Memetic Algorithm Based Path Planning for a Mobile Robot

Neda Shahidi, Hadi Esmaeilzadeh, Marziye Abdollahi, Caro Lucas

***Abstract*—** In this paper, the problem of finding the optimal collision free path for a mobile robot, the path planning problem, is solved using an advanced evolutionary algorithm called memetic algorithm. What is new in this work is a novel representation of solutions for evolutionary algorithms that is efficient, simple and also compatible with memetic algorithm. The new representation makes it possible to solve the problem with a small population and in a few generations. It also makes the genetic operator simple and allows using an efficient local search operator within the evolutionary algorithm. The proposed algorithm is applied to two instances of path planning problem and the results are available.

***Keywords*—** Path planning problem, Memetic Algorithm, Representation.

I. Introduction

The problem of finding optimal path between two points in a known and static environment with different walls or obstacles for motion of a mobile robot is considered as the problem of mobile robot path planning. The path is highly desirable to be optimal or near optimal with respect to time, distance or energy while it is collision free. Distance is a commonly adopted criterion [5]. Path planning is usually carried out offline and considers existing knowledge about environment [3]. The best path is defined to be the path with the lowest cost which assumes the shortest collision free path in this paper and majority of similar works mentioned above.

There have been some efforts for solving this problem using evolutionary algorithms. One of the main challenges when using an evolutionary algorithm for solving a real problem is to representing the problem at hand using evolutionary algorithm fundamentals. Candidate solutions should be coded as chromosomes and well defined genetic operations as well as a suitable penalty function should be designed.

N. Shahidi is graduated in electrical engineering from University of Tehran, Tehran Iran. (Corresponding author to provide phone: 98212351840, ; fax: 9821633029; e-mail: n.shahidi@ ece.ut.ac.ir).

H. Esmaeilzadeh is graduated in computer engineering from University of Tehran, Tehran Iran. (Corresponding author to provide phone: 98212351840, ; fax: 9821633029; e-mail: hadi@ cad.ece.ut.ac.ir).

M. Abdollahi is graduated in electrical engineering from University of Tehran, Tehran Iran. (Corresponding author to provide phone: 98212351840, ; fax: 9821633029; e-mail: marziye@abdollahi.org).

C. Lucas is with the Center of Excellence for Control and Intelligent Processing, ECE Dept., Univ. of Tehran; and School of Intelligent Systems, I.P.M.,Tehran, Iran, (Corresponding author to provide fax: 9821633029; e-mail: lucas@ipm.ir).

In earlier works like [1] and [2], the path is a set of consequent points in a grid and the chromosome is a fixed or variable length string of the distances or strongly connected locations. In [1], the cost of passing over each point is calculated based on repulsive potential field around the obstacles and an attractive potential field around the end point. In [3] each gene has a triple structure each triple consisting of an (x, y) position and an additional information ‘q’ for quality of terrain at that particular location. The q = 0 means a reachable position and q = 1 is a point inside an obstacle. A different approach is taken in [4] in which a gene specifies the next movement direction and distance. The main shortage of cited approaches is that they lead to some invalid solutions like paths that not reach to the end point. These solutions should be eliminated in each generation. Moreover the number of genes in a chromosome is large. In [5] these problems are not occur because the chromosome is a relatively small variable–length set of points in a grid those are connected consequently with straight piece lines. Some new operators are also used in [5] to modify solutions.

In this paper a novel genetic representation of path planning problem and a suitable local search operation is proposed. The approach that is taken in this paper for coding is more similar to [5] except that allow to be a sub-path with specifiable shape between two points instead of a straight line and this shape is encoded in gene too. In addition, the chromosome length is fixed in contrast to [5]. This constrain simplifies the genetic operators [2].

The evolutionary algorithm that is noticed is a fast and accurate one known as memetic algorithm [6,7,8,9,10].These algorithms allow chromosomes to improve (or grow up) throughout their life time. Memetic algorithms use local search methods to find local optimums i.e. a point with the best fitness value among its neighbor points. The memetic algorithm is faster and more accurate that a simple genetic algorithm for some reasons: first, local search methods can serve the genetic operators with solutions those are better in compare to randomly generated solutions. Moreover, genetic algorithms are not good hill-climbers and the combination of them with local search methods alleviates this problem [11].

The proposed algorithm is explained in the next section and the experimental results of applying this algorithm to some instances of path planning problem are available in section III.

II. Memetic Path Planner

In this section, proposed memetic algorithm for solving the path planning problem is described. The proposed representation is described in the first subsection. The set of operations including crossover, mutation, local search, and selection are described in the second subsection.

