Computers and Electrical Engineering 38 (2012) 1564–1572

Contents lists available at SciVerse ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Dynamic path planning of mobile robots with improved genetic algorithm ☆

Adem Tuncer *, Mehmet Yildirim

Networked Control Systems Laboratory, Kocaeli University, Umuttepe 41380, Kocaeli, Turkey

ARTICLE INFO

Article history:
Received 25 October 2011
Received in revised form 8 June 2012
Accepted 11 June 2012
Available online 21 July 2012

ABSTRACT

In this study, a new mutation operator is proposed for the genetic algorithm (GA) and applied to the path planning problem of mobile robots in dynamic environments. Path planning for a mobile robot finds a feasible path from a starting node to a target node in an environment with obstacles. GA has been widely used to generate an optimal path by taking advantage of its strong optimization ability. While conventional random mutation operator in simple GA or some other improved mutation operators can cause infeasible paths, the proposed mutation operator does not and avoids premature convergence. In order to demonstrate the success of the proposed method, it is applied to two different dynamic environments and compared with previous improved GA studies in the literature. A GA with the proposed mutation operator finds the optimal path far too many times and converges more rapidly than the other methods do.

1. Introduction

Recently, interest of researchers on autonomous vehicles has increased with technological developments. There are many studies in the literature about autonomous vehicles. One of the main subjects in autonomous vehicle research is path planning. Path planning tries to find a feasible path for mobile robots to move along from a starting node to a target node in an environment with obstacles [1].

The path planning environment can be either static or dynamic. In the static environment, the whole solution must be found before starting execution. However, for the dynamic or partially observable environments re-plannings are required frequently and more planning update time is needed. Generally, the major problems for path planning of mobile robots are computational complexity, existence of local optima and adaptability. Researchers have always been searching alternative and more efficient ways to solve these problems [2].

There are many studies on robot path planning using various approaches, such as the grid-based A^* algorithm, road maps (Voronoi diagrams and visibility graphs), cell decomposition, and artificial potential field. Some of the previous approaches that use global methods to search the possible paths in the workspace, normally deal with static environments only, and are computationally expensive when the environment is complex [3]. Each method differs in its effectiveness depending on the type of application environment and each one of them has its own strength and weaknesses. For example, the artificial potential field method can give only one solution route that may not be the shortest path, even in a static environment. Ajmal Deen Ali et al. [4] investigated the effectiveness of genetic algorithms (GAs) on the study of collision free path planning of manipulators to reduce search time and improve the quality of the solutions. The results from this approach are found to be better than the conventional A^* search, both in the distance traveled and in computation time. Chen and Zalzala [5] compared the GA with the modified A^* method in mobile robot path planning. It was observed that although the modified A^*

☆ Reviews processed and approved for publication by Editor-in-Chief Dr. Manu Malek.

* Corresponding author. Tel.: +90 262 303 2269; fax: +90 262 303 2203.
E-mail addresses: adem.tuncer@kocaeli.edu.tr (A. Tuncer), myildirim@kocaeli.edu.tr (M. Yildirim).

0045-7906/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compeleceng.2012.06.016

method can obtain a solution quicker than the GA, the modified A* method can easily fall into some local minima, while the GAs can always reach the global optimum or a near global optimum.

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 prior knowledge about the problem to be solved [6]. One technique that outperforms the others in path planning is the GA method because of its capacity to explore the solution space while preserving the best solutions already found [7]. In the last decade, GAs have been widely used to generate the optimal path by taking the advantage of its strong optimization ability [8]. GAs have been recognized as one of the most robust search techniques for complex and ill-behaved objective functions. The basic characteristic that makes the GA attractive in developing near-optimal solutions is that they are inherently parallel search techniques [9]. They can search all working environment simultaneously in a parallel manner and so they can reach a better solution more quickly.

In recent years, lots of GA based path planning has been implemented by means of customizing genetic operators and most of these were at the same time improved approaches. They have succeeded in achieving better solutions. While conventional random mutation operator in simple GA can cause an infeasible path, the proposed mutation operator in this study increases the diversity of the population and avoids premature convergence.

