Genetic algorithm for path planning of mobiles robots in dynamic environment

Célestin Désiré MELEDJE*, Nahéto Elias LOROUGNON et Boko AKA

Laboratoire de Mathématique et Informatique, UFR-SFA, Université Nangui Abrogoua, 02 BP 801 Abidjan 02, Côte d'Ivoire

* E-mail de l'auteur correspondant : desiremeledje@gmail.com

Abstract
Genetic algorithms are commonly used to solve mobile robots path planning problems. Several techniques have been proposed by researchers to improve the efficiency of the genetic algorithm. In this paper, we propose an improvement of the genetic mutation operator for dynamic environments. We compared our method with conventional genetic algorithm and improved genetics algorithms provided by the literature. The comparison was made by an application of these different genetics algorithms in two dynamics environments. Experimental results demonstrate the superiority of our method as compared to what is already known.

Keywords: Robot, Dynamic Path Planning, Energy, Genetic Algorithm

1. Introduction

New technological developments in the field of mobility autonomous robots have increased the interest of path planning researchers; for path planning is fundamental in robotics. In an environment with obstacles, path planning consist in finding a suitable collision-free path for a mobile robot to move from a start location to a target location [1]. Path planning is an important issue in mobile robotics and a major tasks in intelligent robotic systems such as autonomous mobile robots. There are two types of path planning problems. Firstly, static environment path planning, which enables a mobile robot to move in an environment with fixed obstacles. Secondly, dynamic environment path planning, which allows a mobile robot to generate a new path in response to environmental changes [2].

In general, major problems for path planning of mobile robots are the complexity of calculations, the existence of local optima, and the environment adaptability. Researchers have been looking for effective ways to solve these problems [3]. Different methods have been used by researchers to solve the path planning problem for mobile robots in static or dynamic environments. Each method differs by its efficacy in the environment in which it is applied. Moreover, each method has its own strengths and weaknesses.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

Among the existing global optimization methods for solving path planning problems, genetics algorithms have been recognized as a robust optimization heuristic and powerful technic that simulate the natural evolution of the population [4]. The major benefit of genetic algorithms is their ability to provide effective research techniques to find out near-optimal solutions, thus their attractiveness [5].

Recently, researchers have improved path planning genetic algorithms by customizing genetic operators in order to find better solutions. In conventional genetic algorithm, classical random mutation operator are known to cause an unfeasable path. In this article, we present a new mutation operator that increases the diversity of the population and prevents premature convergences.

This article is organized as follows. In Section 2, we explain how genetic algorithms are applied to mobile robots path planning. Section 3 analyzes the mutation operators which were previously reported and a new one proposed in this article. We evaluate the experimental studies and results in Section 4 before concluding in section 5.

2. Path Planning with Genetic Algorithms

Genetics algorithms are global search techniques. They can be considered as stochastic optimization methods. The advantage of using a genetic algorithm to optimize irregular features is that they provide a continue stochastic search in a wide search space or not [6-8].

2.1. Representation of the environment

Several researchers in mobile robot's path planning use the representation of the environment in a grid $n * m$. [1, 9-15]. In this method the grid elements are numbered. Mélèdje and Aka [16] have improved this method by adding the representation of gradients in the environment as shown in Figure 1. This new environmental representation method will be used in this article.

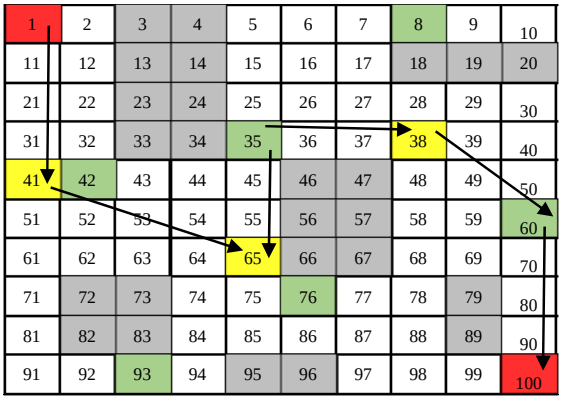


Figure 1: Representation of the environment with gradient.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

2.2. Coding a chromosome

In genetic algorithm methods a chromosome represents a possible solution to a problem. In path planning, the chromosome accounts for a path coding. Many encoding methods have been proposed by researchers. The most used methods are the binary coding [17-19] and the decimal coding [10, 20, 21]. However, the decimal method is more employed in the path planning field. In fact, this method is considered to be more flexible. Moreover, the use of decimal coding requires less memory and space [12]. Figure 2 shows an example of a valid path. That path begins with the starting node and ends by the termination node through the passage nodes.

