

How to Cite:

Suresh, K. S., Venkatesan, R., & Venugopal, S. (2022). Multi-objective optimization for solving mobile robot path planning problem employing hybridization of algorithms. *International Journal of Health Sciences*, 6(S5), 1973–1992. <https://doi.org/10.53730/ijhs.v6nS5.9058>

Multi-objective optimization for solving mobile robot path planning problem employing hybridization of algorithms

Suresh K.S.

School of Computing, SASTRA Deemed University, Tamilnadu, India

Dr. Venkatesan R

School of Computing, SASTRA Deemed University, Tamilnadu, India

Dr. S. Venugopal

Director, National Institute of Technology, Nagaland, India

Email: kssuresh@cse.sastra.ac.in

Abstract--Mobile Robot Path Planning problem (MRPPP) is the most prominent research area employed in different real-time environments. In this article, a hybrid strategy is employed and validated with different environments for the path planning of mobile robots. The Artificial Potential Field (APF) algorithm and Multi-objective Genetic Algorithm (MOGA) are hybridized to solve an MRPPP. In the proposed hybrid methodology, the implementation is accomplished in three phases. To generate the initial population, an Artificial Potential Field (APF) algorithm is applied to discover all viable routes in an environment between the start and destination points. The collision-free paths are generated by determining the artificial force produced by obstacles and the target. The population-based evolutionary algorithm is employed to derive an optimal solution path from the initial population consisting of candidate paths. The selection operator is used to select the qualified subpopulation to the next generation to converge into the optimal solution. The genetic operators such as two-point crossover and mid-point mutation are exercised which are specific to the path planning problem. The rank of the paths, length of the path, number of turns, and degree of turns are considered as the multiple objectives to employ non dominated sorting genetic algorithm II (NSGA-II) to find an optimal solution by parallel optimizing all the objectives. Simulations have been conducted and compared by applying the proposed methodology for different environments and initial paths. The efficiency of the methodology is proved through the comparison of the percentage of performance improvement in each objective as well as overall fitness value.

Keywords---APF, Multi-objective Genetic algorithm, Pareto optimization, mobile robot path planning, genetic operators, elitism.

Introduction

In the automation of industries, the development of robotics played an inevitable role, where MRPPP is a significant domain for research. A lot of research has been accomplished to achieve the optimal results to reach the target from the source point. It provides scope to achieve in different ways since this research domain encases almost all the engineering aspects. As the nature of the RPP is categorized as an NP-hard problem[1] due to its complexity, the computational cost and the time to derive the solution path are undesirable using a deterministic approach[2] though it may result in an accurate solution if exist. The non-deterministic approach provides the optimized solution to obtain the solution within a stipulated time or number of generations maybe with a near-optimal solution. The evolutionary algorithms are population-based meta-heuristic algorithms[3] where the quality of the solution is determined by the operators implemented to derive the solution. Though the evolutionary algorithms are outperforming[4] the deterministic algorithms for the large solution space, the solution derived by applying evolutionary algorithms is near-optimal. The Multi-Objective Genetic Algorithm is a population-based algorithm[5], which uses randomly generated huge solution space as an initial population to find the optimal solution. Genetic Algorithm (GA) approach is the most reliable search method for complex optimization problems, because of the inherent nature of the algorithms. Exhaustive experiments are done for finding optimal path employing GA with different variations in phases of execution and number of generations for MRPP while considering only one objective as the length of the path. If more than one objectives are to be accounted for, they are either exercised as a constraint or adding weightage for the objectives and is employed. In all these techniques, only one optimal solution is derived. It requires domain-specific knowledge, because a small deviation from the value may affect the quality of the solution. By applying MOGA, the set of non dominated solutions can be obtained which gives freedom to decide according to the decision makers' preferences by considering the tradeoff between the objectives. But the computational complexity of the methodology increases to apply the pareto optimization technique for the huge population size. The proposed methodology takes the advantage of the Pareto optimality principle at the same time, to overcome the drawbacks of the Pareto optimization, it is hybridized with another methodology called Artificial Potential Field algorithm which is utilized for generating most effective initial population. Due to the initial population less in size, applying Pareto optimization reduces the computational complexity.

The following section 2 introduces the phases of the proposed methodology and in section 3 the APF algorithm is illustrated. Section 4 describes the implementation of each phase of the GA for MRPP. Implementation of Pareto optimization is elaborated in section 5 and in the following section 6, the principle of Elitism is discussed. In section 7 and its sub-sections, the results obtained from experiments for different environments are discussed. Section 8 enumerates the inferences from the results followed by a conclusion in section 9.

Three-phase Hybrid methodology for MRPP

The proposed algorithm includes three phases:

Phase 1: Assign the potential to the nodes and enumerate randomly generated initial feasible paths using the Artificial Potential Field algorithm.

Phase 2: Upgrade the initial set of feasible paths using Genetic Operators.

Phase 3: Apply the Pareto optimization technique and elitism to achieve the optimal solution path.

Phase 1: Generating Initial Population by Artificial Potential Field Algorithm

According to the ideology of the Artificial Potential Field algorithm, a potential field matrix is generated by assigning suitable potential to all the nodes in the environment starting from the destination node which helps in determining all possible paths between start and destination locations[6][7].

Phase 2: Employing GA principle on the initial population

While applying the Genetic algorithm (GA) principle, the RPP problem[5][8] is a special case as compared to the other domain problems because the crossover and mutation operators cannot be applied as in other problems.[9][10] Therefore, the non-conventional crossover and mutation operators are applied on the selected paths to produce offspring.

Phase 3: Applying Pareto Optimization methodology

The population obtained after the genetic operations are ranked based on their domination count using Pareto Optimization.[11][12] The required number of optimal paths is chosen from the resulting sets of ranked paths likely to have the best paths resulting from the objective evaluation. Pareto optimization is applied to rank the best paths to find an optimal solution.

Proposed Hybrid Methodology

The main aim of this article is to implement the hybrid methodology to determine the optimal path for the MRPPP in the context of MOGA. The objectives are to be defined either it may have to be minimized or maximized while implementing the hybrid methodology. In deriving the solution for the MRPP problem, the following are taken into account for attaining the resultant path.

- a. The length of the path is considered a significant objective. Since the length of the navigational path influences the time to reach the target, which is the prominent objective for any MRPP problem[13]-[16].
- b. In a real-time environment, in addition to the length as an objective, the number of turns[17] of the individual path during the traverse over the path.
- c. The smoothness of the path is another objective that is measured in terms of the degree of turning from one point to another point in the navigational path. This objective is desirable, because the lower degree of turning and a higher degree of turning may lead to the toppling of the Mobile Robot. So, this objective is related to the safety of the Mobile Robot.[14], [18]
- d. Another objective energy consumption[19] to reach the goal point is closely related to the number of turns and degree of turns of the subjective paths.

