Proceedings of the 2003 IEEE
International Conference on Robotics & Automation
Taipei, Taiwan, September 14-19, 2003

Genetic Algorithm Based Path Planning for a Mobile Robot*

Jianping Tu and Simon X. Yang†
Advanced Robotics and Intelligent Systems (ARIS) Lab
School of Engineering, University of Guelph
Guelph, ON N1G 2W1, Canada
Emails: {jtu; syang}@uoguelph.ca

Abstract - In this paper, a novel genetic algorithm based approach to path planning of a mobile robot is proposed. The major characteristic of the proposed algorithm is that the chromosome has a variable length. The locations of target and obstacles are included to find a path for a mobile robot in an environment that is a $2D$ workplace discretized into a grid net. Each cell in the net is a gene. The number of genes in one chromosome depends on the environment. The locations of the robot, the target and the obstacle are marked in the workplace. The proposed algorithm is capable of generating collision-free paths for a mobile robot in both static and dynamic environments. In a static environment, the generated robot path is optimal in the sense of the shortest distance. The effectiveness of the proposed model is demonstrated by simulation studies.

1 Introduction

Genetic algorithms (GAs) are search algorithms and optimization techniques using the principles of natural selection inspired by Darwin's theory about evolution (the survival of the fittest.) In GA based approaches, the variables are represented as genes on a chromosome. Genetic algorithms feature a group of candidate solutions (population) on the response surface. Through natural selections and the genetic operations, recombination and mutation, chromosomes with better fitness are found.

Path planning is one of the important tasks in intelligent robotic systems such as an autonomous mobile robot. These are two fundamental types of path planning: (1) dynamic adaptive path planning, which allows a mobile robot to respond to unexpected situations while engaged in the problem-solving process, i.e. it is capable of producing a new path in response to environmental changes, and (2) static path planning, which allows a mobile robot to move through slower, more intensive processes. In a static environment, it is a known terrain and a path can be generated in advance. Sometimes, path planning is either constrained or unconstrained, according to whether there are inherent restrictions on the motion of robots, i.e., restrictions arising due to reasons other than collisions with obstacles. These include bounds on a robot velocity and acceleration and constraints on the curvature of robot paths.

There are many approaches suggested by researchers to solve the path planning problems of mobile robots in presence of static and moving obstacles. One of the popular path planning methods is the artificial potential field [3]. However, it can give only one solution route that may not be the shortest path, even in a static environment. Another method is a position estimation method and a path generation between current location and target location for mobile robots – Dead-reckoning. It has been commonly used for position estimation. However this method accumulates estimation errors [1, 2]. There have also been many attempts to use fuzzy logic controllers for motion planning of mobile robots [4, 5]. In [4], while steering a robot, an off-line process modeling develops an operator the fuzzy rules (knowledge) of the system. The fuzzy logic controllers should be properly designed to produce an efficient controller and the fuzzy rules must adequately model the human approach to control the system.

Recently, it has been widespread interest using genetic and evolutionary algorithms. Compared to traditional search and optimization methods, such as calculus-based and enumerative strategies, the evolutionary algorithms are robust, global and generally more straightforward to apply in situations where there is little or no a priori knowledge about the problem to solve. As evolutionary algorithms require no

*This work was supported by Natural Sciences and Engineering Research Council (NSERC) and Materials and Manufacturing Ontario (MMO) of Canada.

†Corresponding author.

derivative information or formal initial estimates of the solution, and because they are stochastic in nature, evolutionary algorithms are capable of searching the entire solution space with more likelihood of finding the global optimum. The genetic algorithm [6, 7, 8, 10] is a powerful search algorithm based on the mechanism of natural selection and uses operations of reproduction, crossover, and mutation on a population of strings.

