Nat. Volatiles & Essent. Oils, 2021; 8(5): 1675 - 1688

NVEO
Natural Volatiles & Essential Oils

New Paths Generation Using Crossover And Mutation Genetic Operations For Mobile Robot To Find The Optimal Path In 2-D Static Environment

[1]Dr. Aarti Amod Agarkar, [2]Dr. Mohd Ashraf, [3]Dr. Anandakumar Haldorai, [4]Dr. Md. Zair Hussain, [5]Dr.Ram Subbiah, [6]Dr. Bharathababu K

1 Associate Professor, Marathawada Mitra Mandal's College of Engineering, Pune. pratibha26@gmail.com
2 Associate Professor, Computer Science & Engineering, School of Technology, Maulana Azad National Urdu University, Hyderabad (TS). ashraf.saifee@gmail.com

3 Professor (Associate), Department of Computer science and engineering, Sri Eshwar College of Engineering, Coimbatore, Tamil Nadu, India 641202. anandakumar.psgtech@gmail.com

4 Associate Professor, Information Technology, School of Technology, Maulana Azad National Urdu University, Hyderabad (TS). mdzairhussain@gmail.com

5 Associate Professor, Mechanical engineering, Gokaraju Rangaraju institute of Engineering and Technology, Hyderabad. ram4msrm@gmail.com

6 Assistant Professor, Anand Institute of Higher Technology, Chennai-603103. ckbharathababu@gmail.com

Abstract

The development of new pathways is the essential technique of robot navigation decision-making and guidance, as well as hotspots for study in the domain of artificial intelligence. In this research, an enhanced multi-objective genetic algorithm (IMGA) has been suggested to handle 2D stable environment difficulties such as sluggish reaction speed, dangerous factors, a lot of turns within the traditional path planning approach, and the simplest planning path. To make sure the efficiency of such a planned route, the algorithm employs the heuristic median implantation technique to demonstrate the actual population that also enhances the viability of an actual path and creates multiple objective fitness functions depending upon three factors: path protection, path energy usage, and path length. Furthermore, by employing a layered approach, as a single-point crossover technique, as well as an eight-neighborhood-domain single-point modification technique, the selection, crossovers, and modification operators were created. Eventually, the deletion action is included to assure the mobile robot's effective service. Simulation studies in a 2D stable environment allow for a modest converging rate and an easier fall into such a regional optimal through the simplest route to the destination location.

Keywords: Crossover operators, Evolution operators, Shortest route, stable environment, Genetic algorithm.

1. Introduction

Path planning seems to be a significant study area in the realm of mobile robotics, and this is also the major challenge in that study [1]. The goal of the path planning issue is to discover the best and quickest path from starting position to the destination position independently without any conflicts in such a specific environment containing obstacles [2, 3]. Path modeling is frequently employed in areas like logistical delivery, missile guidance, and smart mobility [4–6]. As a result, how to select a quick and efficient approach is becoming a topic of research with both conceptual and applied relevance.

Due to its excellent universal optimization capability and inherent parallel computational properties, genetic algorithm (GA) was widely employed in portable robot path routing issues in current times [7, 8]. This GA seeks the best answer by modeling natural growth using mathematical concepts of genetic inheritance as well as variability from Darwin's evolutionary biology [9, 10]. However, some significant GA results must be published. A generalized fragmentation crossover operator, for instance, was included in GA for enhancing the algorithm's regional optimization capability and implementation performance [11]. A revised greedy sunbath evolution operator to address the traveling salesman problems (TSP) and included a greedy searching strategy further into GA evolution action must be created by Allahverdi and Albayrak [12, 13]. Moreover, an enhanced crossover operator [14] also suggested, that avoids early convergence for producing an optimum route in stable situations. Depending upon the enhanced genetic algorithm [15], the robot's route planning technique was suggested during which the flexibility of such a movable robot route planning approach was increased by inserting chromosomes with varied lengths. To retain population variety, prevent early convergence, and preserve concurrency with classic genetic techniques, as well as a parallel elitist genetic approach has been presented [16].

The updated form of basic GA is called multiple objective GA (MOGA). Concerning fitness function assignments, MOGA differs, unlike GA. All other steps are the same as GA. A primary goal of multiple objectives GA will be to construct the optimum Pareto Front at the objective space that any such improvement for any fitness value is possible without interfering with other objective functions [17]. The primary goals of multiple objective GAs include coverage, variability, and convergence. Multiple objective GAs were broadly classified into two types: decomposition-based multiple objective GAs and Pareto-based multiple objective GAs [18]. Such methods are covered in the prior sections.