Substantial research has been done to get multiple objective solutions not only in the case of the MRPPP but also experimented in other domains in different ways. While solving the MRPPP, the length of the path may be considered as an objective and the remaining objectives may be accounted as constraints for the model. But in this case, only a single solution will be obtained which will not provide the choice for decision-makers to find the tradeoff between a greater numbers of solutions with multiple objectives. Hence, in this proposed methodology, the Pareto optimality principle is adopted to achieve the solution for the MRPPP with multiple objectives[17], [20]–[22][23].

In the case of the RPP problem, the solution space can be generated randomly but the solution space may contain infeasible paths in the population. While applying the GA, the infeasible paths reduce the efficiency of the algorithm by increasing the convergence time. Hence in this proposed methodology, to improve the lack of efficiency, the candidate paths in the solution space are not generated randomly. The population is generated with feasible solution space by exercising the potential field methodology which will reduce the convergence time to attain the solution. Specifically, in this investigation, the intelligent population is created before applying a MOGA where all the paths in the solution space are feasible.

The proposed methodology takes advantage of a hybrid environment modeling approach. The environment is designed with equally spaced nodes forming square grids[24]. The Figure represents the navigational space of the mobile robot environment comprises free space, obstacles, starting point, and goal point of the Robot. In this figure, the environment is denoted as $n \times n$ discretized square grids with a unit length.

According to the principle of the APF algorithm[25], [26][27], a proper potential value is assigned to all the uniformly distributed $n+1$ nodes. The MR uses the pre-calculated potentials to choose the proper set of nodes and trace the feasible paths between the start and the destination locations in the environment. Figure 1 shows the sample environment with 10 X 10 grids which form the 11 X 11 nodes. The black stars denote the equally spaced nodes. The green circle indicates the starting point of the MR and the red plus signal to specify the destination. The blocks are to symbolize the obstacles in the environment.

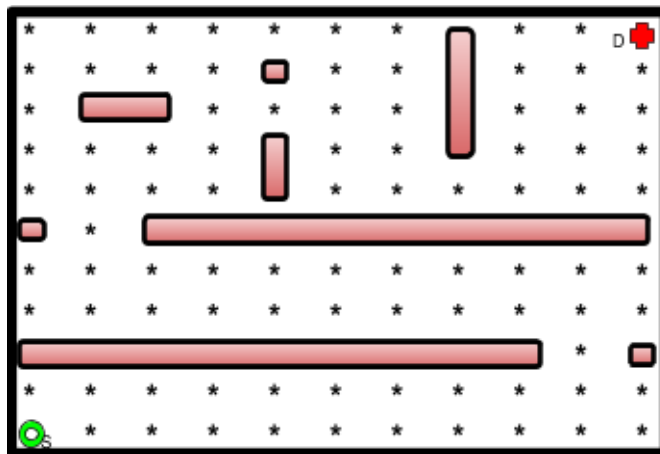


Figure 1 Sample 10×10 environment with equally spaced nodes, obstacles, starting point, and target point

Each feasible initial path in the parent population is represented by the list of coordinates of each node. As each path traverses through a different number of nodes to reach the destination, the resulting chromosomes after genetic operations will be of variable length. Two-point crossover and mid-point mutation processes are employed. Pareto optimization is applied to rank the best paths to find an optimal solution.

Artificial Potential Field (APF) Algorithm:

For a $n \times n$ environment, the destination node is taken as (n, n) . The arbitrarily chosen largest potential value is assigned to the target node. While traversing through its adjacent nodes, the potential of the nodes decreases gradually. This iteration goes on until all the nodes are assigned a proper potential which leads to a potential grid, through which the paths between the start and destination nodes are found. In the initial phase of the proposed algorithm, APF is implemented to generate the initial population to apply MOGA [21], [28], [29][11]. In the APF implementation, three lists are used for allocating potential to the nodes in the environment.

a. List NPL:

All the designated nodes of the environment which are not assigned with appropriate potential are added to the list.

b. List VPL:

The value as the potential allocated for each node is appended in this list.

c. List APL:

In this list, the potential of each node and the potential value of the neighbor nodes are estimated and included.

Initially, all nodes are inserted into the NPL list. Then the destination node is taken out from the NPL list and the largest potential P_{large} is assigned to that node with an arbitrarily chosen value. The value of the largest potential is determined based on the size of the environment as well as how the environment is partitioned. After assigning the largest potential to the destination, the nodes representing the obstacles are identified and taken out from the NPL. The obstacle

nodes are considered as the blocked nodes because the nodes cannot be used for the navigation of the MR. Therefore, those nodes are assigned with artificial potential values as $-P_{small}$ and are appended to the APL list. Then, starting from the destination, the potential of $P_{node} = P_{large} - D_{step}$ (decrement step) is estimated and assigned to all its neighbors that exist in the NPL list. After assigning estimated values to the adjacent nodes, these will be inserted in the VPL list. The same nodes are removed from the NPL list at the same time the destination node will be eliminated from the VPL list and placed in the APL list. At each iteration, the elements of all the above said lists are manipulated.

For example, at the i^{th} iteration, the following changes are taken place in the lists.

- a. The node from the VPL list is extracted.
- b. The potential of the extracted node is evaluated as

$$iP_{node} = iP_{node} - 1 - D_{step}$$
- c. The potential value is allocated to their adjacent nodes which are in the NPL list.
- d. The nodes in the VPL list are excluded from it and placed in the APL list.
- e. The nodes which are assigned with a potential value are deleted from the NPL list and inserted in the VPL list.

Repeating this procedure until all the nodes are assigned a potential will result in the potential grid of the environment as shown in Figure 1. The parameters P_{small} , P_{large} , and D_{step} are intuitively chosen and assigned based on the size of the environment.

In the process of creating the initial population, all feasible paths are generated by moving from the start position to the adjacent node with higher potential and continue traversing to the node with the largest potential nothing but the destination node which builds a feasible path between start and destination locations is found at this point. The potential gradually increases as it moves along the path till reaching the destination.