Genetic algorithms have been applied to multi-robot path planning [12, 13, 14]. Shibata et al. [12] developed a path planner using the concept of free-convex areas introduced by Habib and Asama [15]. They used two gene-based searching algorithms to solve two easier subparts of the problem: one to find a set of optimal trajectories for each robot under selfish planning and another to select a candidate from the set of trajectories for each robot so as to avoid collisions when all robots work simultaneously. This approach does not take full advantage of the free space available at a certain time. Consequently, results from coordinative planning under this approach are not really optimal alternatives. This is because restrictions imposed on relative motions, in order to avoid collisions, are not similar to what happens in real life. As a result of these findings, a new solution of the problem can now be developed and the results can then be compared.

During the last decade, much work has been done to explore the evolution of robot control strategies. Cliff et al have published a series of technical reports on using genetic algorithms to design neural-network control architectures for a simulated visually-guided robot [9]. Koza has used genetic programming to evolve LISP programs that control and guide a variety of simulated robots performing navigation and other tasks [8, 10]. Murray and Louis [11] first used genetic algorithms to design combinational circuits for basic (low-level) behaviors, then they used the genetic algorithm to design a switch to choose between these low-level behaviors for performing more complex tasks [11].

Commonly, a fixed-length chromosome was used in the literature [12]. A fixed-length chromosome is not suitable for a complex environment, especially, in a dynamic environment. In order to reach the target, in a more obstacles environment, longer chromosome may be needed. That means a fixed-length chromosome may be not enough. Therefore, a variable-length chromosome is better suitable in a dynamic and many obstacles environment. In the proposed model, a novel approach with a variable-length chromosome is proposed. Using this proposed method, a mobile robot can find the shortest path in a static environment and the near-optimal obstacle-free path in a dynamic complex environment.

2 Model Algorithm

In the proposed algorithm, any path from the starting point to the goal is a solution, which is generation. First generation is selected by roulette wheel selection, i.e. at the beginning a large random population of strings is generated. Strings representing unacceptable solutions are eliminated and strings representing acceptable solutions get multiplied. Unacceptable solutions are strings that cannot reach the target. Acceptable solutions are strings that can reach the target. In order to move the mobile robot toward the goal by avoiding the obstacles, it should find a best string that reaches the target in the shortest path. The decision whether the string is acceptable or unacceptable is decided by the fact whether the string solution would lead the mobile robot into obstacles and whether the mobile robot is progressing towards the goal. Based on such decisions each string in the population is assigned a fitness value. Acceptable solutions will have higher fitness value and unacceptable solution would have lower fitness value. Thus a string with a higher fitness value would indicate that the mobile robot is moving towards the goal.

One of the more challenging aspects of using genetic algorithms is encoding of the problem in a binary string, which maps a path from the start point to the target. Each of the individuals (or chromosomes) of a genetic population is encoded by a genotype (genetic information) like in biology. Traditionally, this genotype is represented by a string of bits. A chromosome corresponds to a possible solution of the optimization problem, thus each chromosome is equivalent to a particular set of parameters (a point of the search space). For a bounded search space, each parameter is encode as a series of zeros and ones of a fixed length (depending on the precision required) using the natural gray code for instance (See Fig. 1 on which we have represented a chromosome encoding L parameters with a fixed length for each parameter).

There are two types of chromosome. One is a fixed-length string that the number of genes is the same for one solution. Another one is that different solution has different number of genes. The proposed algorithm represents a novel approach to global optimization using these variable length genetic algorithms. The proposed model in the 2-D workplace has n-by-n grids. A

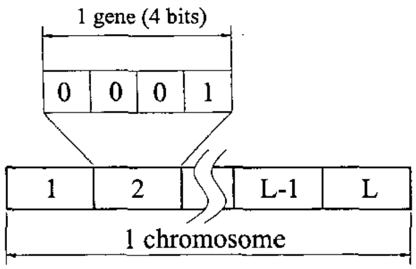

Figure 1: A chromosome encoding.