Crossover processors combine the genetic data for two or even more parents can produce offspring. K-point, two-point, single-point, uniformity, sequence, partly matched, priority maintaining crossover, cycle, shuffling, and simplified surrogacy is examples of really well crossover processors. A unique crossover point being chosen in such a singular point crossover. Genetic data for two parents who have progressed outside that point would be exchanged [19]. To create the unique offspring, it swapped the tailed array elements of each parent. Eshelman et al. [20] presented shuffling crossover to eliminate the bias produced by existing crossover strategies. It scurries the elements of each independent solution before the crossover then rearranges it once the crossover process is completed ensuring that the crossovers point may not induce bias within the crossover. Unfortunately, the use of such a crossover has been fairly restricted in current times. When the parents had identical gene patterns for resolution depictions, reducing surrogate crossover (RCX) avoids needless crossovers [21]. This RCX would be based upon the premise which GA provides superior persons if such parents' genetic compositions are adequately diversified. Unfortunately, RCX cannot create superior offspring for parents with the same genetic make-up. Oliver suggested cycle crossing [22]. It aims to develop offspring from parents, so every component holds its place by referencing its parents' positions. It gets some components from its first parent during the first round.

A genetic activator called mutation preserves genetic variation from one generation to the next. Relocation, basic reversal, and scrambled mutation are so well mutation operators. The displaced mutation (DM) operator moves a segment of an independent response in itself. The position for relocation is picked at arbitrary from the specified substring, so that if the obtained response is legitimate and also a randomized relocation mutation. DM variations include interchange mutation and inclusion mutation. As in interchange and inclusion mutation processors, a portion of such an independent response will be either swapped with some other portion or injected in a different position, accordingly [23].

2. Contribution

In summary, the following are the important features of this article:

(i) A heuristic means inclusion method is intended to establish a viable beginning path and speed up that algorithm's convergence speed.

(ii) Multiple objective fitness functions have been suggested that also assign various measurement prerequisites as well as weights to every indicator based upon planning requisites ensuring that such planned route does have the smallest route length, the best security, and softness, and achieves multiple objective enhancement of such method.

(iii) To preserve inclusivity of the population and prevent early convergence in the final phases of the technique, the preference operator is constructed using the layered technique, a crossover operator has been developed using the single-point crossover technique, as well as mutation operator would be constructed using the eight neighborhood-domain single-point mutation strategy.

(iv) To achieve much more effective routes, the deletion operator is included to eliminate the unnecessary nodes of such an existing route.

(v) The technique intricacy is lowered as well as the operating speed is increased on the basis that every approach is easy and efficient, and also that the ideal path may be constructed.

3. Related Works

Several researchers have used grid-based areas to depict given surroundings while responding with CFSPP difficulties. To put it another way, the surroundings are separated into squares cells. MAs are restricted as they constantly migrate to the middle of their eight surrounding cells (down, up, right, 4 diagonal directions, and left). These are performed to simplify a lot of research [24], however, it is difficult to establish true CFSPP under such a constraint [25], as seen in Figure 2a. As a result, several researchers have hypothesized that MAs may travel to an empty cell without being limited to neighborhood [26]. Many techniques were used to rectify CFSPP troubles, such as the A-star technique [27], the fuzzy concept [28], simulated nitriding [29], artificial potential field (APF) [30], Dubin's equation [31], Voronoi schematic, evolution algorithm (EA), transparency graph [32,33], quickly discovering arbitrary tree [34], as well as nature, influenced methodologies [35]. The transparency graph was initially established in 1979 to prevent conflicts with barriers and has since been revised as an idea using the configuration area methodology. Rashid et al. employed a transparency graph during their study, as well as the MAs but also barriers were supposed to be cycloid in structure. Researchers chose the corners of every impediment to being nodes. They constructed transparency trees via arraying connected nodes till the trees contained the destination node.

Path planning techniques are broadly classified into four types: classic algorithms, graphical approaches, smart bionics methodologies, and other techniques [36]. Both for its resilience, flexibility, and quick arbitrary search capabilities, this genetic algorithm (GA) has been extensively employed in portable robot route planning studies [37]. The classic genetic approach, on the other hand, never only possesses a slower convergence time but is also prone to early convergence [38]. As a result, numerous researchers have enhanced the genetic operators to overcome the restrictions of GA. Zhang et al. [39] presented a better genetic approach based upon observable area. The principle of transparent area, matrix computing, and enhanced mutation operators seem to be the fundamental principles. That approach works in both simple and complex settings. Furthermore, the regularity and protection of a path aren't taken into account, and the picking process is quite arbitrary, which may raise the algorithm's time complexity. To overcome the robot's path scheduling issue, Lamini et al. [40] presented an enhanced similar attribute crossover operator. Given the chromosome's changeable size, the operators may construct a viable path with such a higher fitness function and prevent early convergence. Unfortunately, the method's beginning population value still has to be enhanced, as well as the length of the path doesn't approach the optimum value.

The viable path might be better in line with real needs by optimizing the fitness value. Chen and Chen [41] utilized previous experience to build continuous viable routes and modify the fitness value, significantly improving the genetic approach and allowing it to finish development in less time. Moreover, the optimum path produced by this technique still contains duplicate nodes, as well as the genetic algorithms operator maximization, as well as the algorithm's speed and reliability, must be increased. To overcome the path scheduling issue of a tunable robot in such a complicated impediment ecosystem, Cheng et al. viewed path planning like a multiple objective optimization issue and analyzed the quality of the outcomes premised on four self-defined fitness objective values, however, the algorithm's efficacy still wants to be enhanced.

