

A Hybrid Genetic Algorithm for Mobile Robot Shortest Path Problem

Eşref Boğar*¹, Selami Beyhan²

Accepted 3rd September 2016

Abstract: This paper proposes an algorithm to solve the problem of shortest path planning for a mobile robot in a static environment with obstacles. The proposed algorithm is a Hybrid Genetic Algorithm (HGA) which includes Genetic and Dijkstra Algorithms together. The Genetic Algorithm (GA) is preferred since the structure of robot path planning problem is very convenient to apply genetic algorithm's coding and operators such as permutation coding, crossover and mutation. GA provides diversification while searching possible global solutions, but Dijkstra Algorithm (DA) makes more and more intensification in local solutions. The simulation results show that the mobile robot can plan a set of optimized path with an efficient algorithm.

Keywords: Robot path planning, genetic algorithm, Dijkstra algorithm, Shortest Path.

1. Introduction

Robot path planning (RPP) is a popular issue in mobile robotics. An intelligent robot must be able to move by itself from a start location to a target location without collision with obstacles. Therefore an important research subject in this field is navigation of autonomous mobile robots, which is to find a globally optimal path from a starting point to a target in a given environment and at the same time avoid collisions. Moreover, the "optimal" here means that the path must satisfy some criterions like length of the path is the shortest, or energy consumption of robot is the lowest or needed time to reach a target is minimum etc.

There are several search algorithms for the shortest path problem such as the A* Algorithm, the Dijkstra Algorithm and the Bellman-Ford Algorithm [1], [2], [3]. These algorithms ensure to find a shortest path in environment with obstacles. However, these algorithms are not effective for large dimensioned problems because of with low searching efficiency and high computational time.

In recent years, many researchers have studied the robot path planning on various intelligent methods, such as Particle Swarm Optimization [4], Artificial Immune System [5], Neural Networks and Ant Colony Optimization [6], Differential Evolution [7], and so on. Additionally, Genetic Algorithm is one of the popular metaheuristic algorithm for RPP [8] and [9] problem.

The GA was introduced by Holland [10] in the early 1970s. It imitates the evolutionary processes such as inheritance, mutation, selection and crossover. Also for centuries, there is a biological evolution in the world and GA simulates it by using iterative process. The GA has a wide using area in many types of problem thanks to easy adaptation and high performance computing.

In this paper, a Hybrid Genetic Algorithm (HGA) is proposed to solve the problem of robot path planning under shortest path objective. The aim is to combine the global search capability of Genetic Algorithm and local exact solution capability of the Dijkstra Algorithm to get faster solution for shortest path problem. In numerical simulations, 20x20 and 50x50 environments with obstacles are solved to get shortest distances and, Dijkstra, genetic and proposed hybrid genetic algorithms are compared in (Table.2) for iteration number and required time, respectively.

The paper is organized as follows. Section 2 describes the method of grid modeling, Dijkstra Algorithm and Genetic Algorithm. Section 3 describes the proposed hybrid genetic algorithm for the robot path planning problem. Section 4 presents the numerical simulations on robot path planning. Section 5 presents concluding remarks.

2. Preliminary Works

2.1. Grid modelling

The method was proposed by the Howden in 1968, and its primary task is to construct a path grid map according to the environment. Given environment is divided into small units and a robot goes one unit with one move. Two kind of grid can be described in method which are free and obstacle grid. In black grid's mean is obstacle, white grid's mean is free namely movable area. In real environment sizes of obstacles can be different and they can occupy lower one or more grids. A robot cannot enter an obstacle even if the obstacle is very small.

The grid can be divided into intermediate grid and boundary grid. For intermediate grid, robot may have eight directions for the next motion. Such as up (7), down (2), left (4), right (5), right-up (8), right-down (3), left-up (6) and left-down (1). (Figure.1) shows the motion direction of the robot and (Figure.2) shows a 10x10 grid map. And for boundary grid, it has to subtract inaccessible directions. Robot must avoid obstacles to select an optimization motion path moving to the target position [11].

¹ Electrical and Electronics Engineering, Muğla Sıtkı Koçman University, Muğla – 48000 Kötekli Campus, Turkey

² Electrical and Electronics Engineering, Pamukkale University, Denizli – 20020 Kinikli Campus, Turkey

* Corresponding Author: Email: esrefbogar@mu.edu.tr

Note: This paper has been presented at the 3rd International Conference on Advanced Technology & Sciences (ICAT'16) held in Konya (Turkey), September 01-03, 2016.