This article is organized as follows: In Section 2, we explain how the GAs are applied to the path planning problems of mobile robots; Section 3 analyzes the mutation operators which were reported previously and proposed in this article; Section 4 evaluates the experimental studies and results; and in Section 5 we summarize the main findings of the article.

2. Path planning with genetic algorithm

GA is a parallel and global search technique that emulates natural genetic operators. Because it simultaneously evaluates many points in the parameter space, it is more likely to converge to the global optimal. It is not necessary that the search space to be differentiable or continuous [10–12].

2.1. Representation of environment

Many path planning methods use a grid-based model to represent the environment space as shown in Fig. 1. It has been determined that calculation of distance and representation of obstacle is easier with grid-based representation. The grid-based environment space is represented in two ways, by the way of an orderly numbered grid [1,13–15] or by the way of (x,y) coordinates plane [9,16,17]. It has been found that in the literature the orderly numbered grid representation is widely used, therefore this representation method is used in the present study.

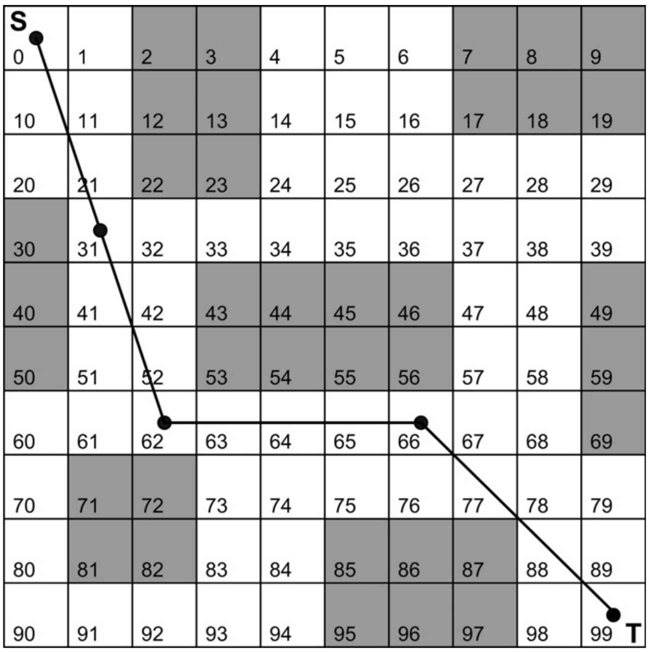

Fig. 1. Grid-based environment representation.

2.2. Encoding of chromosomes

A chromosome represents a candidate solution for the path planning problem. A chromosome or a path consists of a starting node, a target node and the nodes which mobile robot goes over them. These nodes or steps in the path are called as genes of the chromosome. Different coding methods are used to create chromosomes. Binary coded string method [9,18,19] is used in general, however decimal coded string method is also used [1,13,14] and it is considered to be more flexible. Decimal coding needs less memory and less space in optimization. As shown in Fig. 2, a valid path consists of a sequence of grid labels which begins from starting node and ends at the target node.

2.3. Initialization of population

The initial population is generally generated randomly. Some of the generated chromosomes may include infeasible paths which intersect an obstacle. An optimal or near optimal solution can be found by genetic operators, even though the initial population includes infeasible paths. However, this process reduces the search capability of the algorithm and increases the time to find an optimal solution. Furthermore, crossover of two infeasible chromosomes may generate new infeasible paths. In order to solve this problem, each chromosome must be checked whether it intersects an obstacle, when generating the initial population. If it does, the intersected gene on the chromosome is changed randomly with a feasible one.

In this study, GA is run with both randomly generated and feasible initial population. For the same environment and 10 runs, the average fitness value, number of generation and solution time of each method are given in Table 1. It is determined that starting the GA with the feasible initial population is fairly beneficial, as also stated in [13,14].