4. Proposed Algorithmic approach

The enhanced multiple objective genetic algorithms (IMGA) has been described in this part for planning and selecting the ideal operating route of such a robotic system in such a grid stationary context

4.1. Initialization of the Population:

The starting population is generally generated by an arbitrary approach in a conventional genetic method. However this approach is quick and simple to use, the fraction of impossible pathways in the created pathway is too high, affecting the method's convergence rate as well as operational performance. This work presents a heuristic means inclusion approach to generate the beginning population to effectively produce a better quality starting population as well as increasing the algorithm's overall efficiency. This strategy's precise procedure is as mentioned earlier:

(1) Compute that population M's size.
(2) Calculate the grid maps size n $*$ n (indicating n rows as well as n columns), a portable robot route planning beginning spot S (beginning point S could be stated as N_1), the destination point G (destination point G could be represented as N_{n*n}), as well as the amount of impediment grids f (total number of empty grids equals n $*$ n − f).
(3) Create a pathway chromosome from the beginning site S to a destination site G, with the beginning site S usually being the chromosome's initial gene as well as the destination point G usually being the chromosome's final gene.
   (i) Create a grid value N_i at random (not a part of the barrier grid, beginning point, or finishing place). During that stage, the robot's route could be written as$S - N_i - G(or\ N_1 - N_i - N_{n*n})$.
   (ii) As per formula(2), evaluate whether the nearby nodes as in path is consistent

$$\Delta = max\{abs(x_{i+1} - x_i), abs(y_{i+1} - y_i)\} \quad (1)$$

The rectangular dimensions of two neighboring path locations N_i and N_{i+1} are represented as in formula by (x_i, y_i) and (x_{i+1}, y_{i+1}). While$\Delta = 1$, the two points N_i and N_{i+1} are continual; else, they are in discontinuity. And next inclusion location is chosen using the median technique to complete the uneven path at that point. Equation 3 shows the actual computation:

$$\begin{cases} x_i' = int\left[\frac{(x_i + x_{i+1})}{2}\right], \\ y_i' = int\left[\frac{(y_i + y_{i+1})}{2}\right], \\ N_i' = x_i' + \mathrm{n}.\, y_i' . \end{cases} \quad (2)$$

The dimensions of the nominee grid were x_i' and y_i' within the formula, n represents the total number of columns and rows within a grid, as well as N_i' seems to be the number of an applicant grid. If N_i' would be an empty grid, this is simply placed among N_i andN_{i+1}; however, the empty grid in N_i' 's eight adjacent nodes are arbitrarily chosen as the recently implanted node. As seen in Figure 1, a grey region around the N node seems to be the eight communities of such a spot. When there is no available grid within eight communities ofN_i', it suggests that such action is illegitimate, as well as the individual has been immediately disposed of. Reiterate the above implant stages to create a consistent viable path.

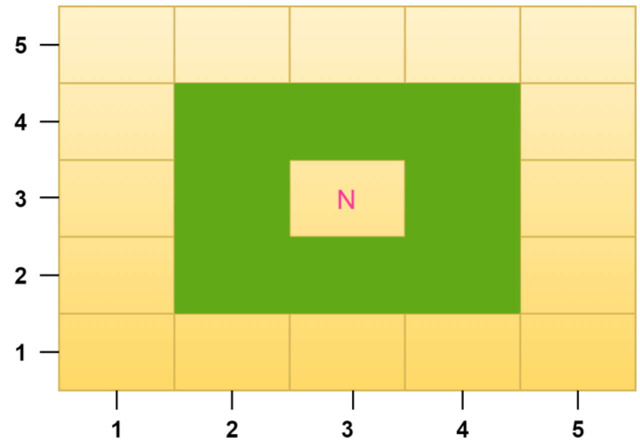


Figure 1: Eight neighborhood nodes.

(iii) Follow the previous steps until you have a starting population with M no repeating chromosomes.

4.2. Fitness Function:

Just after the basic population is generated, the method must create a fitness value to evaluate every individual's productivity and evaluate their advantages and disadvantages. The fitness value is identical to the problem's objective value in this case. Under this article, a multiple objective fitness value depending on path energy utilization, path length, path security, and path length is created to accelerate the convergence of such genetic method whereas maintaining low intricacy, and to identify the effective path which can seamlessly avert barriers and quickly achieve the specified point, that is particularly described in the following manner:

$$F(N) = \frac{1}{a \cdot L(N) + b \cdot S(N) + c \cdot E(N),} \qquad (3)$$

Here $L(N)$ denotes the length of the path, $S(N)$ denotes the route security, and $E(N)$ denotes the pathway energy usage. The values of the three variables are $a, b,$ and$c$. The length of the path is equal to the total of the Euclidean distances among all neighboring nodes, as stated in solution 5.

$$L(N) = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}, \qquad (4)$$

Here n denotes the number of nodes on the path.