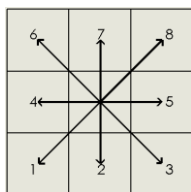


Figure 1. The motion direction of the robot.

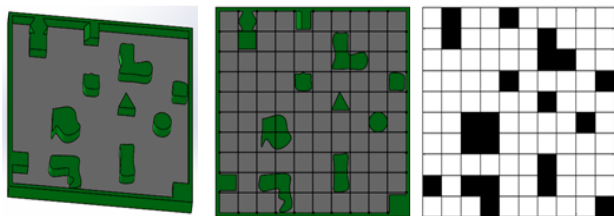


Figure 2. 10x10 grid map

2.2. Dijkstra Algorithm

Dijkstra algorithm is introduced by W. Dijkstra in 1959[1]. It is one of the most common used algorithm for solving the shortest path problem. It is possible to find a shortest path from starting point to any node by Dijkstra Algorithm. If the performance criterion is different from distance, it is also useful for measuring other criteria such as time, cost and energy. Dijkstra algorithm guarantees the shortest path which is generated. Dijkstra algorithm is a useful method for not only robot path planning but also network optimization, transportation, logistics, electronics and other fields. However, it takes much time while solving large problems and this is a disadvantage for this method.

2.3. Genetic Algorithm

Genetic algorithm combines genetics and computer science. GA is biology inspired, population based, has memory, stochastic and greedy algorithm.

Here are a few basic concepts of genetic algorithm:

Chromosome and Coding: In order to use genetic algorithm, feasible solution of the problem needs to be coded into symbol string which have fixed structure. The string is called chromosome. Each bit of the string represents a gene.

Population: the total number of chromosomes in each generation is called population. A population contains a set of feasible solutions in current generation.

Fitness: Each chromosome corresponds to a feasible solution, and each solution corresponds to a function value. This function value is used to measure the environmental adaptability of a chromosome.

Select: Select operation can select adaptable individual in current population, so that adaptable individual has the opportunity to breed the next generation as a parent.

Crossover: Crossover operation can produce new individual, the new individual combines the characteristics of parents.

Mutation: mutation operation can change one or a number of genes in a fixed probability. The purpose of mutation is to create new chromosomes for next generation.

The process of genetic algorithm: determine the encoding rules, generate a population randomly, calculate fitness function and selection probability, select, crossover, mutation, loop all the operation above until it meet the terminate conditions [12].

3. Proposed hybrid genetic algorithm

In this section, proposed algorithm steps are explained in detail.

3.1. Problem Description

A road from starting node (S) to target node T is an (S, T) path,

the length of a path is the sum of the length of the edges ($|p_i p_{i+1}|$) in one path. Robot path problem is to find the path from S to T, where the length of the path is minimum one at the same time avoid obstacles. A candidate path for robot path planning is denoted by

$$P = \{p_1, p_2, \dots, p_n\}$$

where p_i denotes the i th node of the planned path P , n denotes the number of without obstacle nodes in a given environment and $p_1 = S$ and $p_n = T$.

Total length of the path can be obtained by (Equation.1).

$$f(P) = \sum_{i=1}^{n-1} |p_i p_{i+1}| \quad (1)$$

where $|p_i p_{i+1}|$ is the length of the path segment $p_i p_{i+1}$ which is computed by Euclidean distance and it can take value $\sqrt{2}$ or 1. Additionally $f(P)$ is objective function of given path which we try to minimize.

3.2. Chromosome encoding and initial population generation

For a given graph, each node will be coded as a gene sorted by the node number. A chromosome includes path information and sequence of genes. In this paper, permutation coding is used for encoding. Nodes take place as combinatorics in a solution if a node is visited as an example shown below:

$$[1 \rightarrow 9 \rightarrow 16 \rightarrow 17 \rightarrow \dots \rightarrow 71 \rightarrow 79]$$

The first and end nodes must be in the solution, indicating the starting node S and target node T. There is no need to the test chromosome if the path is connected. Producing initial solution strategy guarantees that all the nodes which are included solution are connected. In other word, length of a chromosome is equal to number of passed nodes. Therefore, there is a remarkable decreasing usage of memory for a chromosome and computation time. Cross-over and mutation operators are modified because of different length of chromosome which are shown next subsections.

While generating initial solution, a node is selected (initial gene of solution is starting node) and other neighbors are found which are connected to the selected node. Randomly selected a neighbor node is added to solution. This procedure is repeated until reached target nodes. Additionally, generating strategy checks if a node is visited before. (Table.1) shows generating a chromosome for given map in (Figure.3).

