4th WSEAS Int. Conf. on COMPUTATIONAL INTELLIGENCE, MAN-MACHINE SYSTEMS and CYBERNETICS Miami, Florida, USA, November 17-19, 2005 (pp192-197)

Navigation of Autonomous Robots Using Genetic Algorithms

TERRENCE P. FRIES
Department of Computer Science
Coastal Carolina University
Conway, South Carolina
USA

Abstract: **-** Optimal motion planning is critical for the successful operation of an autonomous mobile robot. Many proposed approaches use either fuzzy logic or genetic algorithms (GAs), however, most approaches offer only path planning or only trajectory planning, but not both. In addition, few approaches attempt to address the impact of varying terrain conditions on the optimal path. This paper presents a fuzzy-genetic approach that provides both path and trajectory planning, and has the advantage of considering diverse terrain conditions when determining the optimal path. The terrain conditions are modeled using fuzzy linguistic variables to allow for the imprecision and uncertainty of the terrain data. Although a number of methods have been proposed using GAs, few are appropriate for a dynamic environment or provide response in real-time. The method proposed in this paper is robust, allowing the robot to adapt to dynamic conditions in the environment.

Key-Words: **-** Motion planning, navigation, robotics, genetic algorithms, fuzzy sets

1 Introduction

Optimal motion planning is essential to the successful operation of an autonomous mobile robot. Motion planning is composed of two functions: path planning, and trajectory planning [1, 2]. Path planning generates a collision-free path through an environment containing obstacles. The path is optimal with respect to some selected criterion. Trajectory planning schedules the movements of the robot along the planned path.

Many approaches to motion planning have been proposed. However, most approaches address only path planning or only trajectory planning, but not both [1, 3-5]. The GA coding scheme used in this research combines path planning with trajectory planning, thus, eliminating the additional step of trajectory planning once an optimal path is found and reducing the computational time to allow a real-time response.

It is common for GA-based approaches to motion planning to function only in a static environment due to the processing time required to produce an optimal solution [1, 3, 6-8]. However, many applications require that the robot respond to a changing environment and moving obstacles. This research provides a method that allows the robot to function in a dynamic environment.

In most cases, GAs do not provide real-time solutions to motion planning problems [1, 3, 6-8]. Those that do offer real-time response usually have unacceptable restrictions, such as limiting solutions to x-monotone or y-monotone paths [9]. An x-monotone path is one in which the projection of the path on the x-axis is non-decreasing. This places an unacceptable restriction on the solution path because even a simple path between two rooms in a building is neither x-monotone nor y-monotone as shown in Figure 1.

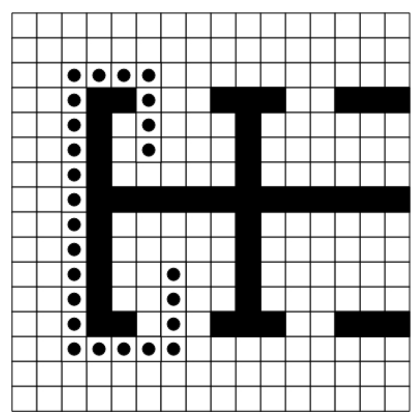


Fig. 1 Non-monotone path between rooms

In an effort to reduce the computation time, some researchers have proposed encoding all chromosomes with a fixed length [9, 10]. However, it has been shown that for robot path planning fixed length chromosomes are too restrictive on the solution path by placing unnecessary constraints on the representation of the environment and on the path [6, 10].

Research into using genetic algorithms for path planning include the work of Shibata and Fukuda [11] who proposed a motion planning strategy for a point robot in a static environment. Davidor [12] proposed GA approach attempts to minimize the accumulated deviation between the actual and desired path. However, this assumes that a desired path is already known. Nearchou [13] presented an approach using GAs that compares favourably with other evolutionary techniques, but it requires that the map be converted to a graph. None of these approaches account for dynamic conditions.