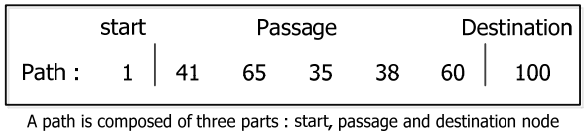


Figure 2: Decimal coding of chromosome

2.3. Initialization of the population

In general, the initial population of most genetic algorithms is randomly generated. This method is simple but produces a lot of unfeasible paths. Albeit, some of them may become feasible after the use of peculiar genetic operators. However, the use of these genetic operators, significantly increases the computation time.

In this study, we will employe the randomly generated initial population method. However, we'll take into account only the produced feasible paths. Tuncer and Yildirim [12] have shown that for a given environment, the use of an initial population containing only feasible paths leads to a shorter execution time and a better evaluation function. This was also demonstrated by other authors [20, 21].

2.4. Fitness function

The evaluation function is designed to assess the quality of a given path S_k. The purpose of evaluation function in this present article is to minimize the required energy to cover a path [16]. This evaluation function is defined by the following equation. When the path is unfeasible, a penalty is added.

$$f(S_k) = E_u(S_k) + Pen(S_k) \tag{1}$$

Where $E_u(S_k)$ represent the sum of the useful energy expended by the robot to traverse the path given S_k. The useful energy can be written from the following formulation:

$$E_u(S_k) = \sum_{e_{ij} \in S_k} \varepsilon(e_{ij}) \tag{2}$$

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

In equation (2), $\varepsilon(e_{ij})$ represents the energy expended to cover the distance between node e_i and node e_j. In equation (1), the term $Pen(S_k)$ accounts for the penalty, defined as:

$$Pen(S_k) = \Gamma \sum_{e_{ij} \in S_k} \beta(e_{ij}) \qquad (3)$$

The parameter Γ denotes the weight of the penalty and $\beta(e_{ij})$ indicates the number of obstacle nodes intersected by edge e_{ij}. If the edge e_{ij} does not intersect any obstacle, then it will be zero. The weight of the penalty Γ can remain constant.

2.5. Selection method

Selection plays an important role in the research process of the genetic algorithm. The selection procedure needs to be done in order to determine the best chromosomes. The selection of individuals depends primarily on their quality (fitness). The selection process is articulated in two steps. In the first step, we calculate the value of evaluation function for each chromosome. In the second step, the chromosomes are classified, selected according to their fitness values and put into a mating pool, to produce new chromosomes.

2.6. Crossover operator

Generally, a crossover generates two offsprings from two parents chosen by the selection operator. In the present study, we use a single crossover point, as shown in Figure 3. Chromosomes gene fragments located after the crossover point are swapped.

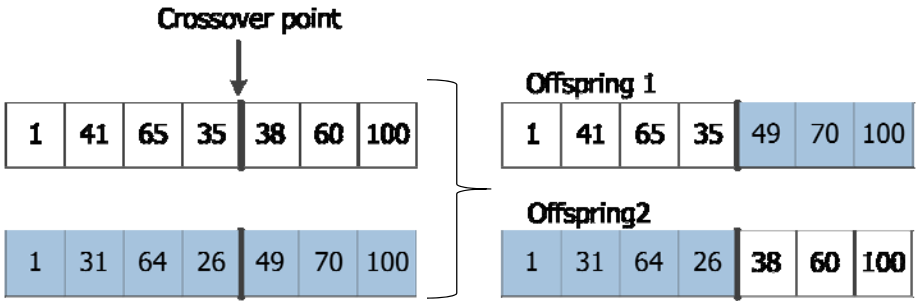


Figure 3: Single point crossover

2.7. Mutation operator

The mutation operator allows small modification chromosomes with very low probability. The objective of the mutation operator is to help the algorithm to escape the search of stagnation problem caused by local optima. It allows a global search, the ability of a genetic algorithm to cover the entire search space [21].