4.3. Genetic Operation:

4.3.1. Selection Mechanism:

The selective operation seems to be the aspect of genetic action that most reflects the "preservation of a fittest." For preventing the non-directional mistake generated by the standard roulette selecting approach this work uses a layered strategy to construct the selecting operator, like observes: A maximum of M members are formed just after the population was initialized, the fitness function of every person is determined using the fitness value, as well as the fitness functions are organized in decreasing sequence. This population can be separated into three groups each of which represents a layer, with M/3 persons in every layer. To construct an offspring population, replicate the aliquot mostly with greater fitness function in the first layer, make copies with medium fitness function as in the second layer, but do not repeat the final fitness with smaller fitness. As well as the population size remains M (when M wasn't divided by 3, and the amount of persons during the final portion becomes$\frac{M}{3} + mod(M/3)$). While repeating its first two levels, mod $(M/3)$ people with higher fitness functions must be chosen from its third layer and enter the progeny population, ensuring also that the progeny population's number remains M). It guarantees that its best people are passed down to future generations thus retaining population diversity. Figure 2 shows the procedure of decision.

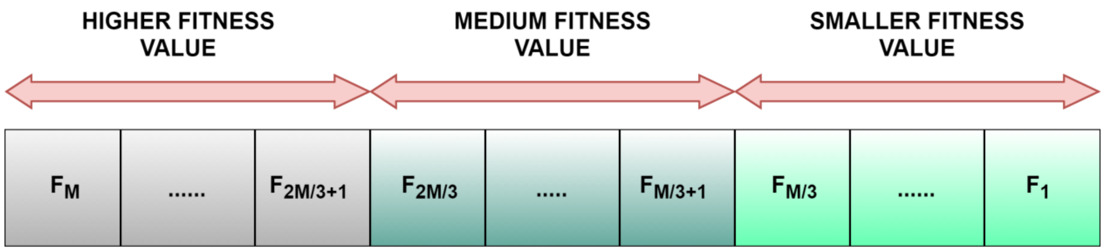


Figure 2: Fitness segmentation.

4.3.2. Crossover Mechanism:

The crossing process is a more critical genetic action as in method, so it involves chromosome crossing. Such that, new child chromosomes were generated by swapping many of all genes of its parent chromosomes. A single-point crossover approach is used in our study. Crossover at this place with any of these two parent entities must be selected with the same series number (excluding the beginning site S as well as the goal site G). M_1: $S - 2 - 12 - 67 - 69 - 89 - G$; M_2: $s - 11 - 22 - 67 - 78 - 80 - G$are two examples of parent individuals. $M_1' = S - 2 - 12 - 67 - 78 - 80 - G$; $M_2' = S - 11 - 22 - 67 - 69 - 89 - G$ seem to be the two offspring produced if the identical series number 67 gets chosen as the crossing point. There is no crossing operation if this sequential number cannot be present in both parent persons.

4.3.3. Mutation Mechanism:

To sustain population diversity, the recombination process involves changing either gene on such a person's chromosome to form a different chromosome. Conventional multi-point recombination, single-point mutation as well as other processed approaches, on the other hand, are susceptible to producing impossible routes, that will reduce the algorithm's performance. As a result, this research employs the eight-point neighboring single-point alteration approach. To displace an existing node, initially pick a variance point N_i (other than the beginning spot S and destination spot G) also in the person of its path to only be evolved, and afterward pick the non-obstacle grid called N_i' (other than the neighboring nodes N_{i-1} and N_{i+1} within a path to also be evolved) and in eight neighboring of a mutation spot as depicted in Figure 1. N_{i-1} to N_i' and N_i' to N_{i+1} are then linked in a continuous path using the basic path construction approach. This evolution failed when there exists no empty grid within eight neighbors of evolution spots to pick from, or when a suitable continuous path may not be constructed. Such that exit the evolution process and reassign a next person to also be altered as well as the alteration point.

4.3.4. Termination Constraints:

This terminating constraint is a criterion for determining if or not the genetic technique is capable of terminating the process. The ideal fitness function for a specified evolving mathematical threshold of 40 but rather 50 successive evolving populations is still the same, or the method operates moreover 5 minutes according to the terminating criterion described in this study.

