Journal of Computer Science 4 (4): 341-344, 2008
ISSN 1549-3636

A Mobile Robot Path Planning Using Genetic Algorithm in Static Environment

Ismail AL-Taharwa, Alaa Sheta and Mohammed Al-Weshah
Department of Information Technology, Faculty of Science and Information Technology,
Al-Balqa Applied University, Al-Salt, Jordan

Abstract: In this study we present our initial idea for using genetic algorithms to help a controllable mobile robot to find an optimal path between a starting and ending point in a grid environment. The mobile robot has to find the optimal path which reduces the number of steps to be taken between the starting point and the target ending point. GAs can overcome many problems encountered by traditional search techniques such as the gradient based methods. The proposed controlling algorithm allows four-neighbor movements, so that path-planning can adapt with complicated search spaces with low complexities. The results are promising.

Key words: Path planning, genetic algorithms, robotics

INTRODUCTION

In the past two decades, different conventional methods have been developed to solve the path planning problem, such as the cell decomposition, road map and potential field [6, 7]. Most of these methods were based on the concept of space configuration [8]. These techniques show lack of adaptation and a non robust behavior. To overcome the weakness of these approaches researchers explored variety of solutions [1]. GA has been recognized as one of the most robust search algorithms for complex and ill-behaved optimization problems [9]. The basic characteristic which makes GA's attractive in solving such types of problem, is that they are inherently parallel search techniques and can search for optimal in dynamic environments [1, 10, 11]. GA was used to control a mobile robot moving in an environment which has number of static obstacles and to control robot motion with dynamic obstacle [12]. Some of the proposed techniques in [10-12] suffer from many problems. They include (1) computationally expensive (2) requires large memory spaces when dealing with dynamic and large sized environments, (3) time consuming. In [13,14], authors used Fuzzy Logic and Neural Network to handle the path planning problem, respectively.

In the last decade, genetic algorithms have been widely used to generate the optimum path by taking the advantage of its strong optimization ability. This research is motivated by earlier work presented in [1]. In this study, we provide an initial idea based genetic algorithm to select the shortest path in predictable environment which will be able to handle static obstacles. We propose a mobile robot controlling algorithm which allows four-neighbor movements, so that path-planning can handle complicated search spaces with low complexities.

Problem formulation: The mobile robot path planning problem is typically formulated as follows: given a mobile robot and a description of an environment, we need to plan a path between two specified locations, a start and end point. The path should be free of collision and satisfies certain optimization criteria (i.e., shortest path)[3]. According to this definition, path planning problem is categorized as an optimization problem.

Researchers distinguish between various methods used to solve the path planning problem according to two factors, (1) the environment type (i.e., static or dynamic[5], 2) the path planning algorithms (i.e., global or local)[3]. The static environment is defined as the environment which doesn't contain any moving objects other than a navigating robot; while the dynamic is the environment which has dynamic moving objects (i.e., human beings, moving machines and moving robots).

The global path planning algorithms requires a complete knowledge about the search environment and that all terrain should be static. On the other hand, local path planning means that path planning is being implemented while the robot is moving; in other words, the algorithm is capable of producing a new path in response to environmental changes [3].

Corresponding Author: Ismail AL-Taharwa, Department of Information Technology, Faculty of Science and Information Technology, Al-Balqa Applied University, Al-Salt, Jordan

Proposed algorithms: To use GAs for solving the path planning problem, we considered number of steps. These steps are:

- First: Convert the search environment to a grid graph (i.e., node). Thus, the robot shall move in a step fashion on the proposed grid as they appear in the real environment
- Second: Specify the staring and ending point where the path need to be established
- Third: Defining the static obstacles locations on each node of the grid

Path planning technique: In this study, we present the main steps for the proposed path planning techniques.

Initialization: Create an initial population with a predefined population size. The population contains number of individuals (i.e., chromosomes). Each individual represents a solution for the problem under study. In our case, each solution is in fact a path between the start and end point in the search space. The initial population with size n can be presented as:

$$\text{Initial Population} = <p_1, p_2, \ldots, p_n>$$

Each structure p_i is simply an integer string of length L, in general. Each structure p_i represents a vector of node numbers in the grid which can take values of 1, 2, …, L (i.e., search space).

