Robotica (2024), 1–25
doi:10.1017/S0263574724000377

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Multi-objective optimization approach for coverage path planning of mobile robot

Monex Sharma and Hari Kumar Voruganti

National Institute of Technology, Warangal, Telangana, India
Corresponding author: Hari Kumar Voruganti; Email: harikumar@nitw.ac.in

Received: 5 April 2023; **Revised:** 8 February 2024; **Accepted:** 17 February 2024

Keywords: coverage path planning; mobile robot; NSGA-iI; genetic algorithm; multi-objective optimization

Abstract

Coverage path planning (CPP) is a subfield of path planning problems in which free areas of a given domain must be visited by a robot at least once while avoiding obstacles. In some situations, the path may be optimized for one or more criteria such as total distance traveled, number of turns, and total area covered by the robot. Accordingly, the CPP problem has been formulated as a multi-objective optimization (MOO) problem, which turns out to be a challenging discrete optimization problem, hence conventional MOO algorithms like Non-dominated Sorting Genetic Algorithm-2 (NSGA-II) do not work as it is. This study implements a modified NSGA-II to solve the MOO problem of CPP for a mobile robot. In this paper, the proposed method adopted two objective functions: (1) the total distance traveled by the robot and (2) the number of turns taken by the robot. The two objective functions are used to calculate energy consumption. The proposed method is compared to the hybrid genetic algorithm (HGA) and the traditional genetic algorithm (TGA) in a rectilinear environment containing obstacles of various complex shapes. In addition, the results of the proposed algorithm are compared to those generated by HGA, TGA, oriented rectilinear decomposition, and spatial cell diffusion and family of spanning tree coverage in existing research papers. The results of all comparisons indicate that the proposed algorithm outperformed the existing algorithms by reducing energy consumption by 5 to 60%. This paper provides the facility to operate the robot in different modes.

1. Introduction

Path planning is a field of robotics that solves the optimal route for a robot to travel from one point to another, its types include shortest path planning for efficient point-to-point motion [1, 2] and coverage path planning (CPP) that determines the most efficient path to systematically cover an entire area, ensuring least cost value [3]. CPP becomes particularly crucial in scenarios where exhaustive exploration is required, such as in vacuum cleaning [4], lawn mowing [5], window glass cleaning [6], planetary exploration [7], and underwater exploration [8]. All of the aforementioned applications need to cover the entire accessible area while avoiding obstructions.

Regular cleaning in hospitals, offices, clinics, etc. is a crucial activity for reducing the risk to the safety of people. To reduce the workload of the cleaning team and the risk of direct contact with disinfectants, autonomous mobile robots would be the preferred option. Furthermore, a mobile robot can perform the cleaning task more precisely and cost-effectively than a human worker. In recent years, numerous researchers have proposed several methodologies in this domain. Survey of various methods of CPP is provided in ref. [9–11]. Based on cellular decomposition, CPP approaches can be categorized into two groups: (1) the exact cellular decomposition methods and (2) the grid-based methods. The exact cellular decomposition techniques break the entire environment into smaller sub-regions. Each sub-region is free from obstacles and is covered using simple algorithms such as zigzag, spiral, etc. Oksanen et al. [12] proposed a CPP algorithm based on the trapezoidal decomposition method for agricultural applications. A disadvantage of trapezoidal decomposition is that it produces numerous sub-regions that may not be