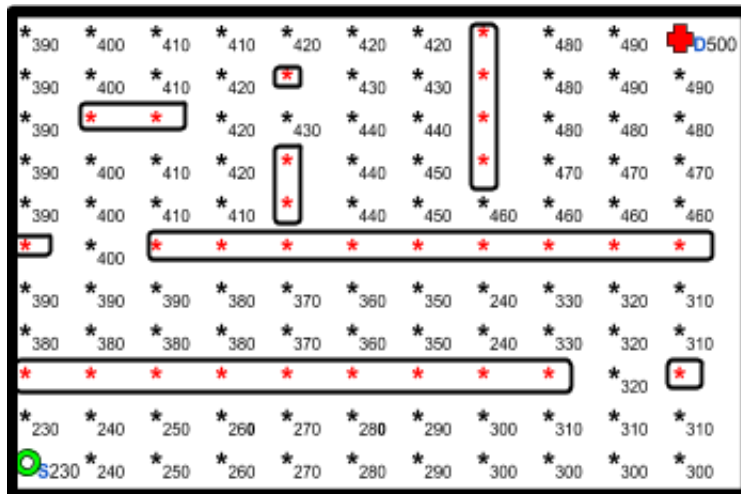


Figure 2 Sample environment after assigning potential using the APF algorithm

When traversing towards the largest potential, if more than one adjacent node has the same potential, then that will lead to the subpaths.[Figure 3] So, the searching for feasible paths will continue with all the subpaths. If the searching continues the initial population with feasible paths will be formed. The genetic algorithms can be now applied to this resulting initial population.

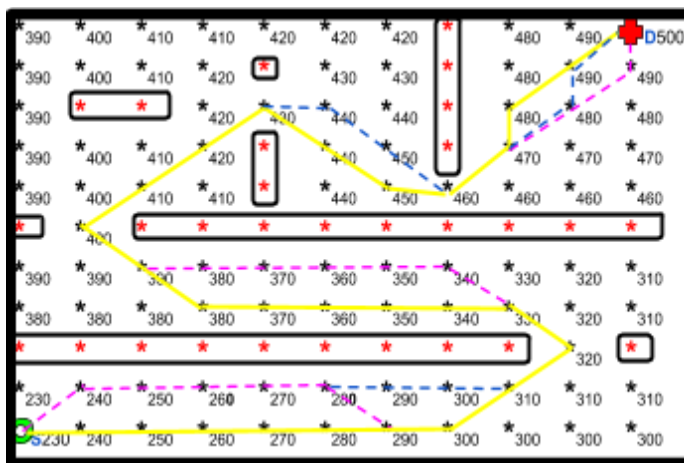


Figure 3 Feasible paths generated using APF algorithm

Genetic Algorithm Procedure for MRPPP

On the initial population, the genetic operators are exercised to get the optimal solution. Since the MRPPP is the special case of the research, the operators cannot be applied randomly as in the other domains. If the operators are employed randomly, the resultant paths may not be the feasible solution paths. If the unfeasible paths are added to the population, then the convergence time will increase due to the noisy population. Hence, they cannot be added to the population for further generations. Therefore, the GA operators are to be customized to the MRPPP[30]. Additionally, it is involved with variable length; hence the genetic operators are to be employed cautiously. In this article, a two-point crossover and a midpoint mutation are applied on the variable-length chromosomes to produce offspring. The number of generations to be iterated is decided by either of the two conditions. First, the number of generations may be fixed at the starting of the execution. Second, after applying the GA operators in every generation, if the individuals in the population repeat, the execution may be terminated.

Selection Operator

Initially, the paths generated using the Artificial Potential Field algorithm is sorted based on their length, the number of turns, and the penalty for vulnerability. The best of individuals are selected for the next generation which will undergo further operations.

Crossover Operator

The crossover operator is used for the diversity of the population by stochastically combined existing individuals of the population produces offspring to generate the new population set. In this implementation, the two-point crossover operator is employed to create offspring not by combining randomly selected individuals. Before performing crossover on the population select the paths with common nodes of the two parents. Then, it sorts all the common nodes with respect to their x or y coordinate and two points are selected arbitrarily, each from the first and second half of the sorted common nodes.

Let m and n be the two parents,

The first common point of parent m = CP_{m1}

The first common point of parent n = CP_{n1}

The second common point of parent m = CP_{m2}

The second common point of parent n = CP_{n2}

Two new offsprings are generated as follows after crossover operation

NP_{1m} (New Path 1m) = m[nodes up to CP_{m1}] + n[nodes from CP_{n1+1} to CP_{n2}]
+ m [nodes after CP_{m2+1}]

NP_{2n} (New Path 2n) = n[nodes up to CP_{n1}] + m[nodes from CP_{m1+1} to CP_{m2}]
+ m [nodes after CP_{n2+1}]

where NP_{1m} and NP_{2n} are the two offsprings generated from the m and n paths.

For the first offspring NP_{1m} , the path is realized from the following nodes.

m[nodes up to CP_{m1}] - nodes of the first path up to a first common node
including common node

n[nodes from CP_{n1+1} to CP_{n2}] - nodes of the second path nodes from the first
common node to second common node excluding the first common
node and including second common node

m [nodes after CP_{m2+1}] - nodes of the first path from a second common node
excluding second common node

Similarly for the second offspring NP_{2n} is formed with the following three segments
of nodes respectively.

n[nodes up to CP_{n1}], m[nodes from CP_{m1+1} to CP_{m2}] and m [nodes after CP_{n2+1}]

Mutation Operator

The midpoint mutation operator creates a new offspring by finding the midpoint of a path's any random node. A random node is selected from a path and that node's previous and next non-obstacle node's midpoint are taken. And the resultant midpoint result is the mutated node.

Here,

x_1, y_1 = previous node's x and y values

x_2, y_2 = next node's x and y values

x, y = node to be mutated

$m(x,y)$ = mutation point in the path

$$m(x, y) = \left[\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2} \right]$$

Fitness Function

Three objectives are taken to measure the fitness of the i^{th} path namely length of the path (L_i), the number of turns in the path (T_i), and the degree of turns of the path (D_i).

Initially, $L_i = 0$, $T_i = 0$, $D_i = 0$

Length (L_i) of the path (P_i) = $\sum_{j=1}^n L_j + 1$ where n = number of nodes in the via points.

Number of turns (T_i) of the path (P_i) = $\sum_{j=1}^n T_j + 1$

where $T_j = \angle(n[x], n[y]) \neq 180^\circ$

angle between the nodes x and y which are consecutive nodes of the path.

Degree of Turns (D_i) of the path (P_i) = $\sum_{j=1}^n D_j + 1$,

Where $D_j = 1$, if $\angle(n[x], n[y]) = 135^\circ$ and 225°

$D_j = 2$, if $\angle(n[x], n[y]) = 90^\circ$ and 270°

$D_j = 3$, if $\angle(n[x], n[y]) = 45^\circ$ and 315°

Otherwise $D_j = 0$

Using the above functions the objective value is calculated for each objective.