Normally, GAs individuals can contain any point value between the starting and ending point. Thus, the individual generated by GAs is in the form of:

$$<c_1, c_2, \ldots, c_l>$$

where, l is the number of visited node in the search space. The starting and ending point will not be shown in this individual. This is why we need to make some modification to the individual structure so that we can add the starting and ending point. The modified individual representation will be:

$$<c_{start}, c_1, c_2, \ldots, s_l, c_{end}>$$

Fitness function: Fitness function represents an important part of any evolutionary process using GAs. Appropriate selection of the fitness function will lead the search towards the optimal solution.

The optimal path, in our case, is the shortest path between the starting and ending point. Thus, the fitness function is responsible on finding this path. The shortest path helps computing the total number of steps the mobile robot need to take to reach the ending point.

Consequently, the fitness value for a complete solution will be computed as:

$$pi = d_0 + d_1 + \ldots + d_m$$

d_0 = The distance between c_{start}, c_l
d_1 = The distance between c_1, c_2
d_2 = The distance between c_2, c_3
d_m = The distance between c_1, c_{end}

Fitness computation: To compute the fitness function for an individual, we should have the coordinates of each point in the individual (i.e., column and row for each c_l from a lookup table). Thus, we can compute the distance between any two points in the search space (i.e., environment of the robot).

Assume we have two points in the search space $x_{current}$, x_{next}. The distance between the two points can be computed using the following equation:

$$D = \left| x_{next,col} - x_{current,col} \right| + \left| x_{next,row} - x_{current,row} \right|$$

Absolute value is important since distance is a quantity value.

Experiments: In this study, we present our development experiments in with two types of environment (i.e., obstacle free and obstacle environment).

Obstacle free environment: We used Genetic Algorithms to search a space of 10×10 and 100×100 nodes to find an optimal path for a mobile robot to move from a start to end points. In our experiment we used an individual structure of 7 elements and 18 elements, respectively. This means that the mobile robot could visit the same point number of times. This is why GAs has to pick up the best path which avoids this type of problems.

We ran GAs with various population sizes 10, 20 and 50, respectively. The goal is to investigate about the behavior of GAs in each case. This will also help in showing that GAs will converge to the optimal solution (i.e., optimal path) in each run.

In Table 1 and 2, we show the computed results using GAs with various population sizes, the best fitness value achieved after each run and the generation number where this value was found.

Table 1: A 10×10 grid with obstacle free environment

Exp #	Population size	Best fitness value	Generation No.
1	10	20	60
2	20	18	60
3	50	18	8

Table 2: A 100×100 grid with obstacle free environment

Exp #	Population size	Best fitness value	Generation No.
1	10	198	80
2	20	198	40
3	50	198	30

Table 3: Indoor-like environment

Exp #	Population size	Best fitness value	Generation no.	Color
1	10	38	100	Red
2	20	32	100	Pink
3	50	28	60	Blue

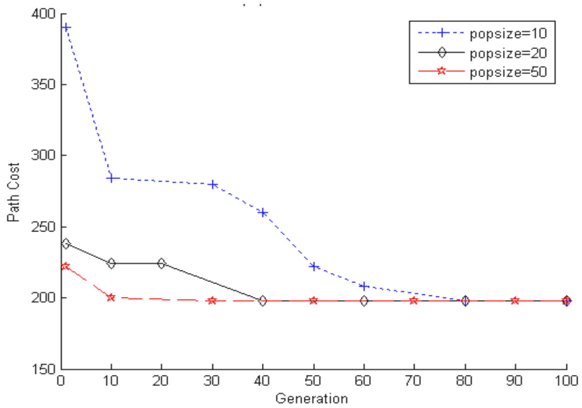

Fig. 1: Convergence process for GAs with a 100×100 grid with obstacle free environment

In Fig. 1, we show the convergence process for the GA in the case of the second experiment (100×100 nodes). It is shown that with various population sizes, GA selected the optimal path in all cases. The best fitness value found equal to 198.