2.4. Fitness function

The purpose of the path planning problem is to find an optimal path between a starting and a target node. Optimal path may be the shortest, the least time and energy requiring path to trip on. Generally, in the path planning problems, the objective function is considered as the shortest path. In this study, the objective function value of a chromosome used in GA is given in the following equations:

$$f = \begin{cases} \sum_{i=1}^{n-1} d(p_i, p_{i+1}), & \text{for feasible paths} \\ \sum_{i=1}^{n-1} d(p_i, p_{i+1}) + \text{penalty}, & \text{for infeasible paths} \end{cases} \tag{1}$$

$$d(p_i, p_{i+1}) = \sqrt{(x_{(i+1)} - x_i)^2 + (y_{(i+1)} - y_i)^2} \tag{2}$$

Here; f is the fitness function, p_i is the ith gene of chromosome, n is the length of the chromosome, d is the distance between two nodes, x_i and y_i are robot's current horizontal and vertical positions, $x_{(i+1)}$ and $y_{(i+1)}$ are robot's next horizontal and vertical positions. The direction of the robot is determined by the following equation:

$$\alpha = \tan^{-1} \frac{(y_{(i+1)} - y_i)}{(x_{(i+1)} - x_i)} \tag{3}$$

Objective function value is defined as the sum of distances between each node in a path. If there is an obstacle in the direction of the robot, a penalty is added to the objective function value. The penalty value should be greater than the maximum path length on the environment. In order to find an optimal path, the algorithm searches for a chromosome whose penalty is eliminated.

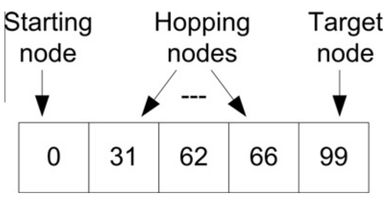

Fig. 2. Decimal coded genes of a chromosome.

Table 1
Comparison of the initial population methods.

Initial population method	Random	Feasible
Fitness value	28.34	28.29
Generation number	81	36
Solution time (s)	0.87	0.49

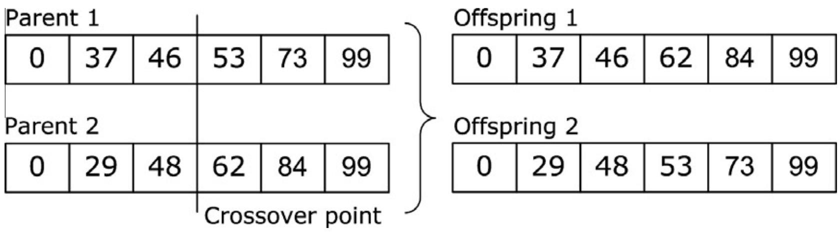

Fig. 3. Single-point crossover.

2.5. Selection method

The main principle of the GA is that the best genes on the chromosomes should be survived and transferred to new generations. At this stage, a selection procedure needs to be done to determine the best chromosomes. The selection process consists of three steps. In the first step, objective function values of all chromosomes are found. At the second step, fitness values are assigned to chromosomes according to their objective function values. In this study, the rank based fitness assignment is used instead of the proportional assignment method. This prevents a few better chromosomes to be dominant in the population. In the last step, chromosomes are selected according to fitness values and then put into a mating pool to produce new chromosomes.

2.6. Crossover operator

Generally, crossover combines the features of two parent chromosomes to form two offsprings. As shown in Fig. 3, single-point crossover operator is used in this study. The genes of two chromosomes after the crossover point are swapped.

2.7. Mutation operator

All candidate chromosomes in the population are subjected to the random mutation after the crossover operation. This is a random bit-wise binary complement operation or a random small change in a gene, depends on the coding of chromosomes, applied uniformly to all genes of all individuals in the population with a probability of mutation rate. The mutation operation expands the search space to regions that may not be close to the current population, thus ensuring a global search [14]. Mutation operation increases the diversity of the population and avoids the premature convergence.

3. A new mutation operator for path planning

In conventional GAs, random mutation is the most commonly used operator. However random mutation can cause generation of infeasible paths. Even though a chromosome is feasible before the mutation operation, the new node changed by the mutation may have an obstacle and therefore it constitutes an infeasible path as shown in Fig. 4a. This makes the optimization slower and increases the number of generations.

In order to overcome this problem, several studies concerned with the improvement of mutation operation have been done in the literature. One of them and the most common is to check the new mutated chromosome for feasibility. If it is infeasible, a new mutation is applied on the chromosome until a feasible one is produced.