necessary. Choset et al. [13, 14] proposed boustrophedon cellular decomposition which is similar to the trapezoidal decomposition, with the exception that it only considers the vertices of an obstacle where a vertical segment can be drawn both above and below the vertex [15]. In cellular decomposition methods, there is some overlapping generated in the path, which cannot be omitted as there is no guarantee that the endpoint of a cell is adjacent to the starting point of the next cell. Changxin et al. [16] used an artificial potential field algorithm to solve a CPP problem and find a path that captures each environment cell. The method underperforms when covering the area near obstacles, and when the robot is trapped in a deadlock, it falls into the local optimum. Pham and Lam [17] presented a CPP problem based on a breadth-first search to determine the optimal path for a cleaning robot in a static environment. The method is simple to implement but struggles to find the optimal solution and consumes large memory in a large search space. Zengyu et al. [18] proposed an enhanced A^* algorithm for an autonomous mobile robot and studied the CPP algorithm. The method does not guarantee the optimal path and is difficult to implement in an environment with complex shapes of obstacles. E. Gonzalez et al. [19] published the backtracking spiral algorithm (BSA), in which the robot follows a wall to cover the previously unexplored region. Mitschke et al. [20] proposed the turn-away starting point (TASP) algorithm, which reduces the number of turns on a path by proceeding straight until an obstacle or previously visited cell is encountered. The paths obtained by methods [19, 20] are not optimal, and they can be improved using population-based methods such as a genetic algorithm (GA). Batsaikhan et al. [21] used the spatial cell diffusion (SCD) algorithm to cover the entire free region of a rectilinear workspace. José Prado et al. [22] published a paper in which the authors used oriented rectilinear decomposition (ORD) to traverse every accessible area. The algorithms in refs. [21, 22] employed the exact cell decomposition method to split the workspace into cells. Y Gabriely and E Rimon [23] presented a work based on spanning tree coverage (STC) approaches to solve the CPP of a mobile robot. The method enables the computation of a spanning tree graph and the robot visits each cell once in a given free space by following the spanning tree graph. K R Guruprasad [24] reported an algorithm that is an extension of the spanning tree coverage (STC) algorithm for mobile robot coverage. The goal of the algorithm is to maximize the coverage area specifically to cover the sub-cells that are partially occupied by obstacles. K. R. Guruprasad and T. D. Ranjitha [25] used cell permeability-based coverage algorithm to solve a CPP problem of a mobile robot. The performance of the algorithms in refs. [24, 25] is compared to spiral and full STC-based algorithms, and a better coverage area is achieved with lower overlap. VG Nair and KR Guruprasad [26] studied the CPP of multi-robots based on the hybrid Voronoi partition method. The authors combined geodesic-distance-based Voronoi partition and Manhattan-distance-based Voronoi partition to solve the optimum path with non-overlapping at the grid level. The CPP of a mobile robot based on a GA has been studied in refs. [27–29]. Schafle et al. [30] reported a CPP problem for grid-based environmental representation based on a hybrid genetic algorithm (HGA). They utilized TASP and BSA to conduct a local search if a path generated by crossover or mutation did not provide a feasible solution. Sharma and Voruganti [31] published a work on the CPP of a mobile robot using different approaches of preference-based MOO methods. The authors adopted two objective functions which are minimization of energy consumption and maximization of coverage area and conducted case studies in two different environments. Shang et al. [32] solved a multi-objective CPP problem using an improved GA. The objective functions of the problem are the length of the driving path, the elevation height, and the number of turns. Using predefined weighting factors, the authors converted the multi-objective problem into a single-objective problem and obtained an optimal solution. Sharma and Voruganti [33] published a similar work based on GA where the authors converted the MOO problem into an SOO problem using the weighted sum method. The results show that the completion time of the task is less than that of the constraint optimization technique. Almost all the methods except a few have focussed on finding a path to cover the area. There can be more than one path for a given environment that covers the entire area. The question of the optimal path arises in such cases. The interest of the present work is to find such an optimal path to multiple-optimal criteria (like energy consumption, length of the path, etc). This led to posing the problem as a MOO problem and finally providing a set of solutions instead of one feasible or optimal solution.

https://doi.org/10.1017/S0263574724000377 Published online by Cambridge University Press

Work floors like offices and clinics are more likely static environments and it is required to generate multiple path plans according to user preference, which may be less energy consumption and effective coverage of the area. This paper presents a modified Non-dominated Sorting Genetic Algorithm-2 (NSGA-II) for solving the CPP problem and generating Pareto-optimal solutions for floor-cleaning by a mobile robot in a known environment with convex and non-convex-shaped static obstacles. The environment is expected as input for the proposed CPP method, which can be obtained using a camera or sensors. The environment is discretized into cells using a grid-based method and the size of the cell depends on the dimensions of a circular robot. The problem in this research belongs to the field of discrete optimization. In this research, every feasible solution obtained from each iteration satisfies three conditions: (1) the robot must cover all the free cells (100% coverage), (2) the robot avoids moving along the cell which is fully or partially occupied by any obstacle, and (3) the robot cannot skip a neighboring free cell while moving along a direction. The total distance traveled and the total number of turns taken by the robot to complete the task are employed as the two objective functions in this paper. Using an energy equation that will be explained in Section 3.6, the paths obtained by the proposed method are used to calculate the energy consumed by the robot. Among the Pareto-optimal solutions produced by the proposed method, an optimal solution is selected based on the minimum energy consumption (higher-level information). The optimum solution is compared to the solution presented in previous publications. The proposed algorithm's optimal solution outperformed the existing results. The remaining sections are organized as follows: In Section 2, environment modeling and problem formulation are presented. The proposed method is presented in Section 3. In Section 4, results and discussion are presented. The results and future scope are discussed in Section 5.