Obstacle avoidance: In this study we propose three different environments with various obstacles locations. They include:

- Indoor environment
- Moderate scattered environment
- Complex scattered environment

The three environments are shown in Fig. 2, 3 and 4, respectively. In Table 3, 4 and 5, we show the best discovered paths using GAs with the three explored environment.

Table 4: Moderate scattered environment

Exp #	Population size	Best fitness value	Generation no.	Color
1	10	26	90	Red
2	20	22	90	Pink
3	50	20	70	Blue

Table 5: Complex scattered environment

Exp #	Population size	Best fitness value	Generation no.	Color
1	1000	26	80	Red
2	1200	26	30	Pink
3	2000	24	20	Blue

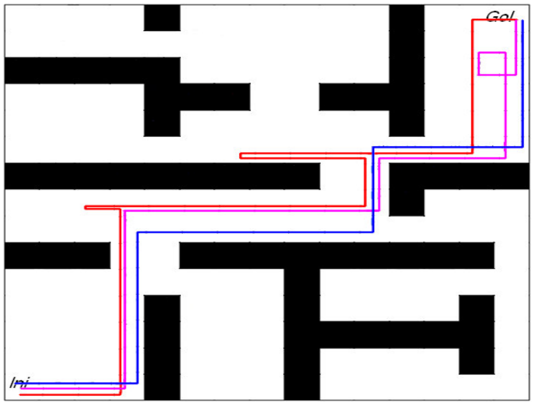

Fig. 2: Indoor-like environment

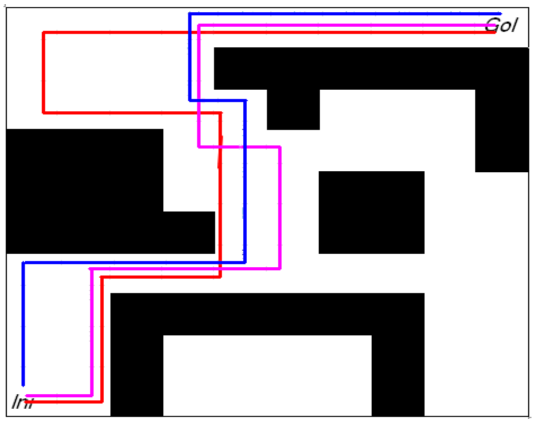

Fig. 3: moderate scattered environment

In Fig. 2, 3 and 4 we also show three best paths developed with various colours (i.e., red, pink and blue). It was found that increasing the number of generation helps finding the shortest path with less number of generations. Although, we know that the larger the number of individual in the population (i.e., large population size) is the larger number of computation.

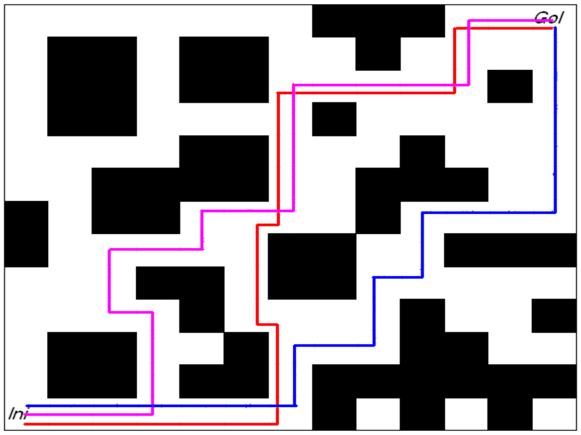

Fig. 4: Complex scattered environment

CONCLUSIONS

In this study we presented the idea of using Genetic Algorithm approach to solve the mobile robot path planning problem in static environment with predictable terrain. We proposed a simplified fitness function which utilizes the path length. We explored the performance of the evolutionary process with varies population size. Our preliminary experiments show that the proposed approach is effective and efficient in handling different types of tasks in static environments.