gene is represented by 4 bits length binary string. The first 3 bits denote the direction the robot will move at the next step. The remainder 1 bit denotes the distance the robot will move at the next step. In this 2-D workplace, one maximum chromosome is assumed to consist of $n \times n$ genes, i.e. the mobile robot goes through all the grids. If $n = 16$, that is $16 \times 16 = 256$ bits binary string to represent a maximum chromosome; one minimum chromosome is assumed to consist of n genes. If $n = 16$, that is 16 bits binary string to represent a minimum chromosome. Therefore, the number of one chromosome is defined as:

$$N_{1chromosome,max} = n^2, \tag{1}$$

$$N_{1chromosome,min} = n. \tag{2}$$

The range of these bits of one chromosome will be between $4n$ and $4n^2$. Through the simulation, the following formula is suggested to represent a chromosome

$$N_{1chromosome,suggest} = 2(n+n) = 4n. \tag{3}$$

After all operations of genetic algorithms, the best string will be obtained. But this is not the last result. In this best string, those genes whose distance is 0 will be eliminated, so the remainder of the best string will be one solution. For the direction of the approach, 000 indicates moving to the right horizontally; 001 indicates moving to the top right corner diagonally; 010 indicates moving to upper vertically; 011 indicates moving to the top left corner diagonally; 100 indicates moving to the left horizontally; 101 indicates moving to the down left corner diagonally; 110 indicates moving to down vertically; 111 indicates moving to the down right corner diagonally (see Fig. 2).

Another of the more challenging aspects of using genetic algorithms is to design a fitness function. It also called objective function. In this 2-D workplace, a damping coefficient to each grid is assigned. If the grid has an obstacle, its damping coefficient is set to 300; if the grid is free, its damping coefficient is set to 0; if the grid is a target, its damping coefficient is set to -0.9. The fitness function is designed as

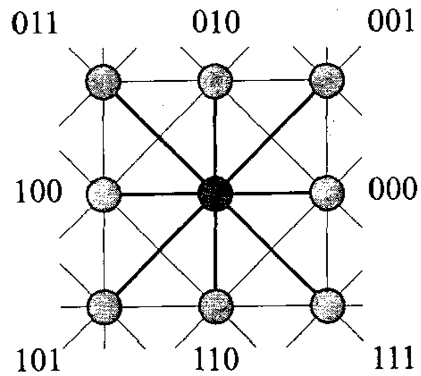

Figure 2: Direction encoding.

$$f = \sum_{j=1}^{K} d_j\,(1 + w_j)\,, \tag{4}$$

where d_j is the distance between two adjacent genes, which is designed as

$$d_j = \begin{cases} 1, & \text{if horizontal or vertical direction,} \\ \sqrt{2}, & \text{if diagonal direction,} \end{cases} \tag{5}$$

parameter w_j is the damping coefficient of jth gene, and K is the total grids a mobile robot will move over.

Choosing the configuration parameter settings is important as well. These include population size, crossover type, crossover probability, mutation probability, number of generation, etc.

The population size dictates the number of chromosomes in the population. Larger population sizes increase the amount of variation present in the initial population at the expense of requiring more fitness evaluations. It has been found that the best population size is both applications dependent and related to the length of the chromosome. A good population of chromosomes contains a diverse selection of potential building blocks resulting in better exploration. If the population loses diversity the population is said to have premature convergence and little exploration is being done. For longer chromosomes and challenging optimization problems, larger population sizes are needed to maintain diversity (higher diversity can also be achieved through higher mutation rates and uniform crossover) and hence better exploration. Many researchers suggest population sizes between 25 and 100.

If the parents (chromosomes in the population) are allowed to mate, a recombination operator is employed to exchange genes between the two parents to produce two children. There are three methods to exchange genes between the two parents. In the one-point method, a crossover point is selected along the chromosome and the genes up to that point are swapped between the two parents. In the two-point method, two crossover points are selected and the genes between the two points are swapped. In the uniform crossover method the recombination is applied to the individual genes in the chromosome. If crossover is performed, the genes between the parents are swapped and if no crossover is performed the genes are left intact. Whether crossover is performed or not depending on crossover rate (recombination rate). The higher the rate is, the more possible to swap genes between two chromosomes. Mutation rate determines the probability that a mutation will occur. Mutation is employed to give new information to the population (uncover new building blocks) and also prevents the population from becoming saturated with similar chromosomes (premature convergence). Large mutation rates increase the probability that good schemata will be destroyed, but increase population diversity. The best mutation rate is application dependent but for most applications is between 0.001 and 0.1. Some researchers have published "rules of thumb" for choosing the best mutation rate based on the length of the chromosome and the population size. DeJong suggested that the mutation rate be inversely proportional to the population size. Hessner and Manner suggest that the optimal mutation rate is approximately as

$$P_m = 1/\left(M\sqrt{L}\right), \tag{6}$$

where M is the population size and L is the length of the chromosome.