Pareto Optimization

Pareto optimality is a measure of efficiency in the multi-objective sense where many competing objectives have to be taken into account in an optimization process[31]. Pareto optimization is defined as if there is no other path that increases the performance of any of its objective parameters without degrading at least one of the other criteria[23]. A path is said to be Pareto optimal or non-dominated if no other feasible path dominates it. The Pareto front is formed by the set of all non-dominated paths[32] considering a trade-off between all objectives.

In this algorithm, after each generation of genetic operations, the child population generated is added with the parent population. Then the acquired population is ranked based on their non-dominated feature. Considering the three objectives namely path length, the number of turns and smoothness in terms of penalty, a list denoted as *Dominated_Paths* is used to maintain the paths dominated by each path and a variable *Dominated_Count* is used to retain the count of each path being dominated by others. A path P_1 is compared with a path P_2 , if the path P_2 satisfies the dominating condition, then P_2 is inserted in the *Dominated_Paths* list. At the same time, the *Dominated_Count* of path P_2 gets incremented. In this paper, the dominating condition is satisfied, if all the objectives of one path come out being 'superior or equal' when compared with another and it emerges with at least one objective being 'superior' to the other. This iteration continues till all the paths in the population are compared once. First, the paths with *Dominated_Count* value zero are categorized as RANK 1 and their *Dominated_Count* value is decremented. Then, the *Dominated_Count* value of the paths in the *Dominated_Paths* list of RANK 1 paths is also decremented, so that the next set of paths with *Dominated_Count* value zero are categorized as RANK 2 and letting the iteration continue until all paths are ranked. The algorithm of the Pareto optimization technique is given in the Algorithm.

Algorithm 1: Pareto optimization for population ranking

P_{ov} - Holds the objective values of each path

Dominated_Paths - Holds the paths dominated by each path

Dominated_Count - Holds the count of each path being dominated by other

Rank - used to store the paths associated with their ranks

Population - Holds both parent and child population which is generated through GA

Initialization

```

    for x equals each row in Pov
      for y equals each row in Pov
        if x.index = y.index then continue
          endif
          if x dominates y then
            add y.index to Dominated_Paths [x.index]
            Dominated_Count [y.index] is incremented
          endif
        endif
      endfor
    endfor

    for i=0 to Population.length
      for j=0 to Population.length
        if Dominated_Count[j] = 0 then
          Add j to Rank[i]
          Dominated_Count[j] is decremented
        endif
      endfor
    endfor

    for m equals each element in rank[i]
      for n equals each element in Dominated_Paths[m]
        Dominated_Count [n] is decremented
      endfor
    endfor
  endfor

```

Implementation of elitism

In genetic algorithms, operators such as selection, crossover, and mutation are implemented to produce next-generation offspring. To enhance the performance of the GA, an ideology called Elitism[28] is implemented. The basic concept of Elitism is to pass the current generation's best individuals to the next generation to maintain the efficacy of the generation in turn the convergence time is reduced as compared to the nonelitism principle. According to the principle of elitism, a small fraction of the fittest candidates is copied to the next generation. This process keeps the population with a high probability of fitness. The candidate solutions that are sustained unchanged by elitism remain eligible for selection as parents when the rest of the next generation is breeding. When using elitism, the elitism rate needs to be specified, so that the required number of the best chromosomes can be retained unaffected and selected for the next generation. For example, if the elitism rate is 0.05 percent for the given population of 100 individuals, the five best individuals of the current generation are transferred to the upcoming generations. This consolidated population will undergo the genetic operators to generate the other individuals.

The crowding distance(CD) is a strategy developed by Deb [28]and used in NSGA-II (Non-dominated Sorting Genetic Algorithm II)[32]. This technique ensures the diversity of solutions by calculating the density of the solution in the Pareto optimal front. The estimated solution density within the neighborhood of each population is the indicator for the diversity in the Pareto optimal solution. The CD is estimated by taking the sum of the normalized distances for each individual between two adjacent solutions, relative to all objectives in a multi-objective environment[29]. The solution with the highest value for the overcrowding distance is the least crowded solution.

$$cd_m^i = \frac{(f_m^{i-1} - f_m^{i+1})}{(f_m^{max} - f_m^{min})}$$

where $i = i^{\text{th}}$ solution's fitness value
 $m = m^{\text{th}}$ objective

Therefore cd_m^i - distance from the neighbor's for the i^{th} solution is calculated

f_m^{i-1} - Fitness value of the $(i-1)^{\text{th}}$ (previous) solution for the m^{th} objective

f_m^{i+1} - Fitness value of the $(i+1)^{\text{th}}$ (next) solution for the m^{th} objective

f_m^{max} - Maximum fitness value of the m^{th} objective

f_m^{min} - Minimum fitness value of the m^{th} objective

$$CD_i = \sum_{m=1}^M cd_m^i$$

where $CD_i =$ sum of crowding distance of i^{th} solution for all m objectives.

By estimating all the crowding distance values, by comparing the CD_i values the one which satisfies the below condition will be rejected.

$$CD_i < CD_j$$

where CD_i and CD_j are crowding distances of two different solutions.

Algorithm 2:

Elitism for further selection.

crowding (path)

for each element in the path

 compute crowding distance

 endfor

sort paths in descending order with respect to crowding distance

return (needed paths in order)

Sort the population based on their distance value and select the solutions and add it to the current population to implement the next generation.

The selection process to create the next generation will be started with the randomly selected pair of paths from the given population. Two scenarios are possible while comparing two paths.

- a. Paths are in the same front or
- b. Paths are in a different front.

The case (b), if both are on a different front, the top-ranked path will be selected for the next generation. (the front with the lowest rank value).

On the other hand, the case (a), if both are in the same front, the path to be selected is based on the CD value. The path with the highest CD will be selected for the next generation.

Results and Discussions

In this section, three different sample scenarios are tested to analyze the performance by implementing the proposed algorithm.

- 10 x 10 Environment with 500 paths
- 13 x 13 Environment with 1000 paths
- 15 x 15 Environment with 5000 paths

In the sample scenarios, the number of paths are generated using APF depends on the size of the environment. To analyze the implementation, the minimum, maximum, average, and standard deviation of the objective values are evaluated for each objective. The inference is deduced by comparing the above parameters.