Changan et al. [20] use an improved GA, the hill-climbing method which searches for a better individual around the present best one and the mutation individual is replaced by the better individual. Therefore, it may avoid the local optimum.

Another and more effective mutation improvement is suggested by Li et al. [14] specifically for solving the path planning problem for mobile robots. This method selects one node randomly from the set of all free nodes in close vicinity to mutation gene, and then accepts this node according to the forward direction which can be determined by comparison of the coordinates of the starting node and the target node (see Fig. 4b). If the new path after the mutation operation is infeasible or undesirable, random node selection is repeated until the search process of the set is finished.

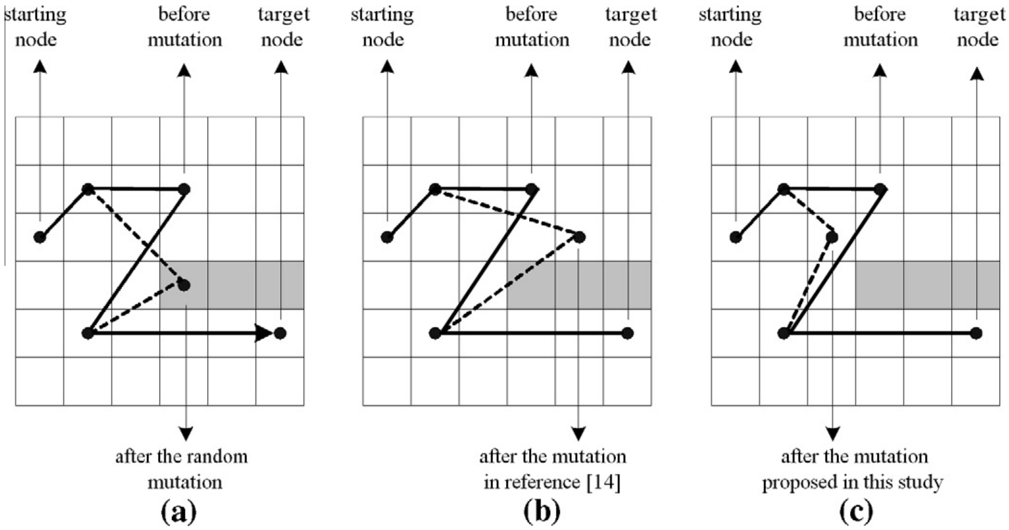

Fig. 4. Comparison of various mutation operators.

Table 2
Procedure of the proposed mutation operator.

Step	Operation
1	Select one node, which is not the start or target node, randomly from the mutation individual as the mutation gene
2	Define a set, which consists of all feasible (non-obstacle) neighbor nodes of mutation node
3	Determine the fitness values of all paths, each one of them consists a neighbor node from the set
4	The mutated node which have the best fitness value is replaced with the original mutation node

In this study, a new mutation operator method is proposed as given in Table 2. Although it looks like the method mentioned in [14], the proposed method differs from that method in two ways. The first, proposed mutation method simultaneously checks the whole free nodes close to mutation node instead of randomly selecting a node one by one. This means that the proposed method guarantees to find the best node, however one by one random selection may accept a better node before finding the best one. The second, the proposed method accepts the node according to the fitness value of total path instead of the direction of movement through the mutated node (see Fig. 4c). Optimal path can be determined by the fitness value, even when the mutated node has opposite direction to the direction from starting to target nodes.

As an example, a path represented by the chromosome (0, 31, 62, 65, 99) is taken from the environment which is shown in Fig. 1. Randomly selected node 65 is taken as the mutation gene. The set of neighbors of mutation node consists of the nodes {54, 55, 56, 64, 66, 74, 75, 76}, however the nodes 54, 55 and 56 are obstacle grids. All neighbors fitness values are found except for the nodes 54, 55, 56. According to fitness values, the path including the node 65 is determined as the fittest path, which has the minimum distance. Thus, the node 65 would be replaced with the node 66 after the mutation, and the new chromosome would be (0, 31, 62, 66, 99).

