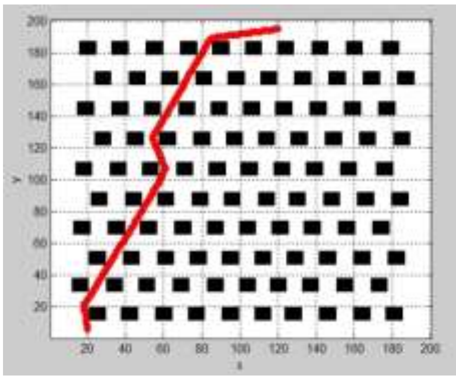
$$Min(path_1, \dots, path_k) \qquad (3)$$

where k=1, … no. of feasible paths, and feasible path defined as follow:

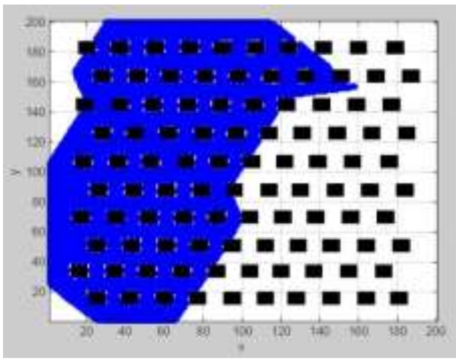
$$path(a,b) = \|a(x,y) - b(a,b)\| \qquad (4)$$

4.3 Selection

The process of choosing two parents from the population to crossover is called selection. Among all types of selection which are Roulette Wheel Selection, Random Selection, Rank Selection, Boltzmann Selection, and Tournament Selection, the Random Selection used here. The reason behind choosing the Random Selection is the wish to not exclude any individual in the population from any crossover operation even if this individual has low-fitness value. Sometimes, portion of the low-fitness individual can give the opportunity to arrive at area in the solution space that may lead to high fitness individuals.



(a) The original path



(b) Population generated from the original path with size 40

Fig. (5) Population generation

4.4 Crossover

The main task of crossover is to combine the features of the two-parent chromosomes to make two new offspring chromosomes. A multi-point crossover between the parents is used here in this paper. In the multi-robot path planning problem, every robot has its path that will be different in length because there is a difference between the start and goal points. Therefore, after selecting the paths, we need to make it equal in size before the crossover. The method adopted here is by extending the shorter one by duplicating some equally distributed points along the path. By applying the crossover, the resulted individuals will consist of disconnected portions with some duplicated points.

The next operation will be responsible for treating this infeasibility in the path.

4.5 Correction

The correction process implemented two times in the optimization algorithm, and it is responsible for two tasks. It is implemented after the initial population generating and after the crossover operator. The main reason of using it after the initial population generating is because of the gaps that resulted from the intersection of the path with the obstacles, and connecting the starting and the ending points. The first task of the correction operation is ensuring there is a path from every point to its pre- and post-neighbor points in the path and fill any gaps that may occur especially after the initial population generating. It is done by drawing a straight line from every point to its next point. If an obstacle exists between the points, then the correction algorithm will choose to move on the boundary of the obstacle and takes the shortest part of the movement. The second task is eliminating any duplicate entries that may occur in the genes of the chromosome after the crossover operation when the algorithm makes the paths equal in length. The resulted path will be a long path, but it combines features from two different paths.

4.6 Refinement

The refining process is responsible for shortening the path resulted after the correct operation. The strategy adopted here is by scanning the path point after point and connecting every point with the farthest reachable point (starting from the end of the path) that not covered by an obstacle. The points between the connected points are discarded. The obtained path will be shorter and created from different features that belong to the parents. This gives the path the opportunity to reach regions in the free space that may lead it to a shorter path to the goal.

Figure (6) shows the GA-based Path Smoother flow chart where two operations added to the original algorithm which are they correction and refinement.

5. Simulation Results

Figure (7) shows the four operations that applied to the parents to produce the children or the offspring after the population generation and fitness evaluation. Where figure (7a) shows the first step in the reproduction operation which will be randomly selecting two individuals from the population. It can be noticed how the individuals shares the same start and end points, to guarantee feasible solutions that have the originals start and end points. After that, figure (7b) show the crossover operation is applied to the selected individuals and how parts from each one are transferred to the other. In this paper, the type of crossover adopted is the multi-points and three points crossover used here in this figure. The idea behind using multi-point crossover is to give the opportunity to the path to cover different regions in the space. The resulted path from the previous process is fragmented path to several unconnected parts. A correction process is applied to every fragmented path which makes sure to convert the path to connected feasible path, as shown in Fig. (7c). In Fig. (7d), the resulted corrected paths are very long paths that need to make them as short as possible by using the refinement operation. The final generated paths can be characterized as feasible near-optimal paths covers new areas in the space that not covered by the initial paths.