2. Environment modeling and problem formulation

2.1. Environment modeling

To facilitate the movement of the mobile robot, workspace W is discretized into cells of a square grid. The area of each square cell in the working environment is equivalent to the area covered by the mobile robot. When a robot passes through a cell, it is considered that the cell is completely occupied by the robot. Each cell, C, is either completely occupied by an obstacle, C_{obs}, or completely free of an obstacle, C_{free}. The free cells and obstacle cells are represented by white and black colors in the grid map, respectively, as shown in Fig. 1. The robot is permitted to move in C_{free}. C_{free} cells that the robot has visited are known as visited cells C_{vis}, whereas C_{free} cells that the robot has not visited are known as unvisited cells C_{unv}. The starting point of the robot for a given environment is signified by P_{start}, which is denoted by green color as given in Fig. 1. Therefore, the initial and final positions are identical. Each cell is labeled with an integer value, beginning with the left bottom corner cell, which is assigned as integer "1" as shown with green color in Fig. 1(a). These integers are termed as decision variables for the problem in this work. The mobile robot considered in this study can only move in four directions. Fig. 1(b) illustrates the direction along which the robot can move by solid black lines (forward, backward, leftward, and rightward).

2.2. Problem formulation

The problem is to find a path, p, which is a set of connected sequence cells (p_{i}) from the start to the goal position. When $C_{\text{unv}} = \Phi$, where Φ is an empty set, the robot has finished covering the environment, and it moves back to the P_{start}. The proposed algorithm creates a set of feasible solutions or paths in each iteration. The fitness evaluation is carried out after a path has been generated to determine the quality of the path. The two objectives of the proposed algorithm are (1) the total distance traveled by the robot and (2) the number of turns it makes (left, right, and U-turns). For each iteration of the algorithm, these two objectives are minimized in order to lower the amount of energy consumed by the robot.

https://doi.org/10.1017/S0263574724000377 Published online by Cambridge University Press

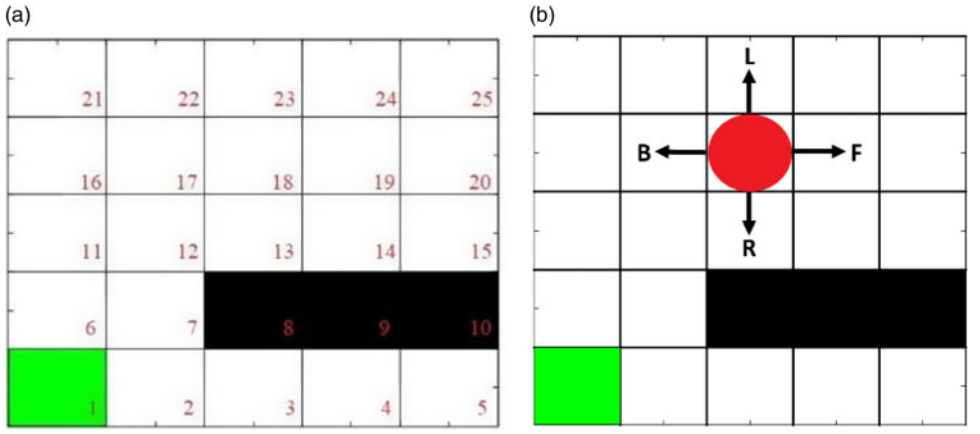

Figure 1. *(a) An environment with decision variables, (b) Possible four directions of motion of a mobile robot.*