4. Experiments and performance evaluation for dynamic environments

In order to demonstrate the success of the proposed method, it is applied to two different dynamic environments and compared with previous improved GA studies in the literature. First, the environment used in the [14] is handled. Fig. 5a shows the initial environment, which is the environment before the robot starts to move, consists of 16×16 grids and has six obstacle regions (shaded areas). Fig. 5b shows the modified version of original environment, which is the environment changed after the robot starts to move, has seven obstacle areas. The parameters of the GA are as follows. The population size is taken as 60, crossover probability is taken as 1 and mutation probability is taken as 0.3.

For the environment #1, GA is run one by one with the methods of random mutation, the mutation in Ref. [14], the mutation in Ref. [20] and the mutation proposed in this study. GA is conducted 100 times for each method.

Tables 3 and 4 give the experimental results for the initial and modified environments, respectively. Tables contain the number of optimal solutions, the number of near optimal solutions and the number of infeasible solutions found in 100 trials. Tables also contain the average fitness value, the average generation number and the average solution time of optimal plus near optimal solutions in 100 trials. It is clearly seen in the Tables 3 and 4 that GA with the proposed mutation operator finds the optimal path 54 and 44 times respectively, while the other methods find it only several times. The proposed

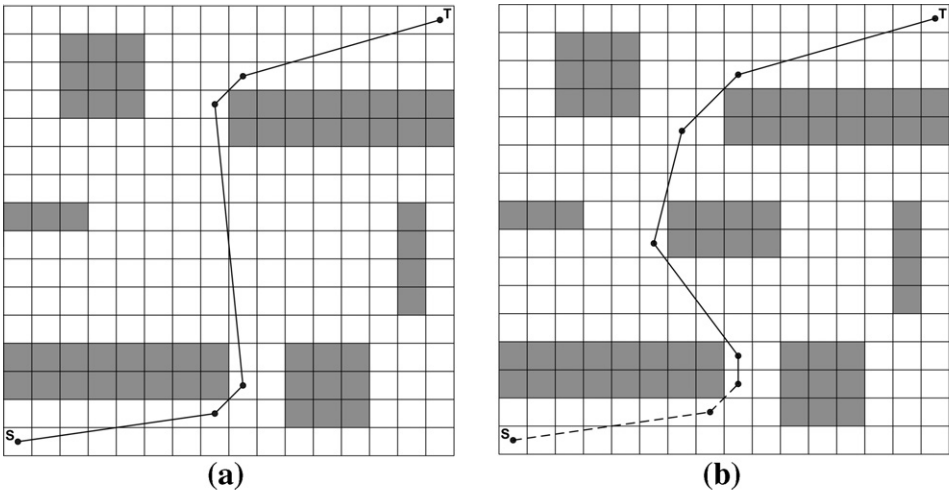

Fig. 5. Initial (a) and modified (b) environments for dynamic path planning example #1.

Table 3
Experimental results for the initial environment in Fig. 5a.

	# of optimal solution	# of near optimal solution	# of infeasible solution	Fitness value	Generation number	Solution time (s)
Random mutation	1	86	13	31.78	21	0.28
Mutation in Ref. [14]	3	69	28	29.25	23	0.31
Mutation in Ref. [20]	2	69	29	29.91	22	0.46
Mutation in this study	54	44	2	27.82	11	0.89

Table 4
Experimental results for the modified environment in Fig. 5b.

	# of optimal solution	# of near optimal solution	# of infeasible solution	Fitness value	Generation number	Solution time (s)
Random mutation	0	95	5	35.37	23	0.20
Mutation in Ref. [14]	0	95	5	31.21	22	0.26
Mutation in Ref. [20]	0	89	11	30.87	23	0.41
Mutation in this study	44	56	0	29.08	11	0.86

method is unsuccessful to find a feasible path in 2 or 0 trials respectively, while the other methods are unsuccessful in 5–29 trials. The average fitness values and the average generation numbers of the proposed method are better than the other methods' values. However, even though all are less than a second the average solution time of the proposed method is worse than the solution times other methods have.

Fig. 6 shows the convergences of the all methods. A GA with the proposed method converges more rapidly than the other methods do. Therefore, this makes the method advantageous in the dynamic environments.