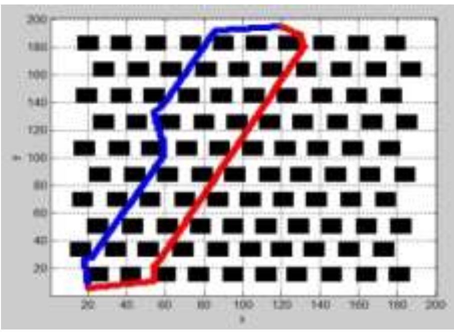
The simulation software is written in MATLAB R2013a 64-bit and implemented in HP Pavilion dv6 Notebook PC with Intel Core i7 at 2.2 GHz processor and 8GB of RAM. The path planner is applied on a grid of size 200×200 as a bitmap image with two colors; white color represents the free space and black color represents the obstacles or reserved space.

A simulation experiment is performed using the proposed algorithm for the optimization path planner. Table (1), shows the measurements of ten robots scenario and 50% selection rate from the initially generated population which was 40 individuals. The crossover implemented with 3-points in this situation. The first column in the above mentioned table represents the length of the original path obtained from the probabilistic path planner before the optimization algorithm. The second column represents the optimized paths after applying the optimization algorithm. The third and fourth columns state the difference in length (between the original and optimized paths) and optimization rate. The difference that noticed between the optimization rates of the robots resulted from the probabilistic roadmap that randomly generated, where some paths optimized more than the others. Therefore, the optimization rate average of the obtained results shown in table (1) is 18.25%.

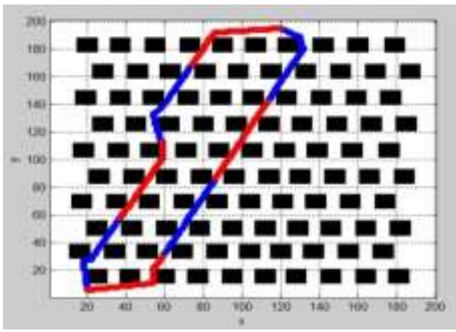
Table (1) Generation Steps

Robot No.	original length	optimized length	difference in length	optimization ratio
1	231	195	36	15.58%
2	268	190	78	29.10%
3	267	232	35	13.10%
4	244	193	51	20.90%
5	225	193	32	14.22%
6	244	204	40	16.39%
7	233	203	30	12.87%
8	255	202	53	20.78%
9	248	200	48	19.35%
10	247	197	50	20.24%

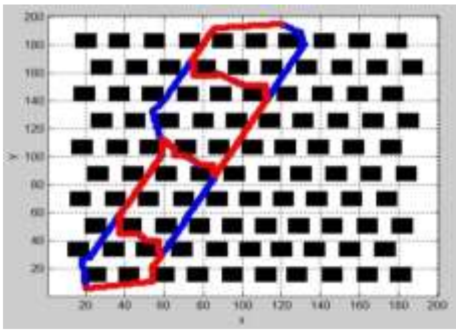
 ISSN (printed) 1813-2065, (online) 2309-1673



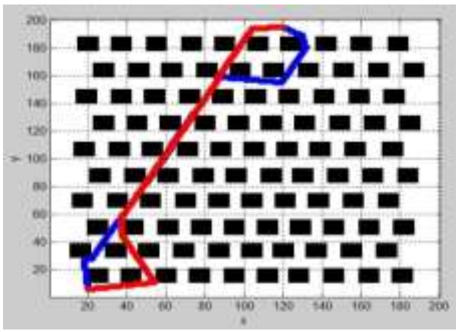
(a) Selection



(b) Crossover



(c) Correction



(d) Refinement

Fig. (7) Operations applied to parents to produce the offspring

6. Conclusion

In this paper, a two steps algorithm is proposed here to find the optimal path planning for the multi-robot problem. In the first step, a probabilistic roadmap is built in the free configuration space of the robots work space which avoids the static obstacles. Then by using the A* algorithm and the probabilistic roadmap created, the near-optimal path for each robot is found which represents the shortest path between the point in the start position and the point in the ending position. The second step used an improved genetic algorithm to make these paths more optimal to guarantee the shortest paths, which represents an intermediate process located between roadmap creation and path following. A new population generation method used in this algorithm and new operators in addition to the selection and crossover in the genetic algorithm also used to ensure the feasibility of the results. The results of the simulation gained from the proposed method show that it finds the shortest path for each robot with optimization rate average of 18.25% from the original path, and it is fast enough to find the optimal paths in few generations.