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

3. New mutation operator for path planning

Generally, the mutation operation is randomly performed in conventional genetic algorithms. However, random mutations can generate unfeasible paths. Even if a path is feasible before the mutation operation, the new node modified by the mutation may make it unfeasible as shown in Figure 4a. This modification slows down the optimization and increases the number of generations required to find the optimal solution.

In order to solve this problem, several studies dealing with the improvement of the functioning of mutation operations were carried out. The most common improvement of mutation operator is to verify the feasibility of the new mutated node. If the path resulting from the mutation is feasible, then a new mutation is applied to the chromosome until the resulting path is possible.

Li et al. [21] suggested an improvement of the mutation's efficiency, specifically in resolving the problem of path planning for mobile robots. This method selects a random node from a set of available nodes located near the node mutation. Then this node is accepted if the direction defined by that node to the target node is the same as the direction defined by the starting node to the target node. If the new path created by the mutation operation is unfeasible or undesired, we randomly select a new node from the set of free neighboring nodes. This process is repeated until the end of the searching process.

Tuncer and Yildirim [12] use a method that randomly selects a mutation node that is neither the start node nor the destination node. Furthermore, they determined the set of neighbors of the mutation node. Then each node of neighbor set is put in place of mutation node and they calculate the fitness of each path. The neighboring node with the best fitness is selected as new mutation node. See Figure 4b.

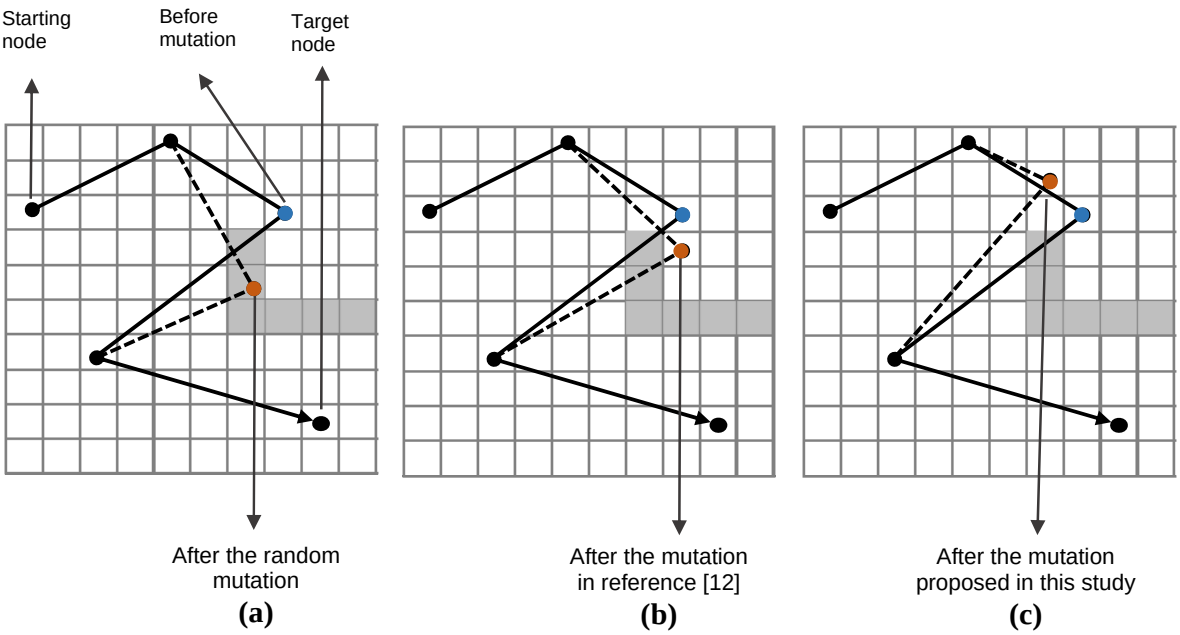


Figure 4: Comparison of different mutations operators

We present, in this study, a new mutation operator method as shown in Table 1. Although it looks like the method mentioned in [12], the proposed method is essentially

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

different in three ways. The first we define $\mathcal{T}_{obs}$, the set of paths that encounters at least one obstacle. The second, in $\mathcal{T}_{obs}$ we define $\mathcal{N}_{obs}$, the set of nodes that precedes an obstacle. The third, in $\mathcal{N}_{obs}$ we select randomly a mutation node, which must not be the starting node or the target node. After defining the set of neighboring nodes of the mutantion node, we calculate the fitness of each path where mutation node is remplace by each node of neighbor set. The node whose path will provide the best fitness will be selected as a new mutation node (see Figure 4c).