A further restriction among current motion planning approaches is that few approaches consider varying terrain conditions with most labeling an area either free of obstacles or totally blocked [8, 10]. In many real world cases, an area may be composed of terrain that is difficult to traverse. Difficult terrain may include sandy areas which cause slippage, rocky areas that require minor course adjustments within them and/or loss of time, or sloped areas that may cause slippage or increased time to climb. Such terrain may be traversable at the cost of increased time, but provide a more optimal path than totally clear terrain. This paper proposes an approach to motion planning that provides real-time motion planning in a dynamic environment without the restrictions of monotone paths or fixed length chromosomes. It also allows terrain to be labeled with the difficulty of traversal, thus, allowing it to be considered as a solution path.

Section 2 presents the representation of the environment and GA basics. In Section 3, the new fuzzy genetic motion planning approach is presented. Section 4 provides a discussion of the implementation and test results of the new motion planning approach.

2 Problem Formulation

2.1 Environment Grid

The environment in which the robot will maneuver is divided into an environment grid and a path is described as a movement through a series of adjacent cells in the grid. The length of the path $d(a, b)$ between two adjacent cells a and b is defined as the Euclidean distance between the centers of the two cells. This representation of distance allows the map data to be stored in any efficient format, such as a quadtree. Storage by such methods provides more compact representation of an environment by storing large obstacles as a single grid location, rather than many uniformly sized small squares. It also allows the path to be represented by fewer grid transitions, thus, reducing the size of the GA encoding string, or chromosome, and the time required to determine a solution. Each cell in the grid is assigned a fuzzy value that indicates the difficulty in traversing the terrain in that cell. The use of fuzzy values allows cells with moderately hostile terrain, such as rocks or loose sand, to be considered in a possible solution path while being weighted by their difficulty of traversal. A cell which contains an obstacle is assigned a fuzzy value indicating it is impassable and any possible solution path containing it is unacceptable. For this paper, the grid will be restricted to 16 by 16 for simplicity, however, the algorithm has been successfully tested for much larger sized grids. Further discussion of this restriction and actual testing is found in the Test Results section of this paper.

For purposes of this research, the robot is considered to be a holonomic point, that is, it is able to turn within its own radius. Because the robot is holonomic, a path can change direction within a cell and does not require a large arc for turning. Since it is a point, when traversing between two diagonally adjacent cells, it is not necessary to consider the other cells sharing the common corner as shown in Figure 2. This is not as impractical as it may appear at first glance. All real obstacles are expanded by half the radius of the robot when marking which cells are obstructed, thus allowing the robot to be treated as a point. This permits navigation of the center of the robot along the side of an obstacle or diagonally between obstacles. In Figure 2, the actual obstacle is solid and the expansion is shaded.

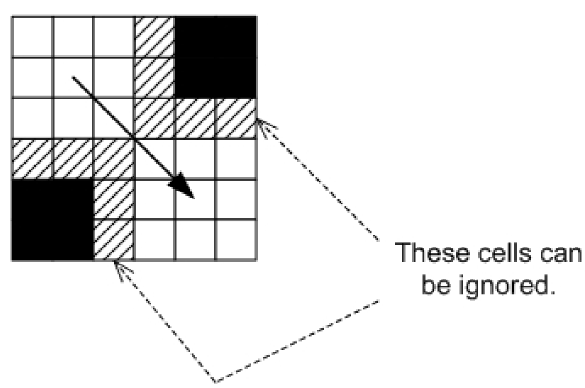


Fig. 2 Diagonal traversal of cells.

2.2 Genetic Algorithms

A genetic algorithm [14, 15] for optimization commonly represents a possible solution as a binary string, called a chromosome. Numerous approaches have been proposed for encoding paths as binary strings.