3 Simulation Studies

The simulation experiments are carried out using the proposed genetic algorithm based approach. In all figures, it uses filled circle to represent the robot; and the solid diamond to represent the target; and the small dot to represent the empty space; and the filled squares to represent obstacles.

The 2-D workplace has 16 by 16 grids, the start point is at (1,1) and the target at (16,16). It has been considered three different static cases and one dynamic case. In all runs, population size equals to 50, uniform crossover probability equals to 0.9, mutation probability equals to 0.02.

3.1 Path Planning in a Static Environment

In the first case, there are no obstacles in the workplace. The chromosome length is initialized with $4 \times 16 = 64$ [from Eq. (3)]. After 10 generations, the chromosome of the obtained solution is given as 0101 0101 0001 0011 0011 0011 0101 0011 0011 0001 0011 0011 0011 0011 0101 0011 0001 0001 0001 0101. Where the chromosome length is 20. The corresponding robot path is shown in Fig. 3A. After 45 generations, the obtained solution has no change if the evolution continues. Thus, we get the best solution with the chromosome as 0011 0011 0011 0011 0011 0011 0011 0011 0011 0011 0011 0011 0011 0011 0011. Where the chromosome length is 15. The corresponding robot path is shown in Fig. 3B.

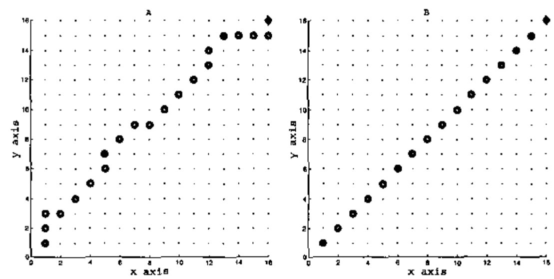

Figure 3: Path planning in a static environment without obstacles. A: one of the obtained path after 10 generations; B: the best robot path after 45 generations.

In the second case, nine stationary obstacles are considered to form an I-shape putting in the middle of the workplace. The chromosome length is initialized with $4 \times 16 = 64$ [from Eq. (3)]. After 60 generations, the chromosome of the obtained solution is given as 0001 0001 0011 0001 0001 0001 0011 0001 0011 0011 0011 0011 0011 0011 0011 0101 0101 0101 0101 0101 0101. Where the chromosome length is 21; The corresponding robot path is shown in Fig. 4A. After 30 more generations, we get the best solution with the chromosome as 0011 0011 0011 0101 0011 0101 0011 0101 0101 0011 0011 0101 0001 0011 0011 0011 0001 0001 0001 0001. Where the chromosome length is 20. The corresponding robot path is shown in Fig. 4B.

In the third case, there are complex obstacles in the workplace. The chromosome length is initialized

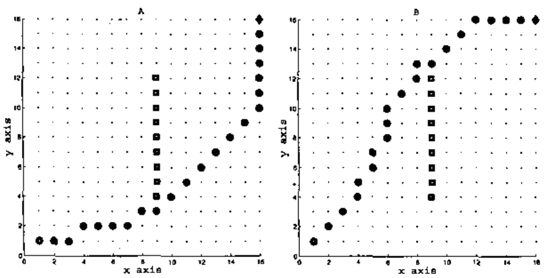

Figure 4: Path planning in a static environment with I-shape obstacles. A: one of the obtained path after 60 generations; B: the best robot path after 90 generations.