10 x 10 Environments

Figure 4 shows the sample 10 x 10 environment with 500 initial feasible paths generated by applying the APF algorithm besides that the minimum and maximum lengths, number of turns, and degree of turns of the initial feasible paths. Here, the minimum lengths, number of turns, and degree of turns are 30, 12, 31 respectively. After applying the GA and Pareto Optimization Technique, the paths with better objective values are obtained. Among them, the optimal path with minimum length, number of turns, and degree of turns is chosen as shown in Figure 5.

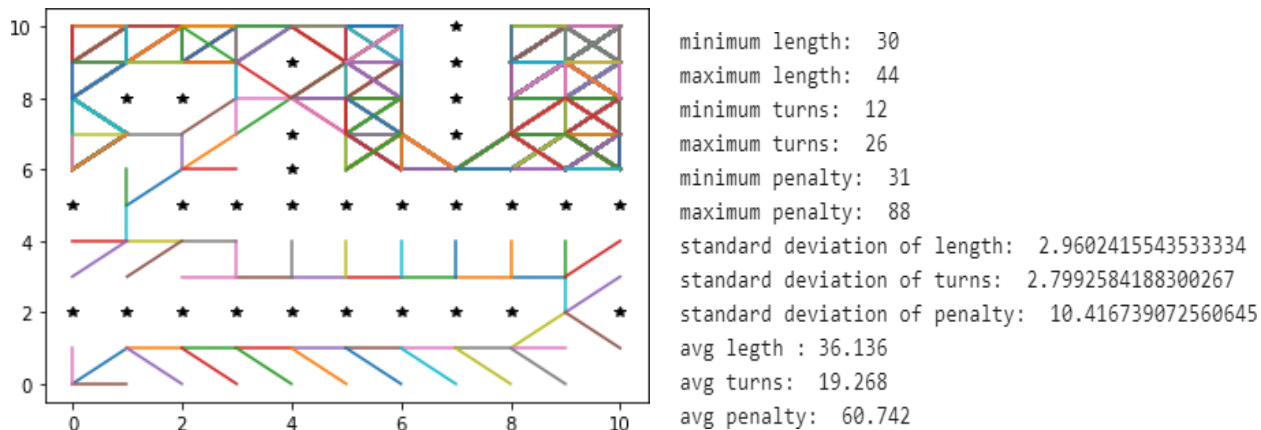


Figure 4 Sample environment with 500 paths corresponding Values for the parameters to compare the results

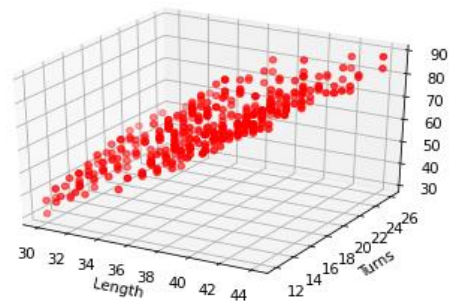
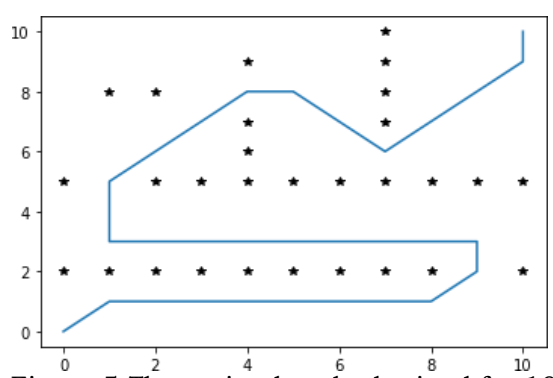


Figure 5 The optimal path obtained for 10 x 10 environment and the objective values for the corresponding objectives

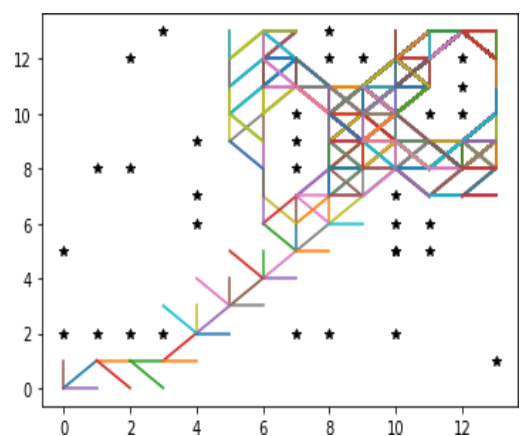
Comparison between APF and hybrid algorithm with minimum of length, turns and penalty values

	APF Algorithm	Hybrid Algorithm
Length	30	30
Turns	12	10
Penalty	31	27

Figure 6 Comparison of objective values for 10 x 10 environment

13 x 13 Environment

Error! Reference source not found. shows the sample 13 x 13 environment with 1000 initial feasible paths generated by applying the APF algorithm. Also, **Error! Reference source not found.** shows the minimum and maximum lengths, number of turns, and degree of turns of the initial feasible paths. Here, the minimum lengths, number of turns, and degree of turns are 16, 5, 15 respectively. After applying GA and Pareto



```

minimum length: 16
maximum length: 27
minimum turns: 5
maximum turns: 17
minimum penalty: 15
maximum penalty: 67
standard deviation of length: 2.760503464837416
standard deviation of turns: 2.6993817184054687
standard deviation of penalty: 10.55132818779354
avg length : 19.671
avg turns: 10.925
avg penalty: 40.298
    
```

Figure 7 Sample environment with 1000 paths corresponding Values for the parameters

Optimization Technique, the paths with better objectives are obtained. Among them, the optimal path with minimum length, number of turns, and degree of turns is chosen as shown in Figure 8.

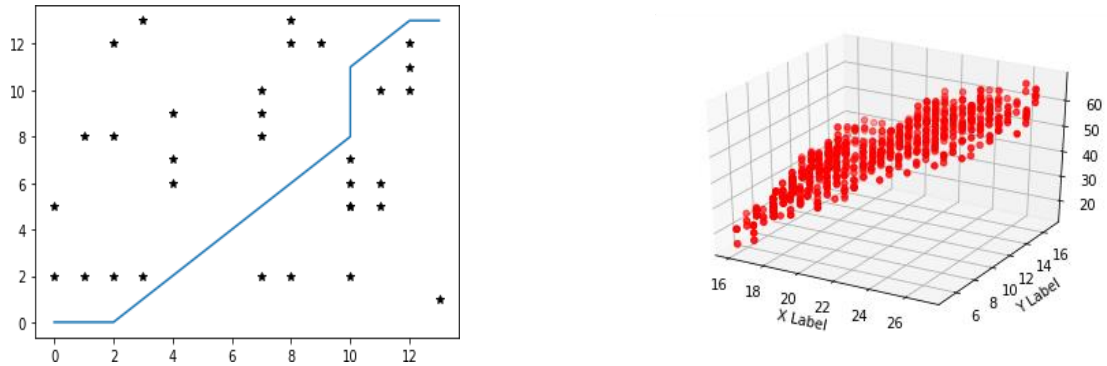


Figure 8 The optimal path obtained for 13 X 13 environment and the objective values for the corresponding objectives