The GA begins with an initial population of chromosomes, or possible solutions. The GA then creates new individuals using methods analogous to biological evolution. The fitness of each chromosome is calculated using a fitness function. The criteria for evaluation is domain specific information about the relative merit of the chromosome. For example, in the case of path planning, the fitness function may calculate the time required or distance traveled to move from the initial location to the goal. The fittest parents are chosen to reproduce to create offspring. The offspring are generated by subjecting the parent chromosomes to various genetic operators including crossover and mutation.

The crossover operator combines parts of two different chromosomes to create two new ones. In single point crossover, the left part of a chromosome is combined with the right part of another, and then the remaining two parts of the originals are combined, thus, creating two offspring. The crossover point is usually randomly selected, although it can be fixed. Multiple point crossover divides the chromosome into multiple strings which are recombined with those of another chromosome.

The mutation operator changes the value of one random position in the chromosome. The offspring produced is identical to the parent except at the mutation point. For chromosomes represented as bit strings, this can be considered as inverting a single bit.

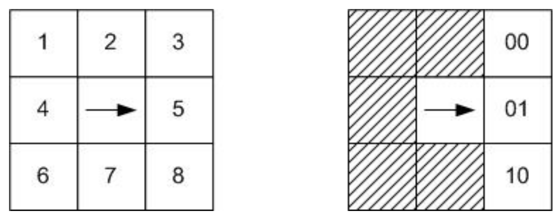
The most fit offspring replace the parents with the poorest fitness and the process continues until the population converges to a solution indicated by exceeding a fixed number of generations or until a chromosome attains a certain fitness value.

3 Motion Planning Algorithm

Several components can significantly affect the performance of a genetic algorithm: encoding of the chromosome, initial population, genetic operators and their control parameters, and fitness function.

3.1 Encoding the Chromosome

The first step is to choose a coding scheme which maps the path into a binary string or chromosome. Emphasis is placed on minimizing the length of the binary string. Minimizing the length of the chromosome reduces the number of generations necessary to produce an acceptable solution because less permutations are possible. A variable length string composed of blocks which encode the direction of movement and the length of the movement was chosen. Consider the robot in the center cell as in Figure 3 (a) having just arrived from cell 4 and facing in the direction of the arrow. There are eight possible directions for movement. However, cell 4 can be eliminated from consideration for the next move since the robot came from that cell and returning to it would create a non-optimal path. Cells 1, 2, 6, and 7 can be eliminated because they could have been reached from cell 4 using a shorter distance than through the center cell in which the robot currently is positioned. Only three cells remain in consideration for possible movement. The three cells require only 2 bits to encode as in Figure 3 (b).



(a) (b)

Fig. 3 Possible movement to next cell.

The largest number of cells that can be traversed in a square grid is found by starting in a corner and moving as far as possible along a side or the diagonal. Since the grid is constrained to 16 by 16 cells, the maximum number of cells that can be traversed in a single move is 15 which requires 3 bits to encode. As a result, each movement can be encoded in a 5-bit block as shown in Figure 4. For larger n x n grids, the block size would be $2 + \log_2 n$. A chromosome composed of these 5-bit blocks contains not only the path, but also the necessary trajectory information for movement of the robot. Thus, this unique encoding provides both path planning and trajectory planning.

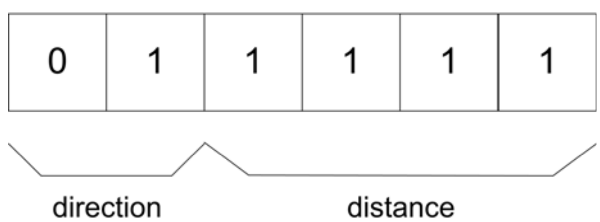


Fig. 4 Block encoding of one movement

3.2 Initial Population

The motion planning approach begins by randomly generating an initial population of chromosomes. In an effort to direct the solution to the shortest path, another chromosome is added to the initial population. It represents a straight line from the start to destination regardless of obstacles. If a straight line is not possible due to grid locations, the closest approximation to a straight line path is used. Through the testing of various combinations of variables, it was found that a population size, $p = 40$, was sufficient to seed the chromosome base.