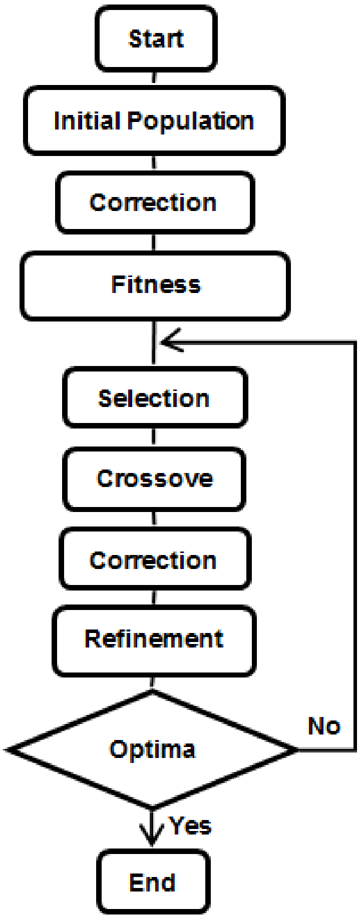


Fig. (6) GA-based Path Smoother

References

[1] J.-C. Latombe, Robot Motion Planning, Kluwer Academic Publishers, 1991.

[2] S.M. LaValle, PLANNING ALGORITHMS, Cambridge University Press, 2006.

[3] J. v. den Berg, J. Snoeyink, M. Lin and D. Manocha, "Centralized Path Planning for Multiple Robots: Optimal Decoupling into Sequential Plans," *Robotics: Science and Systems V, MIT press,* 2010.

[4] J.-H. Oh, J.-H. Park and J.-T. Lim, "Centralized Decoupled Path Planning Algorithm

for Multiple Robots Using the Temporary Goal Configurations," in *Third International Conference on Computational Intelligence, Modelling & Simulation*, Langkawi, 2011.
[5] W.-G. Han, S.-M. Baek and T.-Y. Kuc, "GA based online path planning of mobile robots playing soccer games," in *Proceedings of the 40th Midwest Symposium on Circuits and Systems*, Sacramento, CA, 1997.
[6] J. Reif and M. Sharir, "Motion planning in the presence of moving obstacles," in *26th Annual Symposium on Foundations of Computer Science*, Portland, OR, USA, 1985.
[7] D. Xin, C. Hua-hua and G. Wei-kang, "Neural network and genetic algorithm based global path planning in a static environment," *Journal of Zhejiang University Science*, vol. 6, no. 6, pp. 549-554, June 2005.
[8] S.M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Iowa State University, 1998.
[9] L.E. Kavraki, P. Svestka and J.-C. Latombe, "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces," IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, vol. 12, no. 4, pp. 566-580, 1996.
[10] J. H. Holland, Adaptation in Natural and Artificial Systems, The University of Michigan Press, 1st edition, 1975.
[11] S. Liu, Y. Tian and J. Liu, "Multi Mobile Robot Path Planning Based on Genetic Algorithm," in Proceedings of the 5th World Congress on intelligent Control and Automation, Hangzhou, P.R. China, 2004.
[12] K. Sugihara and J. Smith, "Genetic Algorithms for Adaptive Motion Planning of an Autonomous Mobile Robot," in IEEE International Symposium on Computational Intelligence in Robotics and Automation, Monterey, CA, 1997.
[13] L. Zhang, H. Min, H. Wei and H. Huang, "Global Path Planning for Mobile Robot Based on A* Algorithm and Genetic Algorithm," in International Conference on Robotics and Biomimetics, Guangzhou, China, 2012.
[14] J. Tu and S. X. Yang, "Genetic Algorithm Based Path Planning for a Mobile Robot," in lEEE lnlernational Conference on Robotics and Automation, Taipei, Taiwan, 2003.
[15] A. Tuncer and M. Yildirim, "Dynamic path planning of mobile robots with improved genetic algorithm," *Computers & Electrical Engineering*, vol. 38, no. 6, p. 1564–1572, November 2012.
[16] Y. Liu and K. K. Bharadwaj, "A Hybrid Artificial Potential Field: Genetic Algorithm Approach to Mobile Robot Path Planning in Dynamic Environments," *Computer Science and Convergence: Lecture Notes in Electrical Engineering*, vol. 114, pp. 325-333, 2012.
[17] T. Shibata, T. Fukuda, K. Kosuge and F. Arai, "Selfish and coordinative planning for multiple mobile robots by genetic algorithm," in *Proceedings of the 31st IEEE Conference on Decision and Control*, Tucson, AZ, 1992.
[18] M. Gerke, "Genetic Path Planning for Mobile Robots," in *Proceedings of the American Control Conference*, San Diego, California, 1999.
[19] F.Y. Shih, Image processing and mathematical morphology Fundamentals and Applications, CRC Press, 2009.
[20] S.N. Sivanandam and S. N. Deepa, Introduction to Genetic Algorithms, Springer-Verlag Berlin Heidelberg, 2008.
[21] M. Melanie, An Introduction to Genetic Algorithms, Cambridge, Massachusetts, London, England: Massachusetts Institute of Technology, 1996.