Table 1: Procedure of the proposed mutation operator

Step	Operation
1	Select randomly a node encountering an obstacle which is not started or target node as mutation gene
2	Define a set, which consists of all feasible (non-obstacle) neighboring 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 has the best fitness value is replaced with the original mutation node

4. Experimental results for dynamic environment

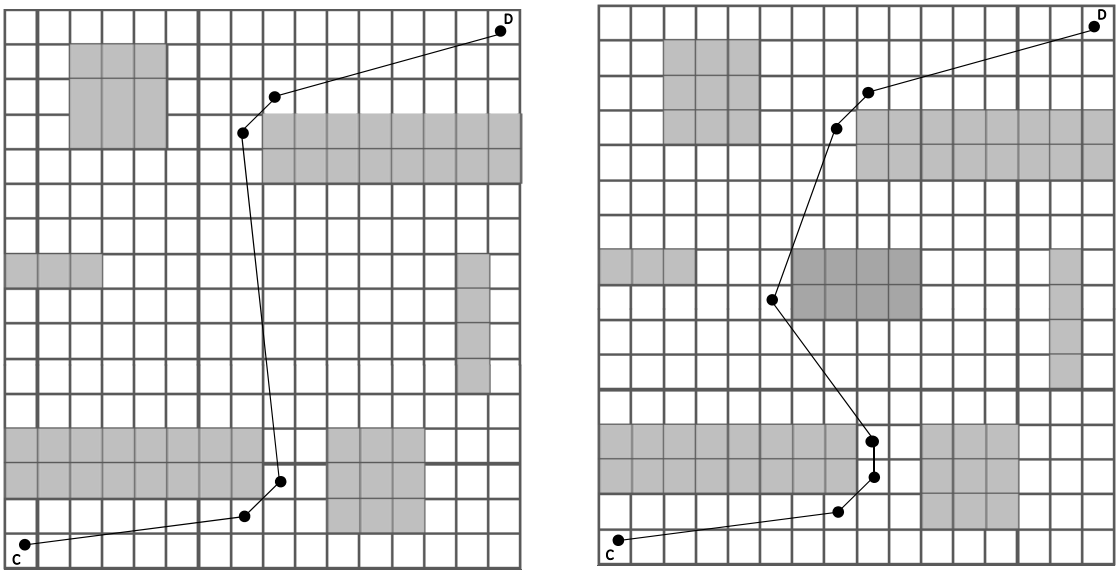
In order to determine the effectiveness of our method, we applied the two dynamic environments presented in [21]. A dynamic environment is an environment that is changed after the mobile robot starts moving. The two environments are represented as grids $16 * 16$ with obstacles regions (shaded areas).

In the first dynamic environment (figure 5), we used the following parameters to complete our tests. The population size was set to 60, whereas the crossover probability was set to 1 and the mutation probability was set to 0.3. We compared our method with random mutation and two improved genetic algorithms proposed in references [12] and [21]. Each genetic algorithm was executed 100 times. The robot ought to begin from the starting point (D) to the target point (C).

Tables 2 and 3 show the experimental results respectively for the initial and the modified environments. The tables contain the numbers of optimal, near optimal and unfeasible solutions found in 100 trials. They also contain the average fitness value, the average generation number and the average sol ution time of optimal and near optimal solutions in 100 trials.

Tables 2 and 3 also demonstrate that our method provides the greatest number of optimal, near-optimal solutions and the smallest number of unfeasible paths. Moreover, our method produced the lowest fitness values and convergence is reach with generation lowest number. However, our method take more time than other methods.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500



(a) **(b)**

Figure 5: Initial Environment (a) and modified (b) for planning path Example 1

Table 2: Experimental results for the initial environment fig. 5a

	Optimal solution	Near optimal solution	Infeasible solution	Fitness value	Generation number	Solution time (s)
Random mutation	2	93	5	30,57	23	0,27
Mutation in Ref. [12]	52	46	2	27,81	13	0,84
Mutation in Ref. [21]	5	64	31	29,63	21	0,47
Proposed mutation	63	36	1	26,43	11	0,91