3.3 Genetic Operators and Parameters

The algorithm used single point crossover. The crossover rate, γ, which is the percentage of parent chromosomes involved in the crossover, was selected as 0.8. The mutation rate, μ, or probability that a particular bit in the string is inverted, was 0.02. These parameters were arrived at through experimentation.

3.4 Fitness Function

Selection of a fitness function is a critical aspect of this research. Chromosomes are selected for reproduction through crossover and mutation based on the fitness function. The value provided by the fitness function is then used to retain the best members of the population for the next generation. Common approaches to using GAs for path planning set the fitness to an unacceptable value for any chromosome whose path traverses a grid cell with an obstacle in it. Otherwise, the fitness is based upon the distance traveled in the path. However, this does not account for terrain conditions. In an effort to consider adverse terrain conditions, each cell is assigned a value corresponding to the difficulty in traversing its terrain. The difficulty in traversing a particular terrain is imprecise because it may vary from one instance to another. In addition, it is problematical to compare different terrain conditions because of the varied nature of each. Further difficulty in a assigning a precise terrain difficulty exists because traversal of an cell in different directions can have significantly different difficulty levels. For example, traversing a sandy hill moving downhill, uphill, or across the side of the hill have dissimilar difficulty levels. Because of the imprecision of terrain conditions and the problems in directly comparing them, this research has chosen to express the terrain difficulty as fuzzy numbers. The terrain condition for each cell is expressed as a triangular fuzzy number using the linguistic variables shown in Figure 5. Terrain conditions represent the difficulty in traversing the cell which can be affected by conditions such as slope, sand, rocks, etc. As a result, the fitness function must be expanded for this research. For any path not passing through an obstacle, the fitness function uses the Euclidean distance between the centers of the cells traversed weighted by the terrain conditions for each cell

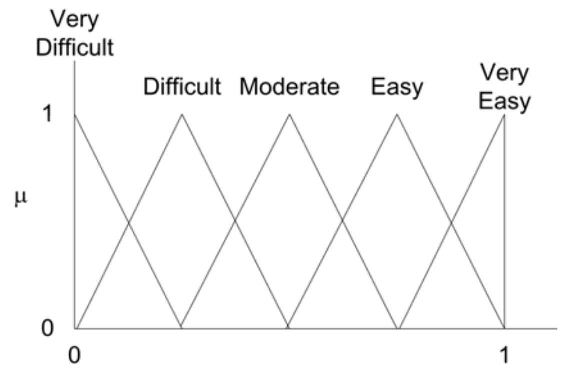


Fig. 5 Fuzzy representation of terrain conditions

3.5 Dynamic Environment

The fuzzy genetic motion planning method allows the robot to function in a dynamic environment. If an obstacle is detected by the robot where it not expected, the planner simply recalculates a new optimal path in real-time and the robot can continue its movement.

4 Test Results

The test software was implemented using C++ and Saphira robot control software. It was tested first in the Saphira simulator and, then, on a Pioneer 2-DX mobile robot. The Pioneer 2-DX is a holonomic 3-wheeled robot with a 250 mm radius. It is equipped with a suite of eight sonar sensors arranged as shown in Figure 6 and tactile bumpers. A predefined map representing the environment as a grid was provided to the robot. For clarity in presenting the test results, all results are shown for a 16 by 16 grid. This allows the demonstration of the algorithm's functionality while maintaining readability of the images. Testing has also been conducted using much larger grids and octree representations of the environment.