	APF Algorithm	Hybrid Algorithm
Length	16	16
Turns	5	4
Penalty	15	12

Figure 9 Comparison of objective values for 13 x 13 environment

15 x 15 Environment

Figure 10 shows the sample 15 x 15 environment with 5000 initial feasible paths generated by applying the APF algorithm. Figure 11 shows the minimum and maximum lengths, number of turns, and degree of turns of the initial feasible paths. Here, the minimum lengths, number of turns, and degree of turns are 19, 5, 15 respectively. After applying GA and Pareto Optimization Technique, the paths with better objectives are obtained. Among them, the optimal path with minimum length, number of turns, and degree of turns is chosen as shown in Figure 12.

Validation of output of employing APF algorithm

After the implementation of the APF methodology, from the observed outcomes, the following statistical data are tabulated which justify the relationship between the different environments. The Standard Deviation (SD) of the fitness values for the objectives confirms the objective values for the derived paths are very low, proved the paths derived after applying the APF algorithm are so close to the optimal solution.

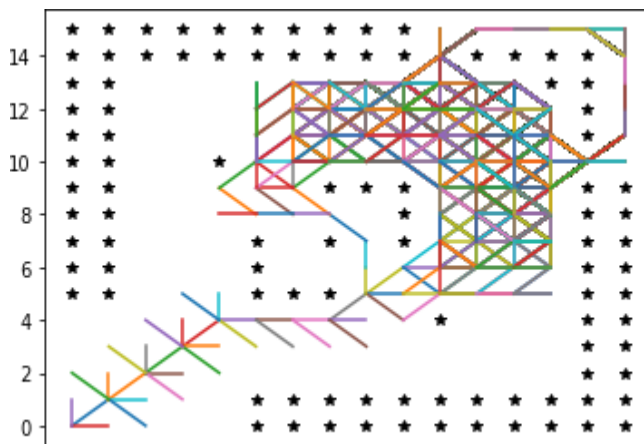


Figure 10 Sample environment with 5000 paths

```

minimum length: 19
maximum length: 38
minimum turns: 5
maximum turns: 24
minimum penalty: 15
maximum penalty: 110
standard deviation of length: 3.147462797939458
standard deviation of turns: 2.738376778581847
standard deviation of penalty: 13.472762103731393
avg legh : 26.5382
avg turns: 13.1606
avg penalty: 58.3206
    
```

Figure 11 Values for the parameters to compare the results

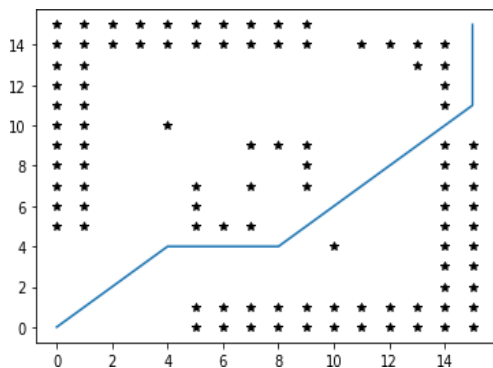
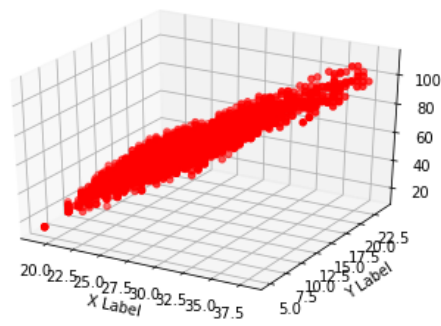


Figure 12 The optimal path obtained for 15 X 15 environment and the objective values for the corresponding objectives



	APF Algorithm	Hybrid Algorithm
Length	19	19
Turns	5	3
Penalty	15	9

Figure 13 Comparison of objective values for 15 x 15 environment

It is observed from Table 1 that the Standard Deviation of length, turns and penalty of the initial feasible paths, are low. Because the initial population selected itself is a better population which decreases the convergence time. This is a significant parameter to validate the efficiency of the algorithm and this is achieved as the APF algorithm is used. After applying a combination of GA and Pareto Optimization Algorithms the average length, turns and penalty of the population is much more optimized. This is proved by comparing the average length, turns, and penalty with the counterpart of the best path.

Table 1
Inference of the results from the execution

OBJECTIVES	10 X 10 ENVIRONMENT (500 Initial Paths)	13 X 13 ENVIRONMENT (1000 Initial Paths)	15 X 15 ENVIRONMENT (5000 Initial Paths)
SD OF LENGTH	2.96	2.76	3.14
SD OF TURNS	2.79	2.69	2.73
SD OF PENALTY	10.41	10.55	13.47
AVERAGE LENGTH	36.136	19.671	26.538
AVERAGE TURNS	19.268	10.925	13.160
AVERAGE PENALTY	60.742	40.298	58.320

Validation through Test

The Student's t-test is applied to the values obtained from the APF algorithm which is used to generate the initial population. This is to justify whatever may be the environment size and obstacle density, there is a correlation between the values shown in Table 1

Performance analysis

Figure 14 depicts the consolidated comparison best path attributes between the APF and Hybrid methodology for all the three environments sizes namely 10 x 10, 13 x 13, and 15 x 15. The following Figure 15 shows the percentage of performance improvement by applying Hybrid methodology over the APF algorithm for the objectives for three different environments. From the values

derived and the performance measurement evaluated as a percentage, it is evident that the proposed Hybrid methodology gives better performance than applying APF methodology alone. In addition to the increase in individual increase in the objective values, the overall improvement is calculated by adding the individual values. It is observed that in terms of absolute values when the size of the environment increases the fitness value which is computed as sum of the objective values decreases. To emphasize the reliability of the result, the percentage of change in the objective value of the APF and Hybrid methodology is compared. The Figure 15 justifies that though the change of the length is 0, the other objectives shows the considerable increase in the performance while the size of the environment increases.