Table 1. Generating a chromosome

Exist Solution	Candidate Nodes	Randomly Selected Nodes
1	2,9	9
1,9	2,3,15,16,17	15
1,9,15	16,24,25	25
...
1, 9, 15, 25, ..., 71	62, 79	79

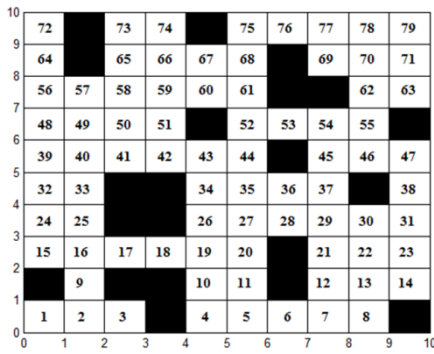


Figure 3. Grid map for initial solution procedure.

3.3. Fitness function

Fitness function must be calculated for each of chromosomes. The function is the sum of the length of the edges ($|p_i p_{i+1}|$) in a path which is given (Equation.1).

3.4. Select, Crossover and Mutation

Select: For a given population X and fitness of each chromosome is $f_i(X)$. The number of chromosome randomly selected in X is M and the best one is selected by using tournament selection for reproduction. In this work M is assumed 2 for the tournament selection.

Crossover: For the given chromosome x_1, x_2 which are parent, their intersected nodes ($intersect(x_1, x_2)$) are found except for starting and target nodes (1,79) and one of the found nodes is selected randomly. This node is called as crossover point and single point crossover is used. A crossover is done at the selected node.

For example:

$$x_1 = 1,2,9,17,18,19,27,36,44,52,61,68,76,77,78,79$$

$$x_2 = 1,9,16,25,33,41,42,43,44,53,54,62,71,79$$

$$(intersect(x_1, x_2)) = 9, 44 \quad (\text{except for } 1,79)$$

For randomly selected node which is 44. The new chromosomes (children):

$$x_1 = 1,2,9,17,18,19,27,36,44,53,54,62,71,79$$

$$x_2 = 1,9,16,25,33,41,42,43,44,52,61,68,76,77,78,79$$

Mutation: The parental chromosome is chosen according to selection method. The parents start and target nodes are not mutated. The mutation operation is done by selecting a random node in the parent according to mutation probability. The two nodes which are coming before and after selected nodes are detected. Mutation operator tries to find a different node to connect these two nodes. If these two nodes connect to each other directly, selected node is deleted from the solution. An example for mutation operator can be shown as

$$[1,2,9,17,18,19,27,36,44,53,54,62,71,79]$$

where selected node is 36, node 27 is coming before and node 44 is coming after node 36. Neighbor nodes are from 27 and 44 are:

$$27 \rightarrow 19,20,26,28,34,35,36$$

$$44 \rightarrow 34,35,36,43,52,53$$

where node 34 and 35 are common neighbors for nodes 27 and 44. Randomly selected one of these nodes is exchanged with node 36. The mutated chromosome is

$$[1,2,9,17,18,19,27,35,44,53,54,62,71,79].$$

3.5. Dijkstra Algorithm

In this algorithm, after standard Genetic algorithm steps are applied in each iteration, Dijkstra algorithm is conducted to all individuals of current generation between two points (randomly selected) which are close to each other. Thus, the proposed algorithm is able to find rapidly a nearly optimum path for a mobile robot without collides with static obstacles.

After generating initial population, selecting, crossover, mutation and Dijkstra algorithm are applied to each generation until termination criteria is achieved.

4. Numerical Simulations

Computer simulation environment includes a notebook computer with Intel® Core™ i7-4712 MQ CPU @ 2.30 GHz, 8 GB RAM. The planned path is shown in (Figure.4& 5).

The simulations are conducted with 20x20 and 50x50 environments with random obstacles. The 20x20 grid map example, which includes 400 nodes and 318 of total nodes are free and 82 nodes are obstacles. At the same time the 50x50 grid map example, which includes 2500 nodes and 2080 of total nodes are free and 420 nodes are obstacles. The resulting paths with HGA are shown in (Figure.4 & 5), respectively.

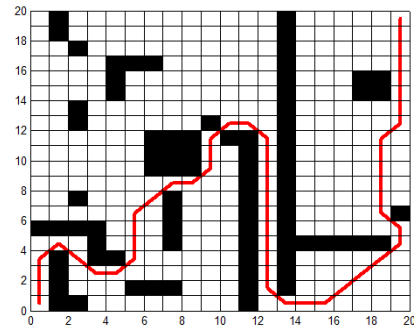


Figure 4. Optimized path with HGA for 20x20.