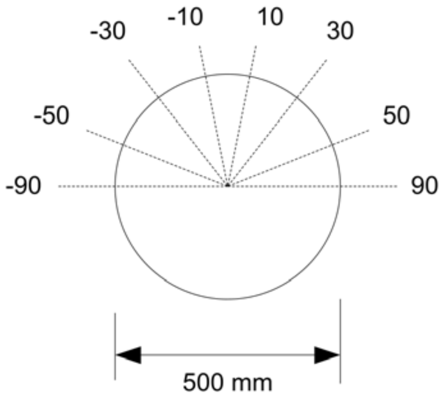


Fig. 6 Sonar sensor suite on Pioneer 2-DX robot

Figure 7 shows the path generated by the fuzzy GA method for a particular environment with no cell labeled as difficult terrain. The *S* and *D* indicate the start and destination cells, respectively, of the robot and black cells indicate solid obstacles. Manual examination confirms that this is the optimal path. This solution required seven 5-bit blocks in the optimal solution chromosome, including one to turn the robot to a starting orientation before beginning movement.

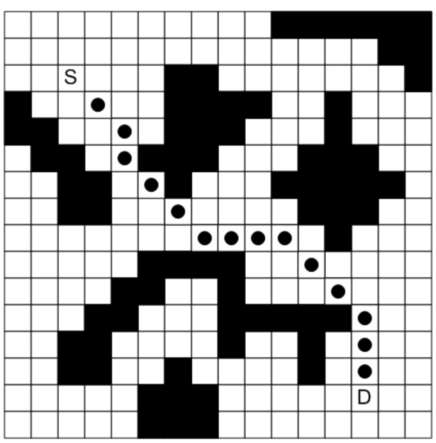


Fig. 7 Path generation with no terrain problems

Next the labeling of terrain difficulty with fuzzy values was verified. The shaded cells on the grid in Figure 8 were labeled as having *Moderate* difficulty to traverse. This had no effect on the generation of the optimal path as should be the case. However, when the same area was changed to *Difficult*, a different path was produced by the fitness function as shown in Figure 9. When the *Moderate* area was enlarged as in Figure 10, the fitness function again detected a more optimal path which avoided the larger *Moderate* terrain area.