3. Proposed method

The GA is an evolution-based algorithm that operates on the survival of the fittest principle. Deb et al. [34] developed the NSGA-II algorithm, which is an extension of GA to solve the MOO problem. A flowchart of the proposed method (modified NSGA-II) is illustrated in Fig. 2. This paper applies the modified NSGA-II algorithm to solve the CPP problem of a mobile robot. Moreover, a discrete optimization approach is adopted as the robot can only move to neighboring cells which are discretized using a grid-based method. The following notations are used in this paper to represent the proposed method. A chromosome is a single feasible solution that is a combination of the decision variables, and each bit of the chromosome is known as a gene or a decision variable. The set of solutions generated by each iteration is referred to as the population. Previous solutions that produce new solutions are termed as parents, and the new solutions are regarded as offspring. Unlike a continuous optimization problem, regular crossover and regular mutation do not guarantee to provide a feasible solution in the case of a discrete optimization problem. Hence, the crossover and mutation are modified by using TASP and A* algorithms. In the proposed method, the next generation of the population is obtained from the previous generation by following five steps, namely tournament selection, modified crossover, modified mutation, non-dominated sorting, and crowding distance sorting. It is considered that the solutions of the most recent generation represent the Pareto-optimal solutions.

3.1. Chromosome design

A gene in a chromosome signifies a cell traversed by a mobile robot in the environment depicted in Fig. 3. The solid black line in Fig. 3(a) represents the path traced by the robot in its environment. Fig. 3(b) demonstrates the arrangement of genes in the chromosome according to the path. The size of chromosomes is proportional to the distance traveled by the robot. Hence, the number of genes present on a chromosome may be the same or different from that of another chromosome.

3.2. Initialization

The initial paths belong to the first parents of the proposed method. The initial paths must possess two properties: (1) a wide variety of initial paths, and (2) the initial paths must already have good fitness values. For the first property to be satisfied, a large number of random paths are generated. In addition to random paths, four distinct methods (Spiral, Zigzag, TASP, and BSA) [13, 14, 19, 20] are employed to generate initial paths that satisfy the second property.

https://doi.org/10.1017/S0263574724000377 Published online by Cambridge University Press