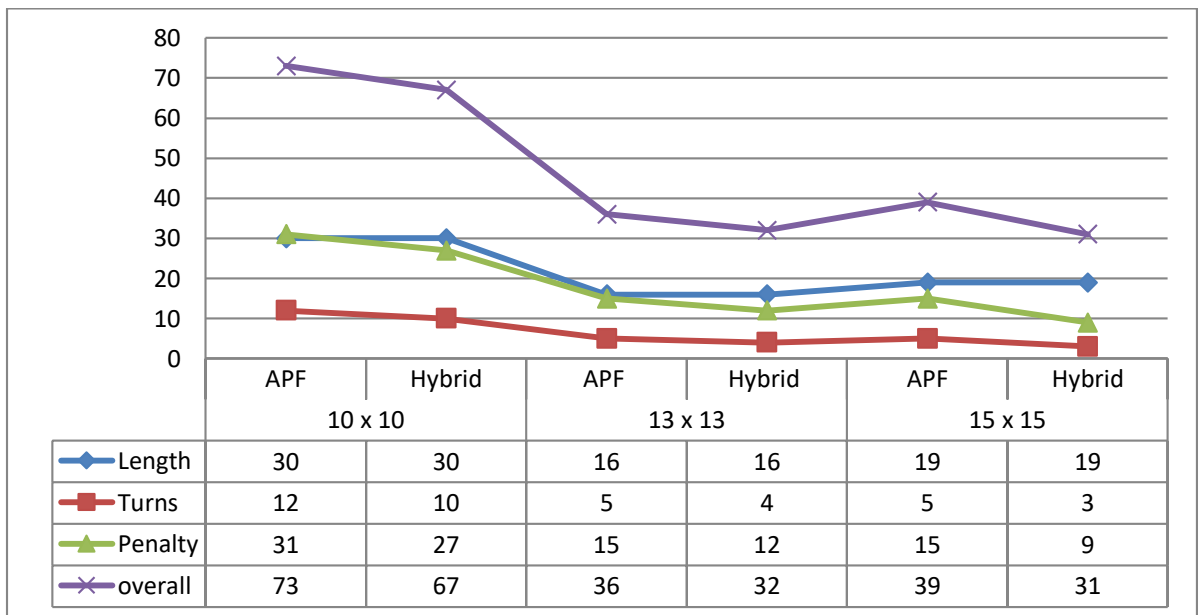


Figure 14 Comparison of objective values for APF vs Hybrid

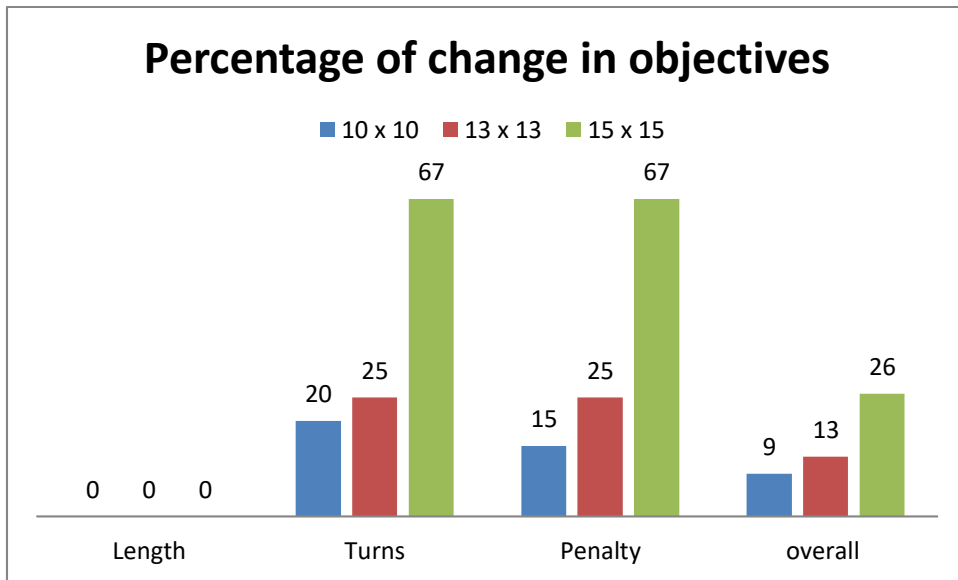


Figure 15 Percentage of improvement APF vs Hybrid

Conclusion

The proposed hybrid strategy is implemented to simulate the mobile RPP problem. It assures to determine an optimal path in any complex environment between the predefined start and destination locations. Instead of randomly generating the paths, first, the APF was used to build a potential grid, making the problem of finding all the feasible paths easier. Next, the improved GA enhances the initial paths obtained by applying the APF algorithm to find the optimal path for multiple objectives such as length, number of turns, and degree of turns. Employing the conventional genetic operators are not relevant to solve the RPP problem. In this proposed methodology the two-point crossover and mid-point mutation which is appropriate for the RPP problem are used to derive the solution. Multiple objectives are considered to obtain the optimal path such as path length, number of turns, and degree of turns. The population is ranked using Pareto optimization and selected using the Elitism process for further generations. The best path for different configurations is obtained over generations. The results are analyzed with statistical data derived from the implementation and the objective values are compared to validate the proposed methodology.

References

- [1] P. Raja, "Optimal path planning of mobile robots: A review," *Int. J. Phys. Sci.*, vol. 7, no. 9, pp. 1314–1320, 2012, doi: 10.5897/ijps11.1745.
- [2] M. N. and S. D. E. Khanmirza, M. Haghbeigi, "A Comparative Study of Deterministic and Probabilistic Mobile Robot Path Planning Algorithms," *5th RSI Int. Conf. Robot. Mechatronics*, pp. 534–539, 2017, doi: 10.1109/ICRoM.2017.8466197.
- [3] K. S. Suresh, V. Vaithyanathan, and S. Venugopal, "Emergence of Meta Heuristic algorithm principles in the field of robot motion planning-A survey