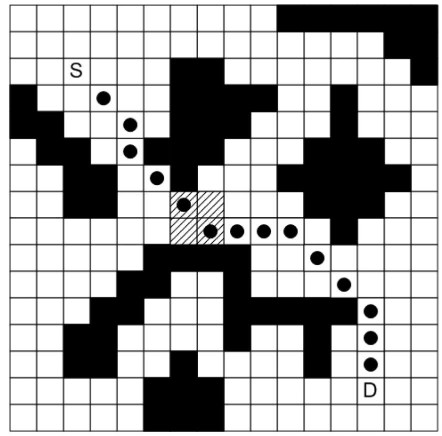


Fig. 8 Path with *Moderate* area of difficulty

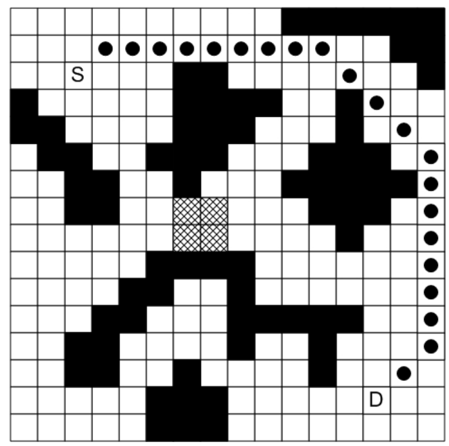


Fig. 9 Path with *Difficult* terrain area

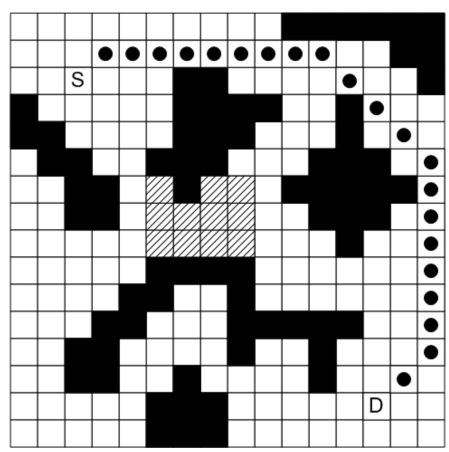


Fig. 10 Path with large area of *Moderate* difficulty

5 Conclusion

This research presents a fuzzy genetic algorithm approach to motion planning for an autonomous mobile robot that performs in real-time without the limitations of monotone paths. Varying terrain conditions are represented as fuzzy values and are included in the path planning decision. The encoding of the chromosome provides full motion planning capabilities and the method is capable of operation in a dynamic environment. Further research directions include the ability to observe and learn terrain conditions during movement along the path.

This research has provided an approach that is preferable to many traditional path planning algorithms, such as those using search algorithms, because it incorporates trajectory planning into the solution. Thus, once the optimal path is discovered, the trajectory information is immediately available for movement of the robot.

We have assumed perfect movement by the robot without accounting for drift and slippage. Additional research will incorporate localization to ensure the robot is on the planned path and provide necessary adjustments to the motion plan. This paper has presented the algorithm using a very simplistic 16 x 16 grid for purposes of demonstrating its functionality and clarity of the images. The approach has been successfully implemented using much larger grids and with octree representations of the environment. It has also been assumed that the terrain conditions are known a priori. Since this is not realistic in many applications, further research directions include the ability to observe and learn terrain conditions during movement along the path and to then adapt when more difficult terrain is discovered along the planned path.

References:

[1] Y. K. Hwang, & N. Ahuja, Gross motion planning – A survey, *ACM Computing Surveys*, Vol. 24, No. 3, Sept. 1992, pp. 219-291.

[2] J. C. Latombe, *Robot Motion Planning*, Kluwer, 1991.

[3] M.-R. Akbarzadeh, K. Kumbla, E. Tunstel, and M. Jamshidi, Soft computing for autonomous robotic systems, *Computers and Electrical Engineering*, Vol. 26, 2000, pp. 5-32.

[4] A. Stafylopatis and K. Blekas, Autonomous vehicle navigation using evolutionary reinforcement learning, *European Journal of Operational Research*, Vol. 108, No. 2, 1998, pp. 306-318.

[5] B.-T. Zhang and S.-H. Kim, An evolutionary method for active learning of mobile robot path planning, *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 1997, pp. 312-317.

[6] D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, IEEE Press, 2000.

[7] A. C. Nearchou, Adaptive navigation of autonomous vehicles using evolutionary algorithms, *Artificial Intelligence in Engineering*, Vol. 13, No. 2, 1999, pp. 159-173.

[8] D. K. Pratihar, K. Deb, and A. Ghosh, A genetic-fuzzy approach for mobile robot navigation among moving obstacles, *International Journal of Approximate Reasoning*, Vol. 20, No. 2, 1999, pp. 145-172.

[9] K. Sugihara and J. Smith, Genetic algorithms for adaptive motion planning of an autonomous mobile robot, *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 1997, pp. 138-143.

[10] O. Cordon, F. Gomide, F. Herrera, F. Hoffmann, and L. Magdalena, Ten years of genetic fuzzy systems: Current framework and new trends, *Fuzzy Sets and Systems*, Vol. 141, No. 1, 2004, pp. 5-31.

[11] Y. Davidor, *Genetic Algorithms and Robotics: A Heuristic Strategy for Optimization*, World Scientific, 1991.

[12] A. C. Nearchou, Path planning of a mobile robot using genetic heuristics, *Robotica*, Vol. 16, No. 5, 1998, pp. 575-588.

[13] T. Shibata and T. Fukuda, Intelligent motion planning by genetic algorithm with fuzzy critic, *Proceedings of the 8th IEEE Symposium on Intelligent Control*, 1993, pp. 565-569.

[14] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.

[15] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.