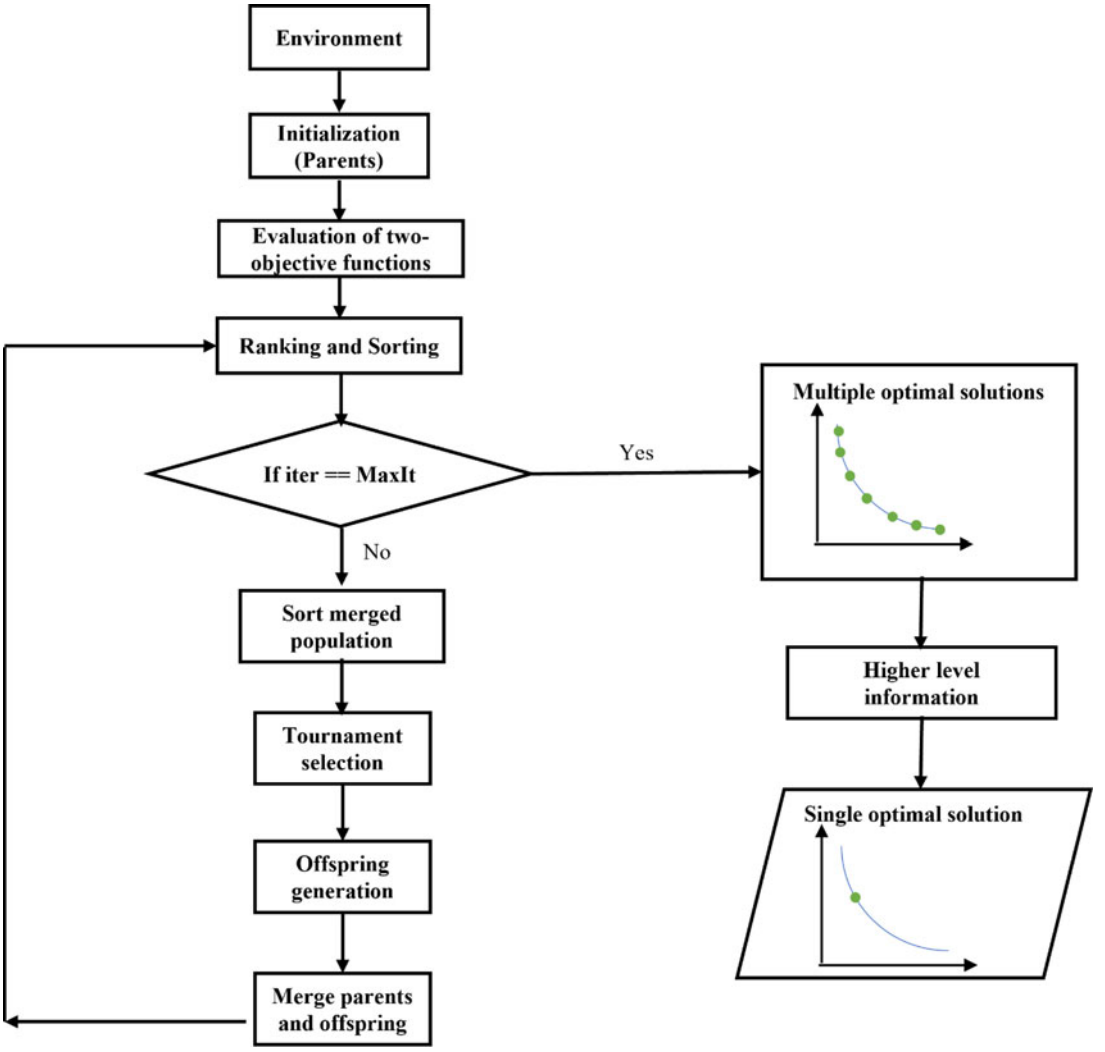

Figure 2. *Flowchart of the modified Non-dominated Sorting Genetic Algorithm-2 method.*

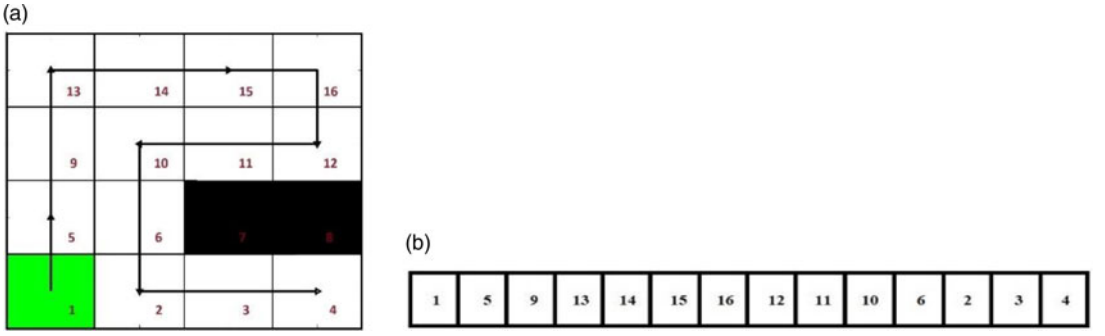

Figure 3 *(a). The path covered by a robot. (b). Chromosome design of the path.*

https://doi.org/10.1017/S0263574724000377 Published online by Cambridge University Press

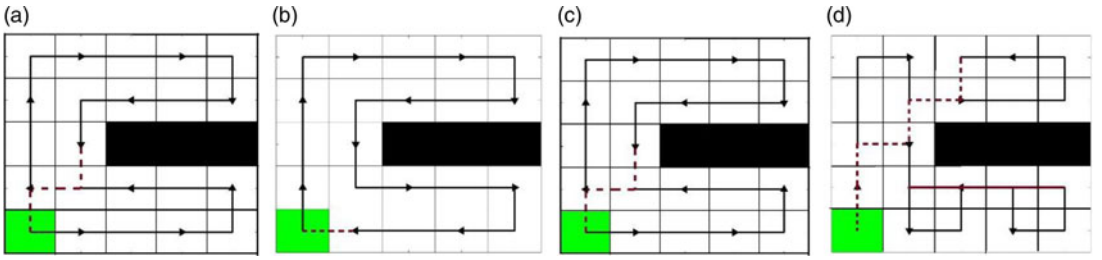

Figure 4. *Path generated by (a) turn-away starting point algorithm method, (b) backtracking spiral algorithm method, (c) Spiral method, (d) Zigzag method.*

Fig. 4 shows the initial paths generated by the four different approaches in an environment. In the figure, the black solid line and the red solid line represent the non-repeated path and the repeated path, respectively, covered by a mobile robot. After the robot has traversed every accessible cell, it moves back to the starting point. The black dotted line and red dotted line indicate the non-repeated and repeated paths that traversed back to the starting point. When the robot is surrounded by visited neighboring cells and obstacles, the A* algorithm is used to traverse the robot to the nearest unvisited cell [35]. When the robot has completed the CPP, the A* algorithm is used to return it to the starting point.