4.3.5. Delete Mechanism:

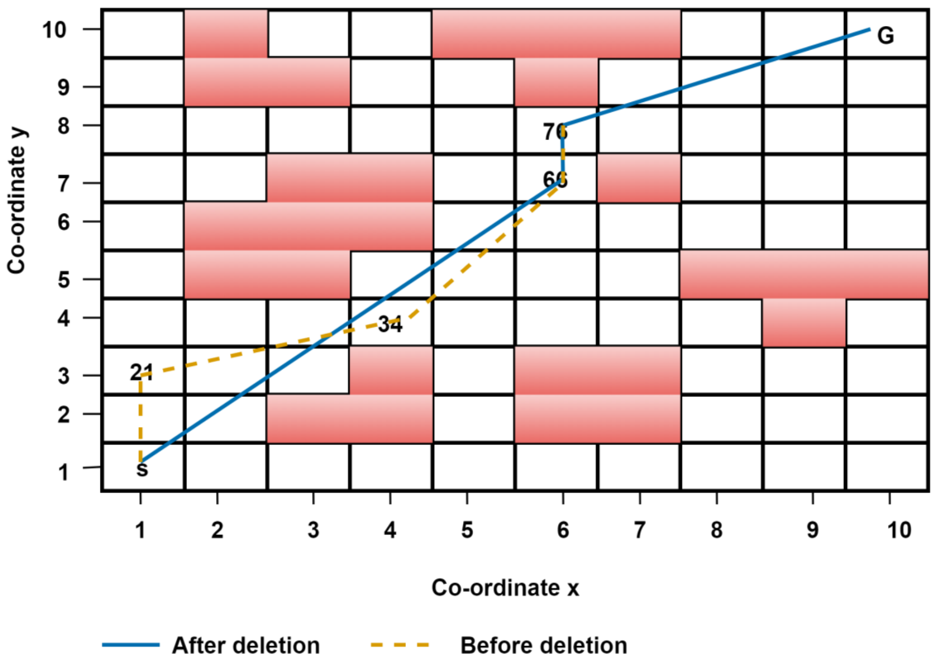


Figure 3: Pathway in deletion mechanism.

In this paper, a remove action is also included the scenario of duplicated nodes within the pathway. The key principle is that whenever a node could link towards its frontal and back nodes (nonadjacent route points) without any barriers, therefore the intermediary nodes among such two places are unnecessary. Remove this superfluous node and link these two places straight to meet the objective of lowering path length as well as preventing wasteful turns. This method is just for the ideal pathways generated by every cycle to prevent the lowering of route nodes that after deletion action, which impacts the viability of crossing and mutation actions in genetic processes. The initial best path is $S - 21 - 34 - 66 - 76 - G$, while the ideal path upon deletion is $S - 66 - 76 - G$, as illustrated in Figure 3. There will be four turns within the initial course, as you can see. The duplicate nodes 21 and 34 within the segment between S site to series site 66 are eliminated just after the elimination process, resulting in only two turns within the entire path as well as a smaller path length.

5. Experimental Analysis

Java jdk1.8, Windows 10, 16 GB RAM, and Intel Core i7-4790K central processing unit (CPU) (4.0 GHz) processors which are components and that are used to execute the suggested IMGA technique and GA. To assess performance, five distinct situations were tested. Stable Ecosystems 1 and 2 are tiny situations having 16 x 16 grid areas, but Ecosystems 3–5 are larger difficulties having 30 x 30 grid areas. For every scenario, GA must be run hundred times. The outcomes of the experiments in Ecosystems 1 and 2 are contrasted to those of prior investigations. The MA proceeds from the unit (1, 1) to unit (16, 16) in Scenario 1, which has seven impediments. There can be ten barriers in Scenario 2, as well as the robotic goes from the unit (1, 16) to cell (16, 1). In both contexts, the GA comes to a halt if the optimal fitness function does not change after 500 consecutive generations.

The suggested technique's fitness function is better in every setting, as well as the mean for all fitness functions is superior. In comparison to previous methodologies, the mean fitness function of this research is superior to that of the optimum fitness function. The minimum collision-free route determined in every setting must be depicted in Figure 4a and Figure 4b. Also, it's worth noting that, when contrasted to earlier techniques, the suggested technique finds good pathways quicker in both contexts. Figure 5 illustrates how quickly the suggested approach converges.

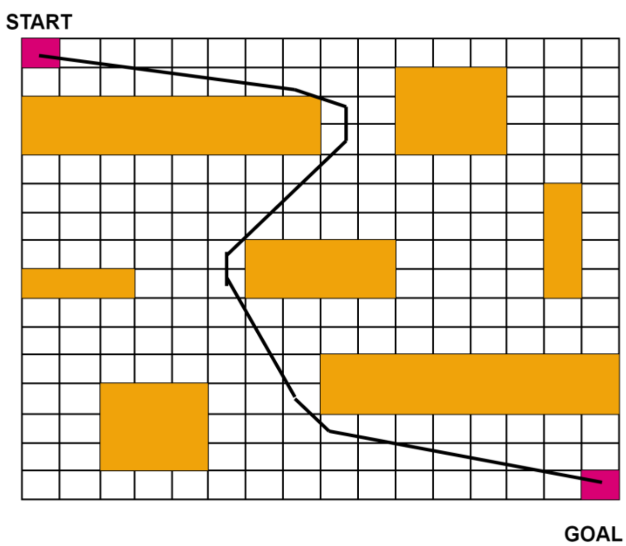


Figure 4a: Collision-free shortest path in environment 1.