with $4 \times 16 = 64$ [from Eq. (3)]. After 82 generations, the chromosome of the obtained solution is given as 0101 0101 0101 0101 0101 0001 0101 0101 0101 0011 0011 0011 0011 0001 0001 0001 0001 0001 0011 0001 0011 0001 0001 0101. Where the chromosome length is 24. The corresponding robot path is shown in Fig. 5A. After 286 generations, the obtained solution has no change if the evolution continues. Thus, we get the best solution with the chromosome as 0011 0011 0011 0001 0011 0011 0011 0011 0011 0011 0101 0011 0011 0011 0011 0011. Where the chromosome length is 16. The corresponding robot path is shown in Fig. 5B. In this case, because there are complex obstacles that the mobile robot must avoid, the number of the generation is relatively large and the time the mobile robot spends is relatively long. But, at last, the mobile robot can find the optimum path from the start to the target with obstacles avoidance.

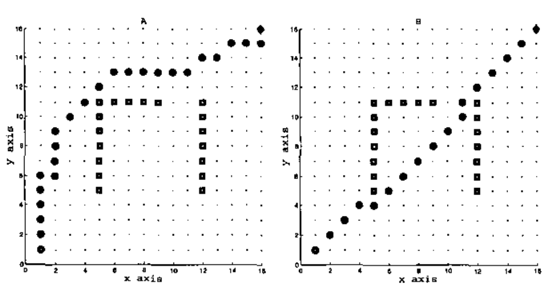

Figure 5: Path planning in a static environment with obstacles. A: one of the obtained path after 82 generations; B: the optimal robot path after 286 generations.

3.2 Path Planning in a Dynamic Environment

Next, simulation studies on the dynamic environment are presented. There are 12 stationary obstacles and 4 moving obstacles in the terrain, the speed of the moving obstacles is 1 grid per 5 generations. After 80 generations, these moving obstacles are gone. Fig. 6A to Fig. 6F are the records of this processing. After 98 generations, Using the proposed algorithm, a mobile robot still can find a feasible optimal path from the starting point to the target with obstacles avoidance.

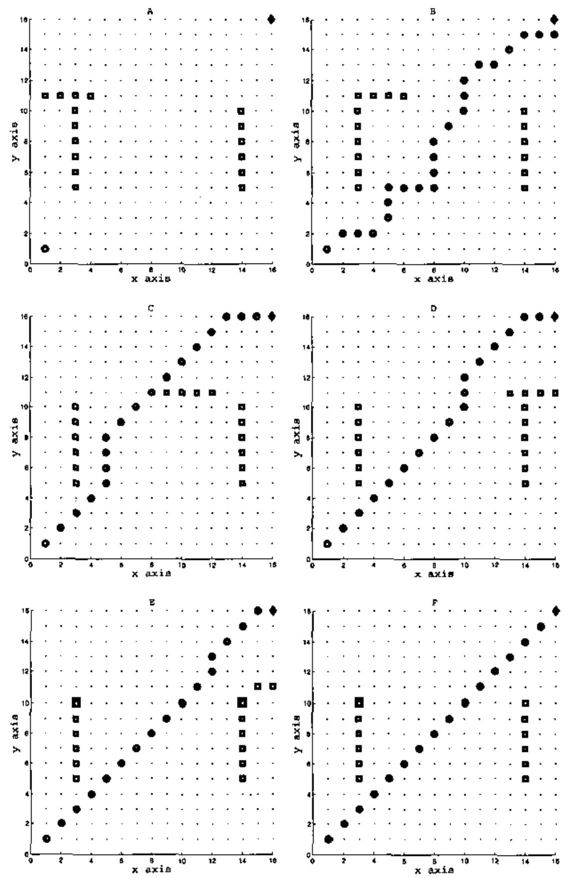

Figure 6: Path planning in a dynamic environment with static and dynamic obstacles. A: The robot path at the 0th generation; B: The generated path after the 10th generation; C: The path after the 40th generation; D: The path after the 60th generation; E: The path after the 70th generation; F: The optimal robot path after the 98th generation.