A. Structure of Chromosome and Gene

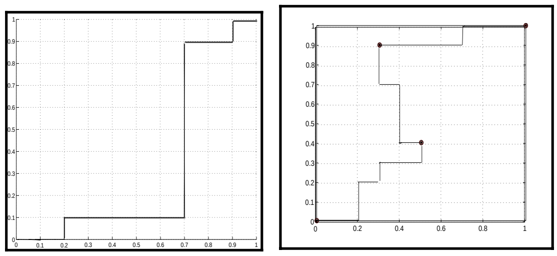
In this paper, a novel representation is proposed, which has the following advantages.

- The number of genes for a chromosome is small and fixed in contrast to variable-length representation that is used in some previous efforts. Moreover, it is independent from the size of the environment and the resolution of grid. These characteristics simplify the genetic operations.
- Chromosomes that represent invalid paths are not produced (neither randomly nor as a product of genetic operators). By definition, an invalid path may not reach the end point, that may be occur in some chromosome coding proposed in some mentioned efforts, or the path can not be recovered from information encoded in the corresponding chromosome.

In the proposed representation, each path is composed of a set of adjacent sub-paths. Each sub-path is represented by a gene and the entire path by a chromosome. In this representation, it is mandatory that all paths have the same number of sub-paths; therefore, the number of genes (the length of chromosome) is fixed. Each sub-path is represented by its start point, end point and a binary string that determines its shape. The start and end points of sub-paths can be anywhere in the grid, provided that the end point of each sub-path is the start point of the next sub-path. In the binary string that represent the shape, ‘0’s indicate one unit of vertical movement in the grid and ‘1’s indicate one unit of horizontal movement. Based on the relative position of the start and end points of a sub-path, only one direction for vertical movement (either upward or downward) and one direction for horizontal movement (either left or right) is allowed in each sub-path. In fact, a sub-path is a set of horizontal and vertical movements toward its end point and no backward movement is allowed in sub-paths. As an example, Fig. 1a shows a sub-path with the start point at (0, 0), the end point at (1, 1) and the binary string ‘11011111000000001101’. In this sub-path the end point is on the upper right of the start point; therefore, each ‘0’ represents an upward movement and each ‘1’ represents a right movement. It is obvious that the number of horizontal and vertical movements is determined by the horizontal and vertical distance of the start and end point respectively and is independent of the shape of the sub-path. Therefore, for each pair of start and end points, the valid values for the binary string are the permutations of a fixed number of zeros and ones. Fig. 1b depicts a path with 3 sub-paths and Fig. 2 shows the proposed chromosome structure.

B. Cost Function

As mentioned above, we assume that the robot environment is a known and static terrain and it is assumed that only topological information of the terrain is available. Therefore, the cost of passing over each point is known and we can calculate the cost of each path by adding the costs of its sub-paths. The cost function associated with a two-dimensional terrain should obtain higher values for longer paths. Moreover, it could separate a path that goes over a wall from a path that does not. In this paper, we use cost function that assigns a cost of ‘1’ to each point on the floor and a cost value that is greater in orders of magnitude (i.e. 100000) to the points placed on a wall. Therefore a path has a much higher cost value if it passes over at least on wall. Otherwise, it has a cost value proportional to its length. The fitness value of each chromosome is the cost of its corresponding path multiplied by -1.



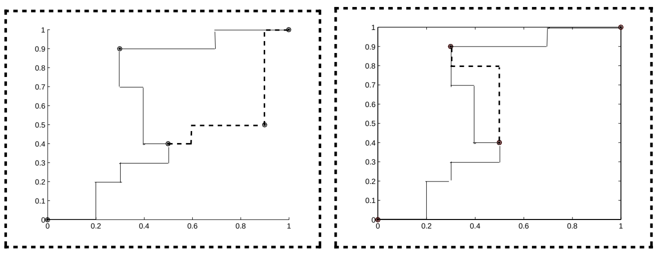
(a) **(b)**

Fig. 1 (a) A sample sub-path. (b) A sample path with three sub-paths.

C. Crossover and Mutation Operators

In mutation operation proposed in this paper, a gene is mutated through changing its binary string, start point or end point. The binary string is simply mutated by exchanging a ‘0’ and a ‘1’. But if the start or end point is changed, of course the end point of the previous gene or the start point of the next gene will be changed respectively to provide connectivity of path and both mutated gene and its neighbor gene will get a new binary string. Two examples of mutation are depicted in Fig. 3a and Fig. 3b.

Two-point crossover operation is also used. In this operation, a number of sub-paths are exchanged between two paths. Some modification should be down in exchanged sub-paths: the start or end point of some sub-paths should be changed to provide connectivity of sub-paths. In addition, the binary string of the sub-paths that their start or end points have been changed should be modified. An example of this operation is shown in Fig. 4. In this example the first two sub-paths are exchanged between two paths. The end point and the shape of the second sub-path are modified.



(a) **(b)**

Fig 3 (a) Mutation of the second sub-path’s end point. (b) Mutation of the second sub-path’s binary string.