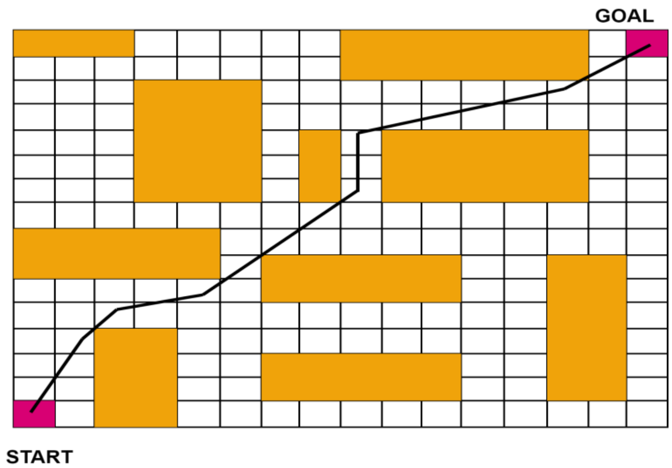
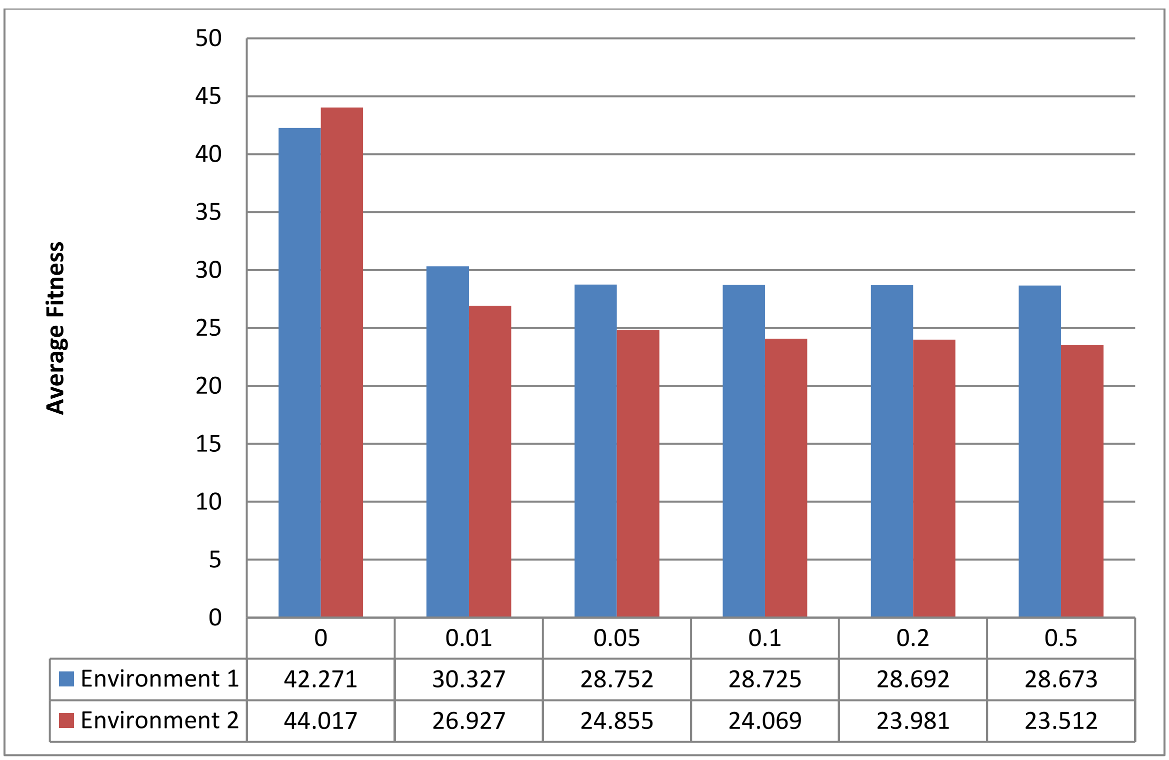


Figure 4b: Collision-free shortest path in environment 2.



	0	0.01	0.05	0.1	0.2	0.5
Environment 1	42.271	30.327	28.752	28.725	28.692	28.673
Environment 2	44.017	26.927	24.855	24.069	23.981	23.512

Figure 5: Average fitness over time in both environments.

The early fitness functions' variations are incredibly large. The variances are gradually reducing as time passes. The technique ensures that the fitness function continuously converges to satisfactory solutions. It means that, at first, the method attempts a variety of approaches to discover a resolution, and that, at the ending of each repetition it regularly converges to specific locations that are regarded to be relatively good. It indicates that the method will be on the correct course to diversify and intensify. Also at the terminating stage, we should anticipate the method to produce fairly consistent results for every occurrence.

6. Conclusion

The path design issue of a robotic system is studied in this research using an enhanced multiple objective evolutionary approaches. A path energy usage, path protection, and path length are used as appraisal predictors to create a multiple objective fitness value, as well as the methodology utilizes the heuristic mean implant technique while the formation of an early population, which efficiently enhance the efficiency of an early population and also increases the convergence speed of the technique, It reduces path length by guaranteeing that robotic constantly goes towards the goal location, reduces path energy usage, also increases the security for scheduled paths to such a significant degree. Lastly, the enhanced genetic process has been used to keep the process from entering a regional optimal too soon. Simulations demonstrate that the method could operate in a variety of contexts with varying sizes and intricacies. Our suggested approach features a reduced path length as well as a better convergence rate.