3.3 Discussion

If a fixed-length chromosome (model B) is used in the first and second cases, the length of chromosome is assumed as 18 genes. We analyze these two cases. In the first case, using model B, the length of the solution one is 18 genes, it is 3 genes more than the proposed model; in the second case, using model B, the length of the solution one is not enough (2 genes less) to reach the target, but the proposed model can do it (20 genes). Therefore, the variable-length chromosome is more suitable than the fixed-length chromosome in dynamic environment. And the generated robot path using the proposed model is shorter than the model B.

4 Conclusion

A genetic algorithm with a variable length chromosome has been developed to generate an optimal or near-optimal obstacle-avoidance path for a mobile robot using the proposed model in both static and dynamic environment. In the static environment, the generated robot path is optimal by a mathematic view (see Fig. 5.3B and 5.5B); in the dynamic environment, the generated robot path is near-optimal (see Fig. 5.6D and 5.6E). The proposal algorithm uses an efficient coding scheme that a variable-length binary string (solution) is used. From above discussion, the proposed model is more suitable than the fixed-length model and the generated robot path using proposed model is shorter than the fixed-length model.

References

[1] H. Chung, L. Ojeda, and J. Borenstein, "Accurate Mobile Robot Dead-reckoning with Precision-calibrated Fiber-optics Gyroscope.," *IEEE Transaction on Robotics and Automation*, Vol. 17, No. 1, February 2001, pp. 80-84

[2] K. Park, H. Chung, and J. Lee, "Dead Reckoning Navigation for Autonomous Mobile Robots," *Proceedings of Intelligent Autonomous Vehicle Madrid*, Spain, March 25-28, 1998, pp. 775-781.

[3] J. Kim and P. Khosla, "Real-time obstacle avoidance using harmonic potential functions," *IEEE Transactions on Robotics and Automation*, 8(3), pp. 338-349, 1992.

[4] J. Xu and K. Kin, "Intelligent mobile robot path planning with fuzzy system approach," *Proceedings of the 2nd IEEE International Workshop on Emerging Technologies and Factory Automation*, Australia, 1993, pp. 28-41.

[5] L. Deist and C. Fourie, "Fuzzy algorithm for the control of a mobile robot," *Proceedings of the 2nd IEEE International Workshop on Emerging Technologies and Factory Automation*, Australia, 1993, pp. 42-50.

[6] D. Goldberg, *Genetic Algorithms in Search Optimisation and Machine Learning*, Addison-Wesley, 1989

[7] H. Holland, *Adaptation in Natural and Artificial Systems*, 2nd Edition, London, MIT Press, 1993

[8] Koza R. *Genetic Programming On the Programming of Computers by Means of Natural Selection*, England: MIT Press, 1993.

[9] D. Cliff, P. Husbands and I. Harvey, "Evolving visually guided robots," *In Technical Report CSRP 220.* School of Cognitive and Computing Science, University of Sussex, 1992.

[10] J. R. Koza, *Genetic Programming*, London, MIT Press, 1992.

[11] A. Murray and S. J. Louis, "Design strategies for evolutionary robots," *Proceedings of the Third Golden West International Conference on Intelligent Systems*, 1995, pages 609-616. Kluwer Academic Press.

[12] T. Shibata, T. Fukuda, K. Kosuge and F. Arai, "Selfish and Coordinative Planning For Multiple Mobile Robots By Genetic Algorithm," *Proceeding of the 31st Conference on Decision and Control*, 1992, pp. 2686-2691.

[13] N. Baba and N. Kubota, "Collision Avoidance Planning of a Robot Manipulator by Using Genetic Algorithm," *Proceeding of the 1st Conference on Evolutive Computation*, 1994, pp. 714-719.

[14] Y. Davidor, *A Genetic Algorithm Applied to Robot Trajectory Generation*, Handbook of Genetic Algorithms, pp. 144-165, 1991.

[15] M. Habib and H. Asama, "Efficient Method to Generate Collision Free Paths for Autonomous Mobile Robot Based on New Free Space Structuring Approach", *Proceeding of IEEE/RSJ International Workshop on Intelligent Robotics and Systems*, 1991, pp. 563-567.