Start Point	Shape	End Point	…	Start Point	Shape	End Point
Gene 1(sub-path 1)			…	Gene n (sub-path n)		

Fig. 2 Chromosome structure.

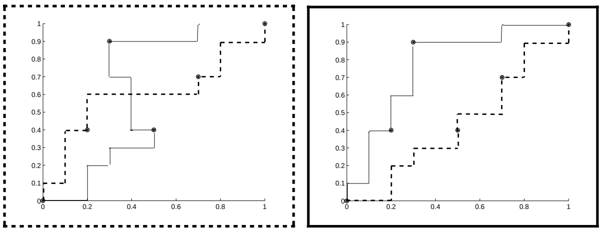


Fig 4 The crossover operator. (a) Two paths before the crossover. (b) Two paths after exchanging two first sub-paths.

A. Local Search Operator

Among various types of search methods that explore a limited neighborhood of a local optimum, called local search methods, which of them that use gradient information as well as value information are generally more efficient. But gradient information obtained through considerable amount of calculation that depends on the dimension of the search space.

The dimension of the search space is equal to the number of genes in the problem at hand. Therefore the amount of calculation is reduced significantly when the proposed representation is used rather than the previous representations. The gradient-based local search method used in this paper reduces the penalty of the chromosome through modifying the start and end points of genes based on the gradient information of the penalty function. This method is generally known as Gradient Ascent [12]. A typical path that is modified with this method is depicted in Fig. 5. It is seen that the path found using the local search has a shorter length, hence a lower penalty.

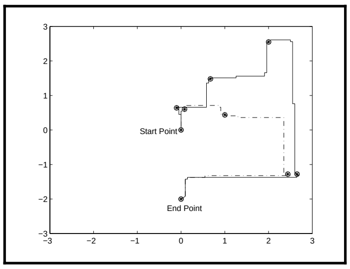


Fig 5 Local Search operator. The path before the hill-climbing (solid) and after it (dashed).

III. Experimental Results

The proposed algorithm is implemented and applied to two moderately difficult instances of the path planning problem. The complexity of these two cases is discussed in the sub-section A and the experimental results are explained in B.

A. Complexity Analysis

In this subsection, the relationship between the number of sub-paths and the complexity of problem is discussed. The number of sub-paths must be more than a specified number for an instance of path planning problem. For the first instance, depicted in Fig. 6, the number of sub-paths in each path cannot be less than three. Therefore at least two points (intermediate points) should be placed in the terrain. The path has a chance to pass over no wall if both of these two points are placed in the shadowed areas of Fig. 6. For the second instance, depicted in Fig. 7, the number of sub-paths in each path should be at least two. If two is chosen, then finding a path is equivalent to finding one intermediate point. If that point resides outside the shadowed area in Fig. 7, the resulting path passes over at least one wall. The smaller the shadowed area, the more difficult the search space becomes. In fact, with a small shadowed area, the global minimum of penalty function, that is associated with the best solution, places in a narrower valley of the cost function and has less chance of discovery by candidate solutions. The difficulty of the problem could be decreased by increasing the number of sub-paths appropriately. In fact more freedom is given to the path when the number of sub-paths increases. But unnecessary large number of sub-paths, means high-dimensional search space, leads to more difficult search space and also increases the amount of necessary computation (i.e. for function evaluation and gradient calculation).

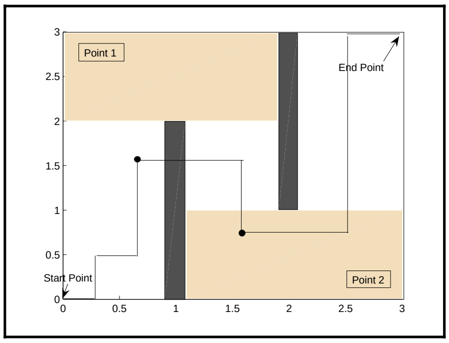


Fig 6 Shadowed area for first problem. It can be seen that the path passes a wall if one of the sub-paths end points locate placed outside of shadowed area.

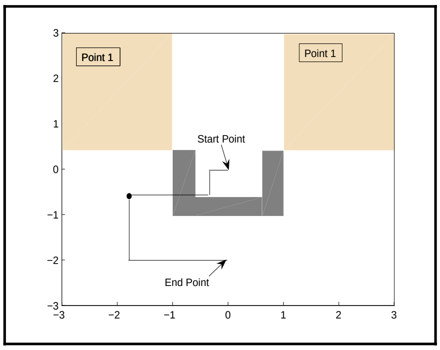


Fig 7 Shadowed area for first problem. It can be seen that the path passes a wall if even one of the sub-paths end-points locate outside of shadowed area.