The goal of this study is to find an economical and reliable way to generate new paths for robotic systems in 2D stable surroundings. Moreover, the present condition of the arts in robotic systems suggests also that restricted resources used to calculate the optimal shortest route are being expanded to construct the route in variable surroundings in a fair amount of period. It may help us focus our studies on the next. Whenever the robotic arm is tasked to travel from source to endpoint without any predetermined path, we can use the evolving method to find a route. Furthermore, we considered the resolution region was 2D, although we may extend the dimensions for feasible paths of robotic systems. While stable contexts are used in this research, the low convergence rate indicates that suggested techniques and algorithms can be implemented to dynamical contexts where actual route planning becomes required. During future studies, the offered principles might be adapted into 3-dimensional contexts with greater restrictions for actual scenarios.

REFERENCE

1. M. A. P. Garcia, O. Montiel, O. Castillo, R. Sep´ulveda, and P. Melin, “Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation,” Applied Soft Computing, vol. 9, no. 3, pp. 1102–1110, 2009.
2. G. Li and W. Chou, “Path planning for mobile robot using self-adaptive learning particle swarm optimization,” Science China Information Sciences, vol. 61, no. 5, Article ID 052204, 2018.
3. R. K. Mandava, S. Bondada, and P. R. Vundavilli, “An optimized path planning for the mobile robot using potential field method and PSO algorithm,” Advances in Intelligent Systems and Computing, vol. 8, no. 7, pp. 139–150, 2019.
4. R. Li, W. Wu, and H. Qiao, “.e compliance of robotic hands—from functionality to mechanism,” Assembly Automation, vol. 35, no. 3, pp. 281–286, 2015.
5. D. C. Robinson, D. A. Sanders, and E. Mazharsolook, “Ambient intelligence for optimal manufacturing and energy efficiency,” Assembly Automation, vol. 35, no. 3, pp. 234–248, 2015.

6. B. Song, Z. Wang, and L. Sheng, “A new genetic algorithm approach to smooth path planning for mobile robots,” Assembly Automation, vol. 36, no. 2, pp. 138–145, 2016.
7. A. Bakdi, A. Hentout, H. Boutami, A. Maoudj, O. Hachour, and B. Bouzouia, “Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control,” Robotics and Autonomous Systems, vol. 89, pp. 95–109, 2017.
8. Y. Deng, Y. Liu, and D. Zhou, “An improved genetic algorithm with initial population strategy for symmetric TSP,” Mathematical Problems in Engineering, vol. 2015, Article ID 212794, 6 pages, 2015.
9. B. K. Patle, D. R. K. Parhi, A. Jagadeesh, and S. K. Kashyap, “Matrix-binary codes based genetic algorithm for path planning of mobile robot,” Computers & Electrical Engineering, vol. 67, pp. 708–728, 2018.
10. M. A. Contreras-Cruz, V. Ayala-Ramirez, and U. H. Hernandez-Belmonte, “Mobile robot path planning using artificial bee colony and evolutionary programming,” Applied Soft Computing, vol. 30, pp. 319–328, 2015.
11. D. Whitley, D. Hains, and A. Howe, “A hybrid genetic algorithm for the traveling salesman problem using generalized partition crossover,” in Proceedings of the International Conference on Parallel Problem Solving from Nature, pp. 566–575, Springer, Krakow, Poland, September 2010.
12. M. Albayrak and N. Allahverdi, “Development a new mutation operator to solve the traveling salesman problem by aid of genetic algorithms,” Expert Systems with Applications, vol. 38, no. 3, pp. 1313–1320, 2011.
13. G. Dantzig, R. Fulkerson, and S. Johnson, “Solution of a large-scale traveling-salesman problem,” Journal of the Operations Research Society of America, vol. 2, no. 4, pp. 393–410, 1954.
14. M. Nazarahari, E. Khanmirza, and S. Doostie, “Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm,” Expert Systems with Applications, vol. 115, pp. 106–120, 2019.
15. J. Ni, K. Wang, H. Huang et al., “Robot path planning based on an improved genetic algorithm with variable length chromosome,” in Proceedings of the 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), pp. 145–149, Changsha, China, August 2016.
16. C.-C. Tsai, H.-C. Huang, and C.-K. Chan, “Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation,” IEEE Transactions on Industrial Electronics, vol. 58, no. 10, pp. 4813–4821, 2011.
17. Li B, Li J, Tang K, Yao X (2015) Many-objective evolutionary algorithms: A survey. ACM Computing surveys.
18. Emmerich MTM, Deutz AH (2018) A tutorial on multiobjective optimization: fundamentals and evolutionary methods. Nat Comput 17(3):585–609.
19. Soon GK, Guan TT, On CK, Alfred R, Anthony P (2013) "A comparison on the performance of crossover techniques in video game," 2013 IEEE international conference on control system. Computing and Engineering, Mindeb, pp 493–498.