In order to make an extra comparison, we produce a new dynamic environment which is more complex than previous one, as shown in Fig. 7. Fig. 7a shows the initial environment, which is the environment before the robot starts to move, consists of 16×16 grids and has nine obstacle regions (shaded areas). Fig. 7b shows the modified version of original environment, which is the environment changed after the robot starts to move, has ten obstacle areas. The parameters of the GA are as follows. The population size is taken as 80, crossover probability is taken as 1 and mutation probability is taken as 0.2. For the environment #2, GA is run one by one with the same methods as in the environment #1. GA is executed 100 times for each method.

Tables 5 and 6 give the experimental results for the initial and modified environments, respectively. Tables contain the number of optimal solutions, the number of near optimal solutions and the number of infeasible solutions found in 100 trials. Tables also contain the average fitness value, the average generation number and the average solution time of optimal plus near optimal solutions in 100 trials. It is clearly seen on the Tables 5 and 6 that GA with the proposed mutation operator finds the optimal path 9 and 32 times respectively, while the other methods find it only maximum 2 and 15 times

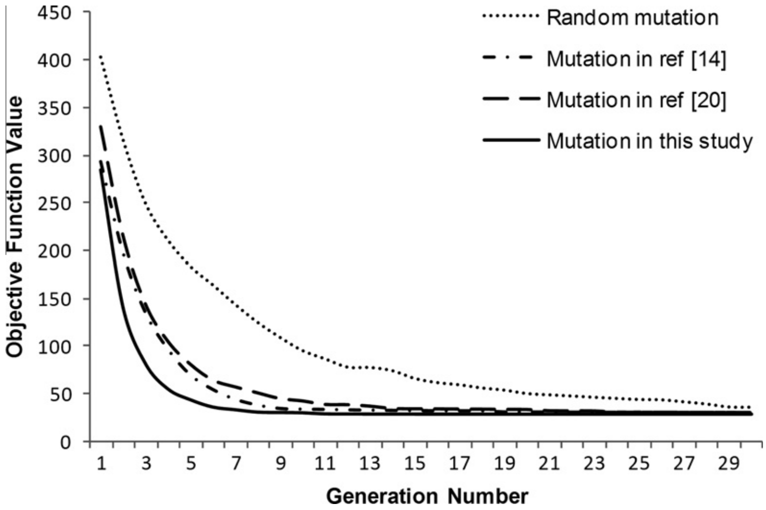

Fig. 6. Comparison of mutation operators' convergences for the example #1.

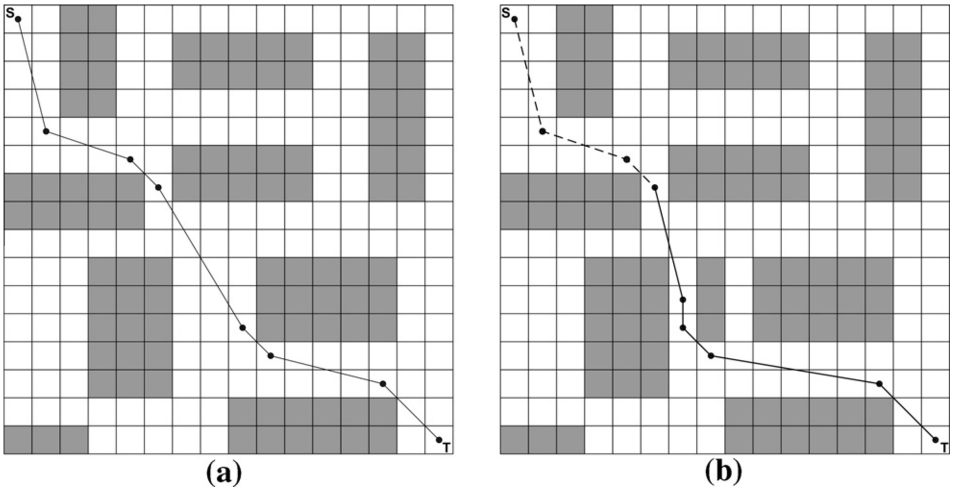

Fig. 7. Initial (a) and modified (b) environments for dynamic path planning example #2.

Table 5
Experimental results for the initial environment in Fig. 7a.