REFERENCES

1. A. Elshamli, H. A. Abdullah and S. Areibi, "Genetic algorithm for dynamic path planning". In: Proc. Canadian Conf. Elect. and Comput. Eng., Niagara Falls, Vol.2, May 2-5, 2004, pp: 677-680, ISBN: 0-7803-8253-6/04/$/7.00© 2004 IEEE.
2. Y. Davidor, "Robot programming with a genetic algorithm,". In Proc. 1999 IEEE Int. Conf. Comput. Syst. and Software Eng., Tel Aviv, Israel, pp: 186-191.
3. K. H. Sedighi, K. Ashenayi, T. W. Manikas, R. L. Wainwright, and H. M. Tai, "Autonomous Local Path Planning for a Mobile Robot Using a Genetic Algorithm," in Congr. Evol. Comput. 2004. CEC2004., Vol. 2, pp. 1338-1345.
4. A. Sheta and H. Turabieh, (2006, January). A comparison between genetic algorithms and sequential quadratic programming in solving constrained optimization problems.The Int. J. of Artificial Intell. and Mach. Learning 6(1), pp. 67-74.
5. A. Ramirez-Serrano, H. Liu and G. C. Pettinaro. (2008). Mobile robot localization in quazi-dynamic environments. Int. J. on Ind. Robot, pp.246-258.
6. B. Graf, J. M. Hostalet Wandosell, "Flexible path planning for nonholonomic mobile robots,". In Proc. 4th European workshop on advanced Mobile Robots (EUROBOT'01)., Fraunhofer Inst. Manufact. Eng. Automat. (IPS). Lund, Sweden, Sept. 19-21, 2001, pp. 199-206.
7. J. C. Latombe, Robot Motion Planning. 1st ed, Boston, UK, MA: Kluwer Academic Publishers (Published by Springer), 1991, ISBN: 0-7923-9129-2.
8. T. Lozano-Perez and M. Wesely, (1979, October). An algorithm for planning collision-Free paths among polyhedral obstacles. Commun. of the ACM [Online], 22(10), pp. 560-570.
9. D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. 1st ed. ISBN: 0-201-15767-5, Boston, MA: Addison-Wesley Longman publishing Co., Inc., 1989.
10. C. E. Thomas, M. A. C. Pacheco, M.M. and B.R. Vellasco, "Mobile Robot Path Planning Using Genetic Algorithms,". In foundations and tools for neural modeling, vol. 1606/1999, Springer Berlin/ Heidelberg, ISBN: 3-540-66069-0, 1999, pp. 671-679.
11. J. Brank, "Evolutionary approaches to dynamic optimization problems-introduction and recent trends," [Online] In Proc. GECCO Workshop on Evol. Algorithms for Dynamic Optimization Problems, Chicago, USA, 2003, pp. 2
12. J. Lu and D. Yang, "Path planning based on double-layer genetic algorithm," in 3rd Int. Conf. on Natural Computation (ICNC 2007) [Online], Hangzhou, Haikou, China, Aug. 24-27, 2007, Vol. 4, pp: 357-361.
13. A. Ramirez-Serrano and M. Boumedine, "Real time navigation in unknown environments using fuzzy logic and ultrasonic sensing," in Proc. 1996 IEEE Int. Symp. In Intell. Control, Dearborn, MI, ISBN: 0-7803-2978-3, Y de Estudios supeiores de Monterrey, 15-18 Sept, 1996, pp: 26-30, DOI: 10.1109/ISIC.1996.556172.
14. I.K. Jung, K.B. Hong, S.K. Hong and S.C. Hong, "Path planning of mobile robot using neural network," in Proc. IEEE Int. Symp. on Ind. Electron,. Sch. Elect. Eng., 1999 ISIE Bled Slovenia, Vol. 3, 12-16 July, 1999, pp. 979-983, DOI: 10.1109/ISIE,1999.796750.
15. J.J. Grefenstette, "A User's guide to GENESIS," Technical Report, CS-84-11, Department of Computer Science, Vanderbilt University, Nashville, TN.