20. Burkowski FJ (1999) Shuffle crossover and mutual information. Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 1999, pp. 1574–1580.
21. Mooi S, Lim S, Sultan M, Bakar A, Sulaiman M, Mustapha A, Leong KY (2017) Crossover and mutation operators of genetic algorithms. International Journal of Machine Learning and Computing 7:9–12.
22. Mudaliar DN, Modi NK (2013) Unraveling travelling salesman problem by genetic algorithm using m-crossover operator. International Conference on Signal Processing, Image Processing & Pattern Recognition, Coimbatore, pp 127–130.
23. Jebari K (2013) Selection methods for genetic algorithms. Abdelmalek Essaâdi University. International Journal of Emerging Sciences 3(4):333–344.
24. Gao, M.; Xu, J.; Tian, J.; Wu, H. Path Planning for Mobile Robot Based on Chaos Genetic Algorithm. In Proceedings of the 2008 Fourth International Conference on Natural Computation, Jinan, China, 18–20 October 2008; pp. 409–413.
25. Gemeinder, M.; Gerke, M. GA-based path planning for mobile robot systems employing an active search algorithm. Appl. Soft Comput. 2003, 3, 149–158.
26. Sugihara, K.; Smith, J. Genetic Algorithms for Adaptive Motion Planning of an Autonomous Mobile Robot. In Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation, Monterey, CA, USA, 10–11 July 1997; pp. 138–143.
27. Tuncer, A.; Yildirim, M. Dynamic path planning of mobile robots with improved genetic algorithm. Comput. Electr. Eng. 2012, 38, 1564–1572.
28. Frontzek, T.; Goerke, N.; Eckmiller, R. Flexible Path Peal-Time Applications Using A*-Method and Neural RBF-Networks. In Proceedings of the 1998 IEEE International Conference on Robotics and Automation, Leuven, Belgium, 16–20 May 1998; pp. 1417–1422. 17. Juidette, H.; Youlal, H. Fuzzy dynamic path planning using genetic algorithm. Electron. Lett. 2000, 36, 374–376. 18. Miao, H.; Tian, Y.-C. Dynamic robot path planning using an enhanced simulated annealing approach. Appl. Math. Comput. 2013, 222, 420–437.
29. Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots. Int. J. Robot. Res. 1986, 5, 396–404.
30. Suresh, P., Mariyal, C., Sivasubramonia Pillai, T.V., Rajesh, K.B. and Jaroszewicz, Z., "Study on polarization effect of azimuthally polarized LG beam in high NA Lens system," Elsevier – Optik, vol. 124. Issue 21, 2013. pp. 5099-5102
31. Wang, C., Chao, K., Sivaperumal, S., Suresh, P., "Anti-PVT-Variation Low-Power Time to Digital Converter Design Using 90 nm CMOS Process" IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 28, Issue 9, 2020, pp. 2069 – 2073.
32. Rashid, A.T.; Ali, A.A.; Frasca, M.; Fortuna, L. Path planning with obstacle avoidance based on visibility binary tree algorithm. Robot. Auton. Syst. 2013, 61, 1440–1449.
33. Janson, L.; Ichter, B.; Pavone, M. Deterministic sampling-based motion planning: Optimality, complexity, and performance. Int. J. Robot. Res. 2018, 37, 46–61.
34. Karthikeyan, V., Suresh, P., Saravanakumar, U., Anil Venkata Rama Sekhar, AKM. and Srikanth, S., "Wide Band Millimeter Wave Fabric Antenna for 60GHz Applications," International Journal of Innovative Technology and Exploring Engineering, Vol. 8, Issue 9, 2019, pp. 1211-1214.

35. M. Albayrak and N. Allahverdi, “Development a new mutation operator to solve the traveling salesman problem by aid of genetic algorithms,” Expert Systems with Applications, vol. 38, no. 3, pp. 1313–1320, 2011.
36. L. Fang, W. Susilo, Y. Ren, C. Ge, and J. Wang, “Chosen public key and ciphertext secure proxy re-encryption schemes,” International Journal of Digital Content Technology and Its Applications, vol. 4, no. 9, pp. 151–160, 2010.
37. C. Ge, W. Susilo, Z. Liu, J. Xia, P. Szalachowski, and F. Liming, “Secure keyword search and data sharing mechanism for cloud computing,” IEEE Transactions on Dependable and Secure Computing, vol. 99, pp. 1–1, 2020.
38. X. Zhang, Y. Zhao, N. Deng, and K. Guo, “Dynamic path planning algorithm for a mobile robot based on visible space and an improved genetic algorithm,” International Journal of Advanced Robotic Systems, vol. 13, no. 3, p. 91, 2016.
39. C. Lamini, S. Benhlima, and A. Elbekri, “Genetic algorithm based approach for autonomous mobile robot path planning,” Procedia Computer Science, vol. 127, pp. 180–189, 2018.
40. Suresh, P., T Venkata Prasad, B Uday kiran Yadav, SV Narayana Reddy, S Yaswanth Reddy, R.Rajkumar, U.Saravanakumar, 2020 “Dynamic Partial Reconfiguration Performance Analysis In Field Programmable Gate Array For Streaming Applications,” International Journal of Scientific & Technology Research, Vol. 9, Issue 02, 2020, pp. 3098-3102.
41. K. P. Cheng, R. E. Mohan, N. H. Khanh Nhan, and A. V. le, “Multi-objective genetic algorithm-based autonomous path planning for hinged-tetro reconfigurable tiling robot,” IEEE Access, vol. 8, no. 1, pp. 121267–121284, 2020.