	# of optimal solution	# of near optimal solution	# of infeasible solution	Fitness value	Generation number	Solution time (s)
Random mutation	0	46	54	29.69	81	1.22
Mutation in Ref. [14]	1	68	31	25.02	65	1.03
Mutation in Ref. [20]	2	39	59	25.93	47	0.85
Mutation in this study	9	78	13	24.68	16	1.68

respectively. The proposed method is unsuccessful to find a feasible path in 13 or 0 trials respectively, while the other methods are unsuccessful in 31–59 and 0–8 trials respectively. The average fitness values and the average generation numbers of the proposed method are better than the other methods have. However, even though the average solution time of the proposed method is not better than the solution times of the other methods all are close to each other.

Fig. 8 shows the convergences of the all methods. A GA with the proposed method converges earlier than the other methods do. Therefore, this makes the method advantageous in the dynamic environments.

Table 6
Experimental results for the modified environment in Fig. 7b.

	# of optimal solution	# of near optimal solution	# of infeasible solution	Fitness value	Generation number	Solution time (s)
Random mutation	6	94	0	25.48	73	0.74
Mutation in Ref. [14]	6	92	2	25.17	31	0.34
Mutation in Ref. [20]	15	77	8	25.84	38	0.49
Mutation in this study	32	68	0	24.71	12	0.69

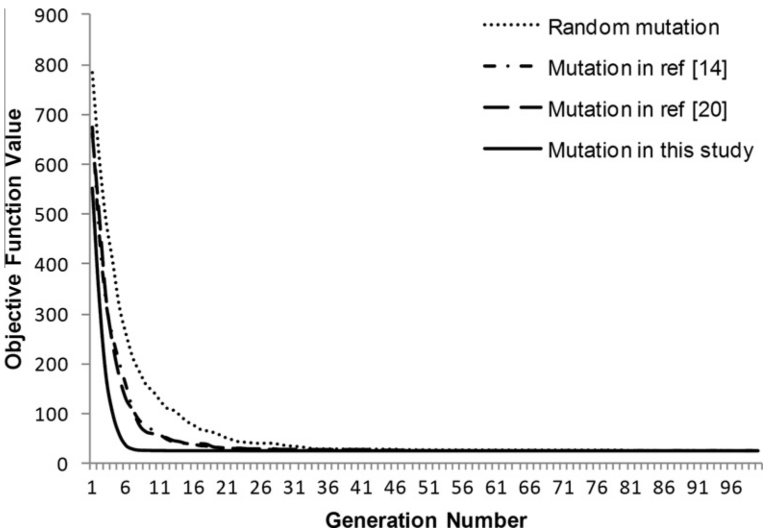

Fig. 8. Comparison of mutation operators' convergences for the example #2.

It is obviously shown on the results that a GA with the proposed mutation method finds the optimal path more frequently than the other methods do. The average fitness values and the average generation numbers of the proposed method are the best in all four methods. Moreover, the proposed method has the most rapid convergence.

5. Conclusion

In this study, we have improved a new mutation operator for the GA and applied to the path planning problem of mobile robots. The improved mutation method simultaneously checks all the free nodes close to mutation node instead of randomly selecting a node one by one. The method accepts the node according to the fitness value of total path instead of the direction of movement through the mutated node. In order to demonstrate the success of the method, it was applied to two different dynamic environments and compared with previous improved GA studies in the literature. It is clearly seen from the results that the GA with the proposed mutation operator can find the optimal path far too many times than the other methods do. The average fitness values and the average generation numbers of the proposed method are better than the other methods'. A GA with the proposed method converges more rapidly than the other methods do. Therefore, this makes the method advantageous in dynamic environments.

Acknowledgement

This work is supported by Kocaeli University's Scientific Research Projects under Project Number 2010/97.

References

[1] Hu Y, Yang SX. A knowledge based genetic algorithm for path planning of a mobile robot. In: Proceedings of the 2004 IEEE, international conference on robotics & automation; 2004. p. 4350–5.
[2] Mohanta JC, Parhi DR, Patel SK. Path planning strategy for autonomous mobile robot navigation using Petri-GA optimisation. Comput Electr Eng 2011;37(6):1058–70.
[3] Willms AR, Yang SX. An efficient dynamic system for real-time robot-path planning. IEEE Trans Syst, Man, Cybern 2006;36(4):755–66.
[4] Ajmal Deen Ali MS, Babu NR, Varghese K. Collision free path planning of cooperative crane manipulators using genetic algorithm. J Comput Civ Eng 2005;19(2):182–93.