B. Experiments and Their Results

The proposed MA is evaluated using two instances of path planning problem mentioned above. Figures 8 and 9 display the best solution found by proposed MA and GA (exactly like MA but without local search operator). Each experiment is done for several time and the results are averaged. It is obvious that the optimal or near optimal solution can be found using the proposed algorithm. The population size and the number of generations for two instance problem are in Table I. Referring to this table, it can easily be seen that the algorithm find solution after few generations while use relatively small population. As mentioned above no invalid solution produced using the proposed representation hence these relatively small populations can easily find the best solution.

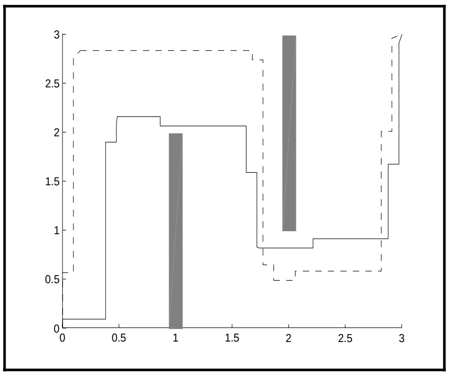


Fig 8 The path found by MA (solid) and GA (dashed).

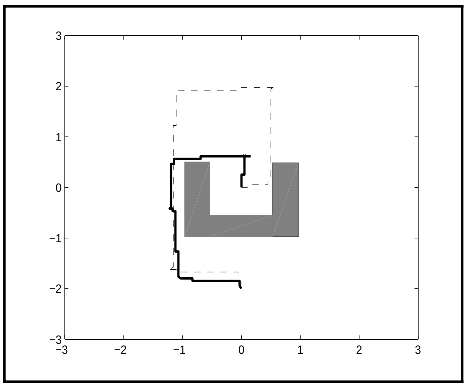


Fig 9 The path found by MA (solid) and GA (dashed).

TABLE I ALGORITHM SETTINGS AND EXPERIMENTAL RESULTS.

Instance problem	Num. of Generations	Population Size
Instance 1 - GA	5	100
Instance 1 - MA	2	50
Instance 2 - GA	10	100
Instance 2 - MA	10	50

IV. CONCLUSIONS

In this paper, a novel representation for the path planning problem that was suitable for evolutionary algorithm especially memetic algorithm was proposed. A local search operator for tuning the start and end points of sub-paths was also proposed. The experimental results illustrate that in the path planning problem, the path found in a few generations with a relatively small population of chromosomes. The results also demonstrate that the solution found using a memetic algorithm is more optimal than that found by a simple genetic algorithm. Optimization of the shape of sub-paths using an appropriate local search method is our future step.

REFERENCES

[1] Ashiru I.: Czarnecki C.: Optimal Motion Planning for Mobile Robots Using Genetic Algorithms. In Proc. of the 1995 International Conference on Industrial Automation and Control (1995).
[2] Sugihara, K., Smith J.: Genetic Algorithms for Adaptive Motion Planning of an Autonomous Mobile Robot. In Proc. of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (1997) 138-146.
[3] Gerke M.: Genetic Path Planning for Mobile Robots. In Proc. of the American Control Conference (1999).
[4] Tu J., Yang S.X.: Genetic Algorithm Based Path Planning for a Mobile Robot. In Proc. of the 2003 IEEE Intern. Conference on Robotics & Automation (2003) 1221-1226.
[5] Hu Y., yang S.X.: Knowledge Based Genetic Algorithm for Path Planning of a Mobile Robot. In Proc. of the 2004 IEEE Intern. Conference on Robotics & Automation (2004) 4350-4355.
[6] Hart W. E.: Adaptive Global Optimization with Local Search. Ph. D. Thesis, University of California, San Diego (1994).
[7] Shahidi N., Esmaeilzadeh H. Abdollahi M., Lucas C.: Self-adaptive Memetic Algorithm: An Adaptive Conjugate gradient approach. IEEE Conference of Cybernetic and Intelligent Systems (CIS'2004), in press.
[8] Baldwin J. M.: A New Factor in Evolution. The American Naturalist 30 (1896), 441-451, 536-553.
[9] Ong Y. S., Keane A. J.: Meta-Lamarckian Learning in Memetic Algorithms, IEEE Transactions on Evolutionary Computation, Vol. 8, No. 2 (2004).
[10] Krasnogor N.: A Memetic Algorithm with Self-adaptive Local Search; TSP as a Case Study. In Proc. of the 2000 International Genetic and Evolutionary Computation Conference (GECCO 2000).
[11] Land, M. W. S.: Evolutionary Algorithms with Local Search for Combinatorial Optimization. Ph. D. Thesis, University of California, San Diego, 1998.
[12] Principe J. C., Euliano N. R., Lefebvre W. C.: Neural and Adaptive Systems. Jone Wiley & Sons, Inc., 2000.