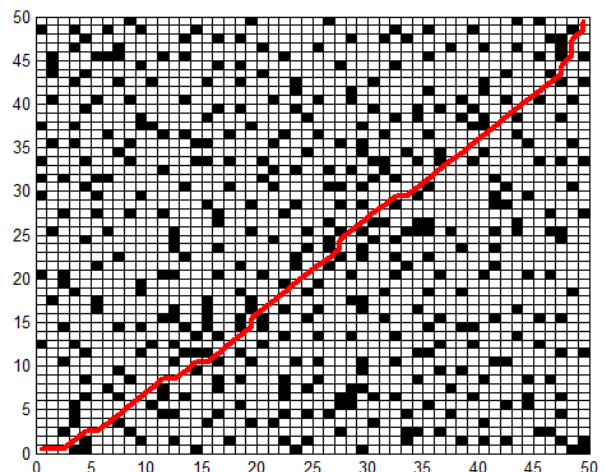


Figure 5. Optimized path with HGA for 50x50.

The performances of the algorithms are given in (Table.2). These algorithms produce same fitness value for 20x20 and 50x50 environments as 59.8701 and 72.8112, respectively. However, the iteration number and required time change with the algorithm therefore they are given in (Table.2). According to the performances, the proposed HGA is always better than the conventional genetic algorithm. However, for small size problem, the Dijkstra algorithm provides the solution in a shorter time. Note that when the problem size increased, the Dijkstra algorithm requires larger time to solve which is seen in given Table.

Table 2. Performances of the algorithms

Method/Performance	20x20		50x50	
	Iteration number	Time (sec)	Iteration number	Time (sec)
Dijkstra Alg.	-	0.352	-	17.104
Genetic Alg.	102	4.097	281	11.768
Proposed HGA	69	2.811	153	7.246

5. CONCLUSION

In this paper, RPP problem is solved with a novel Hybrid Genetic algorithm (HGA). Simulation results show that the proposed method provided the same solutions with lower iteration number and required time for 50x50 environment. It is seen that the proposed method is especially suitable for the large dimensional RPP problems in future applications. The algorithm is planned to be used in real-time with a laboratory mobile robot.

References

- [1] E. W. Dijkstra (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*. Vol.1. Pages. 269-271.
- [2] R. Bellman (1958). On a routing problem. *Quarterly Applied Mathematics*. Vol.16. Pages. 87 – 90.
- [3] P. E. Hart, N.J. Nilsson and B. Raphael, “A Formal Basis for the Heuristic Determination of Minimum Cost Paths”, *IEEE Transactions on Systems Science and Cybernetics*, vol.4, 2, pp. 100–107, 1968.
- [4] X. Chen and Y. M. Li (2006). Smooth path planning of a mobile robot using stochastic particle swarm optimization. In *Proceedings of IEEE International Conference on Mechatronics and Automation*. Pages. 1722–1727.
- [5] X. Z. Hu and Q. G. Xu (2007). Robot path planning based on artificial immune network. In *Proceedings of the 2007 IEEE International Conference on Robotics and Biomimetics*. Pages. 1053-1058.
- [6] Y. Huang (2012). Intelligent Technique for Robot Path planning Using Artificial Neural Network and Adaptive Ant Colony Optimization. *JCIT*. Vol.7. Pages. 246 - 252.
- [7] H. W. Mo and Z. Z. Li (2012). Biogeography based differential evolution for robot path planning. In *Proceedings of International Conference on Information and Automation*, Pages. 1- 6.
- [8] J. C. Mohanta, D. R. Parhi and S. K. Patel (2011). Path planning strategy for autonomous mobile robot navigation using Petri-GA optimization. *Computers & Electrical Engineering*. Vol.37. Pages. 1058-1070.
- [9] E. Bogar (2016). A Hybrid Optimization Method for Single

and Multi Objective Robot Path Planning Problem. Master's thesis. Pamukkale University Institute of Science and Technology.

- [10] J. Holland (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press.
- [11] J. Cao (2006). Robot Global Path Planning Based on an Improved Ant Colony Algorithm. *Journal of Computer and Communications*. Vol.4. Pages. 11-19.
- [12] B. K. Oleiwi, H. Roth and B. I. Kazem (2014). Modified Genetic Algorithm based on A* Algorithm of Multi Objective Optimization for Path Planning. *Journal of Automation and Control Engineering*. Vol.2. Pages. 357-362.