[5] Chen MW, Zalzala AMS. Safety consideration in the optimization of paths for mobile robots using genetic algorithms. In: The IEEE international conference on genetic algorithms in engineering systems: innovations and applications, Sheffield; 1995. p. 299–306.
[6] Tu J, Yang SX. Genetic algorithm based path planning for a mobile robot. In: Proceedings for the IEEE international conference on robotics and automation, ICRA '03; 2003. p. 1221–6.
[7] Allaire FCJ, Tarbouchi M, Labonté G, Fusina G. FPGA implementation of genetic algorithm for UAV real-time path planning. J Intell Robot Syst 2009:495–510.
[8] Al-Taharwa I, Sheta A, Al-Weshah M. A mobile robot path planning using genetic algorithm in static environment. J Comput Sci 2008;4(4):341–4.
[9] Elshamli A, Abdullah HA, Areibi S. Genetic algorithm for dynamic path planning. In: Canadian conference on electrical and computer engineering; 2004. p. 677–80.
[10] Yildirim M, Erkan K. Determination of acceptable operating cost level of nuclear energy for Turkey's power system. Energy 2007;32:128–36.
[11] Goldberg DE. Genetic algorithms in search, optimization and machine learning. Addison-Wesley Publishing Company Inc.; 1989 [ISBN: 0-201-15767-5].
[12] Gelenbe E, Liu P, Lainé J. Genetic algorithms for route discovery. IEEE Trans Syst, Man, Cybern, Part B: Cybern 2006:1247–54.
[13] Yao Z, Ma L. A static environment-based path planning method by using genetic algorithm. In: International conference on computing, control and industrial engineering (CCIE), 2010. p. 405–7.
[14] Li Q, Zhang W, Yin Y, Wang Z, Liu G. An improved genetic algorithm of optimum path planning for mobile robots. In: Sixth international conference on intelligent systems design and applications, ISDA '06; 2006. p. 637–42.
[15] Tuncer A, Yildirim M. Chromosome coding methods in genetic algorithm for path planning of mobile robots. In: 26th International symposium of computer and information sciences; 2011. p. 377–83.
[16] Manikas TW, Ashenayi K, Wainwright RL. Genetic algorithms for autonomous robot navigation. IEEE Instrum Meas Mag 2007:26–31.
[17] Naderan-Tahan M, Manzuri-Shalmani MT. Efficient and safe path planning for a mobile robot using genetic algorithm. IEEE congress on evolutionary computation, CEC '09; 2009. p. 2091–7.
[18] Sugihara K, Smith J. Genetic algorithms for adaptive motion planning of an autonomous mobile robot. In: Proceedings of the IEEE international symposium on computational intelligence in robotics and automation, CIRA'97; 1997. p. 138–43.
[19] Nagib G, Gharieb W. Path planning for a mobile robot using genetic algorithms. In: International conference on electrical, electronic and computer engineering, ICEEC '04; 2004. p. 185–9.
[20] Changan L, Xiaohu Y, Chunyang L, Guodong L. Dynamic path planning for mobile robot based on improved genetic algorithm. Chin J Electron 2010;19(2):245–8.

Adem Tuncer received the B.Sc. and M.Sc. degrees in Electronics and Computer Education from Kocaeli University, Turkey, in 2003 and in 2007 respectively. He is currently a research assistant and Ph.D. student at Kocaeli University. His active research interests include path planning for mobile robots, fpga and optimization.

Mehmet Yildirim received the bachelor's degree in Electronics and Computer Education from Marmara University, Istanbul, in 1995 and the Ph.D. degree from Kocaeli University in 2003. He is currently an associate professor in the department of Electronics and Computer Education, Kocaeli University, Turkey. His research interest includes dynamic system modelling, fault tolerant control, heuristic optimization methods.