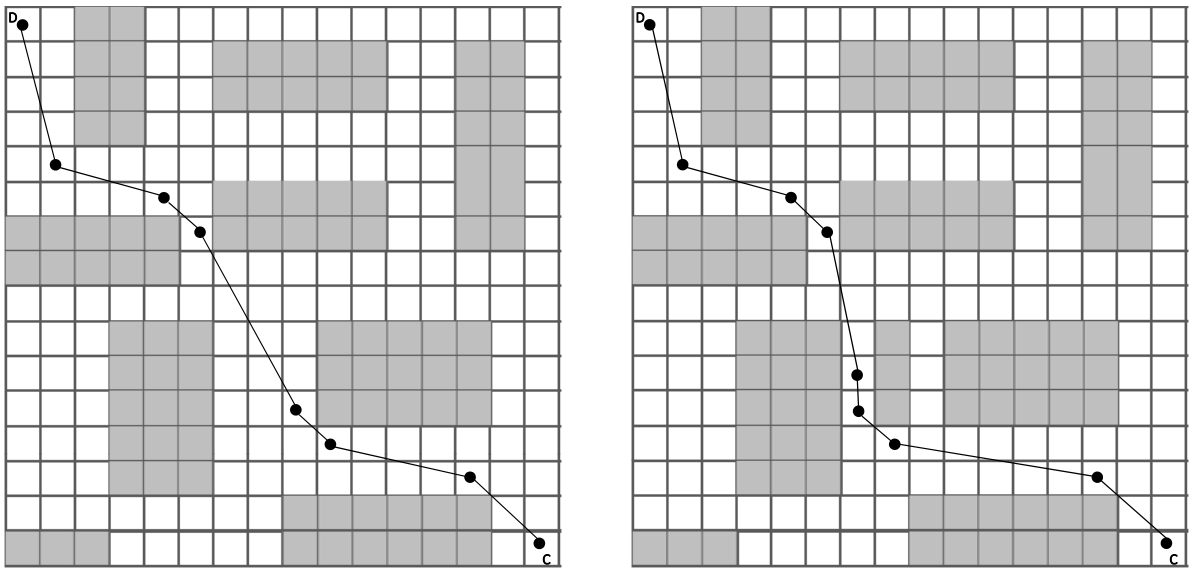
Table 3: Experimental results for the modified environment fig. 5b

	Optimal solution	Near optimal solution	Infeasible solution	Fitness value	Generation number	Solution time (s)
Random mutation	0	83	17	35,87	26	0,30
Mutation in Ref. [12]	43	52	5	30,33	15	0,88
Mutation in Ref. [21]	0	87	13	33,12	24	0,38
Proposed mutation	51	46	3	29,52	13	0,95

In order to compare our method to the others, we used a more complex dynamic environment than the last, as shown in Figures 6a and 6b. This new environments are represented as grids $16 * 16$ with obstacles regions (shaded areas).

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

Figure 6a depicts the initial environment and 6b indicates the changed environment after the robot starts moving. The population size was set to 80, the crossover probability was set to 1 and the mutation probability was set to 0.2. The genetic algorithm was executed 100 times for each method. The robot must start from the starting point (D) to the target point (C).



(a) **(b)**

Figure 6: Initial Environment (a) and modified for path planning Example 2

Table 4: Experimental results for the modified environment fig. 6a

	Optimal solution	Near optimal solution	Infeasible solution	Fitness value	Generation number	Solution time (s)
Random mutation	7	87	6	38,41	75	0,76
Mutation in Ref. [12]	33	63	4	36,70	13	0,64
Mutation in Ref. [21]	15	68	17	37,84	38	0,48
Proposed mutation	41	56	3	35,87	11	0,80

Tableau 5: Experimental results for the modified environment fig. 6b

	Optimal solution	Near optimal solution	Infeasible solution	Fitness value	Generation number	Solution time (s)
Random mutation	1	45	54	40,32	83	1,24
Mutation in Ref. [12]	10	77	13	33,01	17	1,67
Mutation in Ref. [21]	3	37	60	37,87	47	0,91
Proposed mutation	25	68	7	34,11	14	1,89

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

Despite the complexity of the environment in Figure 6, the results obtained after the tests confirm the results observed in environments shown in Figure 5. Our method always provided optimal solutions and less unfeasible paths. However, as in the environment of Figure 5, our method runs longer. Tables 4 and 5 show these results.

The experimentales results demonstrate that mutation proposed in this articles is better than main mutations present in literature. Indeed, mutation proposed find more soptimal solution than other mutation. In addition, our method provides more solutions near to the optimal solution. The average value of fitness and the number of generation show that our method is the best of the four tested methods.