- of paradigm shift," *Int. J. Appl. Eng. Res.*, vol. 8, no. 20 SPEC. ISSUE, pp. 2717–2722, 2013.
- [4] W. J. Suh, C. S. Park, and D. W. Kim, "Heuristic vs. meta-heuristic optimization for energy performance of a post office building," *Proc. Build. Simul. 2011 12th Conf. Int. Build. Perform. Simul. Assoc.*, pp. 704–711, 2011.
- [5] K. S. Suresh, K. S. Ravichandran, S. Ananthkrishnan, and S. Venugopal, "Adaptive strategy to improve the efficiency of robot path planning using population based algorithm," *Int. J. Mech. Eng. Technol.*, vol. 8, no. 8, 2017.
- [6] W. Siming, Z. Tiantian, and L. Weijie, "Mobile Robot Path Planning Based on Improved Artificial Potential Field Method," *2018 IEEE Int. Conf. Intell. Robot. Control Eng.*, vol. 2, no. 1, pp. 29–33, 2018.
- [7] M. N. Zafar and J. C. Mohanta, "Methodology for Path Planning and Optimization of Mobile Robots :," *Procedia Comput. Sci.*, vol. 133, pp. 141–152, 2018, doi: 10.1016/j.procs.2018.07.018.
- [8] C. Chitra and P. Subbaraj, "Multiobjective optimization solution for shortest path routing problem," *World Acad. Sci. Eng. Technol.*, vol. 37, pp. 652–660, 2010.
- [9] C. Lamini, S. Benhlima, and A. Elbekri, "Genetic algorithm based approach for autonomous mobile robot path planning," in *Procedia Computer Science*, 2018, vol. 127, doi: 10.1016/j.procs.2018.01.113.
- [10] K. S. Suresh, G. Kesudh, S. Hemanath, and K. R. Sekar, "Quantified performance analysis of selection algorithms implemented in Mobile Robot path planning," *Int. J. Emerg. Technol.*, vol. 10, no. 1, 2019.
- [11] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Reliab. Eng. Syst. Saf.*, vol. 91, no. 9, pp. 992–1007, 2006, doi: 10.1016/j.ress.2005.11.018.
- [12] K. Deb, A. Sinha, P. J. Korhonen, and J. Wallenius, "An interactive evolutionary multiobjective optimization method based on progressively approximated value functions," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 723–739, 2010, doi: 10.1109/TEVC.2010.2064323.
- [13] G. Li and W. Chou, "Path planning for mobile robot using self-adaptive learning particle swarm optimization," *Sci. China Inf. Sci.*, vol. 61, no. 5, 2018, doi: 10.1007/s11432-016-9115-2.
- [14] A. Hidalgo-Paniagua, M. A. Vega-Rodríguez, and J. Ferruz, "Applying the MOVNS (multi-objective variable neighborhood search) algorithm to solve the path planning problem in mobile robotics," *Expert Syst. Appl.*, vol. 58, pp. 20–35, 2016, doi: 10.1016/j.eswa.2016.03.035.
- [15] H. Yang, J. Qi, Y. Miao, H. Sun, and J. Li, "A New Robot Navigation Algorithm Based on a Double-Layer Ant Algorithm and Trajectory Optimization," *IEEE Trans. Ind. Electron.*, vol. 66, no. 11, 2019, doi: 10.1109/TIE.2018.2886798.
- [16] U. Orozco-Rosas, O. Montiel, and R. Sepúlveda, "Mobile robot path planning using membrane evolutionary artificial potential field," *Appl. Soft Comput. J.*, vol. 77, 2019, doi: 10.1016/j.asoc.2019.01.036.
- [17] B. A and P. Panchu, "Multi-objective Optimization of Mobile Robot Path Planning Using Genetic Algorithm," *IOSR J. Mech. Civ. Eng.*, pp. 2278–1684, [Online]. Available: www.iosrjournals.org.
- [18] A. Hidalgo-Paniagua, M. A. Vega-Rodríguez, J. Ferruz, and N. Pavón, "MOSFLA-MRPP: Multi-Objective Shuffled Frog-Leaping Algorithm applied to Mobile Robot Path Planning," *Eng. Appl. Artif. Intell.*, vol. 44, pp. 123–136,

- 2015, doi: 10.1016/j.engappai.2015.05.011.
- [19] M. Elhoseny, A. Tharwat, and A. E. Hassanien, "Bezier Curve Based Path Planning in a Dynamic Field using Modified Genetic Algorithm," *J. Comput. Sci.*, vol. 25, 2018, doi: 10.1016/j.jocs.2017.08.004.
- [20] M. S. A. Mahmud, M. S. Z. Abidin, Z. Mohamed, M. K. I. A. Rahman, and M. Iida, "Multi-objective path planner for an agricultural mobile robot in a virtual greenhouse environment," *Comput. Electron. Agric.*, vol. 157, no. August 2018, pp. 488–499, 2019, doi: 10.1016/j.compag.2019.01.016.
- [21] S. Geetha, G. M. Chitra, and V. Jayalakshmi, "Multi objective mobile robot path planning based on hybrid algorithm," *ICECT 2011 - 2011 3rd Int. Conf. Electron. Comput. Technol.*, vol. 6, pp. 251–255, 2011, doi: 10.1109/ICECTECH.2011.5942092.
- [22] F. H. Ajeil, I. K. Ibraheem, M. A. Sahib, and A. J. Humaidi, "Multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm," *Appl. Soft Comput. J.*, vol. 89, 2020, doi: 10.1016/j.asoc.2020.106076.
- [23] M. Li, S. Yang, and X. Liu, "Pareto or Non-Pareto: Bi-criterion evolution in multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 645–665, 2016, doi: 10.1109/TEVC.2015.2504730.
- [24] "6 . Neural Network Based Path Planning Algorithm."
- [25] H. Y. Zhang, W. M. Lin, and A. X. Chen, "Path planning for the mobile robot: A review," *Symmetry (Basel)*, vol. 10, no. 10, 2018, doi: 10.3390/sym10100450.
- [26] B. Kovács, G. Szayer, F. Tajti, M. Burdelis, and P. Korondi, "A novel potential field method for path planning of mobile robots by adapting animal motion attributes," *Rob. Auton. Syst.*, vol. 82, pp. 24–34, 2016, doi: 10.1016/j.robot.2016.04.007.
- [27] C. W. Warren, "Multiple robot path coordination using artificial potential fields," pp. 500–505, 1990, doi: 10.1109/robot.1990.126028.
- [28] K. Deb, A. Member, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGAII," vol. 6, no. 2, pp. 182–197, 2002.
- [29] F. Ahmed and K. Deb, "Multi-objective optimal path planning using elitist non-dominated sorting genetic algorithms," *Soft Comput.*, vol. 17, no. 7, pp. 1283–1299, 2013, doi: 10.1007/s00500-012-0964-8.
- [30] H. Qu, K. Xing, and T. Alexander, "Neurocomputing An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots \$," *Neurocomputing*, vol. 120, pp. 509–517, 2013, doi: 10.1016/j.neucom.2013.04.020.
- [31] A. Lavin, "A Pareto Front-Based Multiobjective Path Planning Algorithm," 2015, [Online]. Available: <http://arxiv.org/abs/1505.05947>.
- [32] Y. Xue, "Mobile Robot Path planning with a non-dominated sorting genetic algorithm," *Appl. Sci.*, vol. 8, no. 11, 2018, doi: 10.3390/app8112253.