Conclusion

In this paper, we propose a path planning in a dynamic environment for a mobile robot based on an improvement of the mutation operation in a genetic algorithm. To ascertain the effectiveness of our method, we applied it to two types of dynamic environments and we compared it to subsequent methods in the literature. The results demonstrate that our methods provided more optimal solutions than previous methods for the dynamic environment in path planning.

Reference

[1] Hu Y. and Yang S. X. "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. 4350–4355.

[2] Elshamli A, Abdullah HA and Areibi S. "Genetic algorithm for dynamic path planning". In: Canadian conference on electrical and computer engineering; 2004. p. 677–680.

[3] Mohanta J. C., Parhi DR and Patel S. K. "Path planning strategy for autonomous mobile robot navigation using Petri-GA optimisation". Comput Electr Eng 2011;37(6):1058–1070.

[4] J. C. Gallagher, S. Vigraham, and G. Kramer. "A family of compact genetic algorithms for intrinsic evolvable hardware". IEEE Trans. Evol. Comput., vol. 8, no. 2, pp. 111–126, Apr. 2004.

[5] Yang Linquan, Luo Zhongwen, Tang Zhonghua, and Lv Weixian. "Path planning algorithm for mobile robot obstacle avoidance adopting bezier curve based on genetic algorithm". Chinese Control and Decision Conference, 3286-3289, 2008.

[6] Yildirim M and Erkan K. "Determination of acceptable operating cost level of nuclear energy for Turkey’s power system". Energy 2007; 32. Page: 128–136.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

[7] Goldberg D. E. "Genetic algorithms in search, optimization and machine learning". Addison-Wesley Publishing Company Inc.; 1989 [ISBN: 0-201-15767-5].

[8] Gelenbe E, Liu P and Lainé J. "Genetic algorithms for route discovery". IEEE Trans Syst, Man, Cybern, Part B: Cybern 2006:1247–1254.

[9] M. Gemeinder and M. Gerke. "GA-based path planning for mobile robot systems employing an active search algorithm". Appl. Soft Comput., vol. 3, no. 2, pp. 149–158, 2003.

[10] Y. Hu, S. Yang, L. Xu, and Q. Meng, "A knowledge based genetic algorithm for path planning in unstructured mobile robot environments". In Proc. IEEE Int. Conf. Rob. Biomimetics, 2004, pp. 767–772.

[11] S. Yang, Y. Hu, et M. Meng. "A knowledge based GA for path planning of multiple mobile robots in dynamic environments". In Proc. IEEE Conf. Rob. Autom. Mechatron., 2006, pp. 1–6.

[12] A. Tuncer and M. Yildirim. "Dynamic path planning of mobile robots with improved genetic algorithm". Comput. Electr. Eng., vol. 38, pp. 1564–1572, 2012.

[13] C. Tsai, H. Huang, and C. Chan. "Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation". IEEE Trans. Ind. Electron., vol. 58, no. 10, pp. 4813–4821, 2011.

[14] S. Yun, V. Ganapathy, and L. Chong. "Improved genetic algorithms based optimum path planning for mobile robot". In Proc. 11th Int. Conf. Control Autom. Rob. Vision, 2010, pp. 1565–1570.

[15] Bo-Yeong Kang, Miao Xu, Jaesung Lee and Dae-Won Kim. "ROBIL: Robot Path Planning Based on PBIL Algorithm". Int J Adv Robot Syst, 2014,11:147.

[16] C. D. MELEDJE and B. AKA, "Mobile robot path planning in new map representation and useful energy evaluation" European Journal of Scientific Research, Vol 141 No 2 August 2016

[17] Elshamli A, Abdullah HA and Areibi S. "Genetic algorithm for dynamic path planning". In: Canadian conference on electrical and computer engineering; 2004. p. 677–680.

[18] Sugihara K and 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. 138–143.

[19] Nagib G and Gharieb W. "Path planning for a mobile robot using genetic algorithms". In: International conference on electrical, electronic and computer engineering, ICEEC ’04; 2004. p. 185–189.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

[20] Yao Z and 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. 405–407.

[21] Li Q, Zhang W, Yin Y, Wang Z, and 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. 637–642

https://sites.google.com/site/ijcsis/
ISSN 1947-5500