3.3. Fitness functions

The quality of the path is determined by the fitness functions of the proposed method. This paper employs two objective functions: (1) the distance traveled by the mobile robot, f_1, and (2) the number of turns by the robot, f_2. The mathematical formulation of the two objective functions is explained below.

3.3.1. The total distance traveled by the robot

The Manhattan distance method is employed to determine the total distance, f_1, covered by the robot as shown in equation (1). It is the sum of the absolute differences between the corresponding coordinates of two points in a grid.

$$f_1 = \sum_{i=1}^{N_{gc}-1} (|x_{i+1} - x_i| + |y_{i+1} - y_i|) \tag{1}$$

where N_{gc} is the number of waypoints or positions visited by the robot. The position of the robot at i^{th} waypoint is represented by $p_i(x_i,y_i)$.

3.3.2. Counting total turns and total U-turns

The number of turns (T) and *U*-turns (U) made by a robot during coverage path planning are calculated using the current, previous, and next positions of the robot, i.e. p_i, $p_{i\text{-}1}$, and p_{i+1}. There are only two types of turns possible i.e. either ±90 or 180 degrees turn because the robot considered in the study can move only in four directions (Fig. 1(b)).

The energy required for 180-degree turning is higher compared to 90-degree turning. Hence, the effective number of turns, f_2, is calculated by,

$$\mathrm{f_2} = \mathrm{T} + \mathrm{w} \times \mathrm{U} \tag{2}$$

where is $w = 1.5$ with an approximation that a U-turn takes 50% more energy than a 90-degree turn.

https://doi.org/10.1017/S0263574724000377 Published online by Cambridge University Press

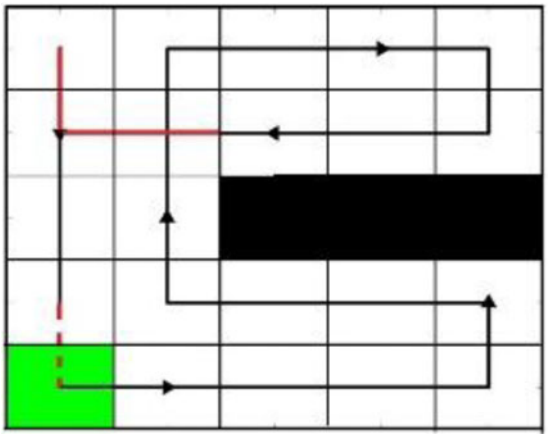

Figure 5. *Path produced by a random approach.*

The problem is to minimize two objective functions f_1 and f_2. The constraint of the problem is 100% coverage of an entire free area. The MOO problem is stated as,

$$\begin{aligned} &\text{Minimize } \mathrm{f} = [\mathrm{f}_1 \quad \mathrm{f}_2] \\ &\text{Subject to:} \\ &\mathrm{p_i} \text{ in } \mathrm{C_{vis}} \in \mathrm{C_{free}}, \\ &\mathrm{C_{unv}} = \Phi. \end{aligned} \tag{3}$$

For example, the path shown in Fig. 5, $f_1 = 24$ and $f_2 = 7 + 1.5 \times 1 = 8.5$. Hence, the values of the objective functions for the MOO problem are written as $\mathrm{f} = [24, 8.5]$.

3.4. Operators of NSGA-II

Reproduction, mutation, and the survival of the fittest are the operators of natural evolution. In the proposed method, crossover, mutation, selection, non-dominated sorting, and crowding distance are equivalent to the natural operators. The operators are discussed in detail in the following sections.

3.4.1. Selection

Selection is the process of choosing parents to produce offspring for the next generation through other GA operators. Selecting only the best chromosomes in each generation may result in premature convergence. Therefore, it is essential to select parents with the most advantageous as well as diverse chromosomes.

This method employs tournament selection to achieve the desired chromosome diversity and convergence rate. In this method, a number of parents, (h), are selected at random from the entire population of chromosomes. The fittest parent is selected and added to a mating pool. The procedure of selecting parents and determining the optimal chromosome is repeated until the size of the mating pool matches that of the initial population. This method provides a chance for weaker chromosomes to be selected from the parent population and appends to the mating pool, as the set of parents is always different and is chosen at random.

3.4.2. Modified crossover for discrete optimization

The crossover operator works similarly to reproduction in nature, with parents mating to generate offspring. In this paper, a modified two-point crossover is employed to generate two offspring from two

https://doi.org/10.1017/S0263574724000377 Published online by Cambridge University Press

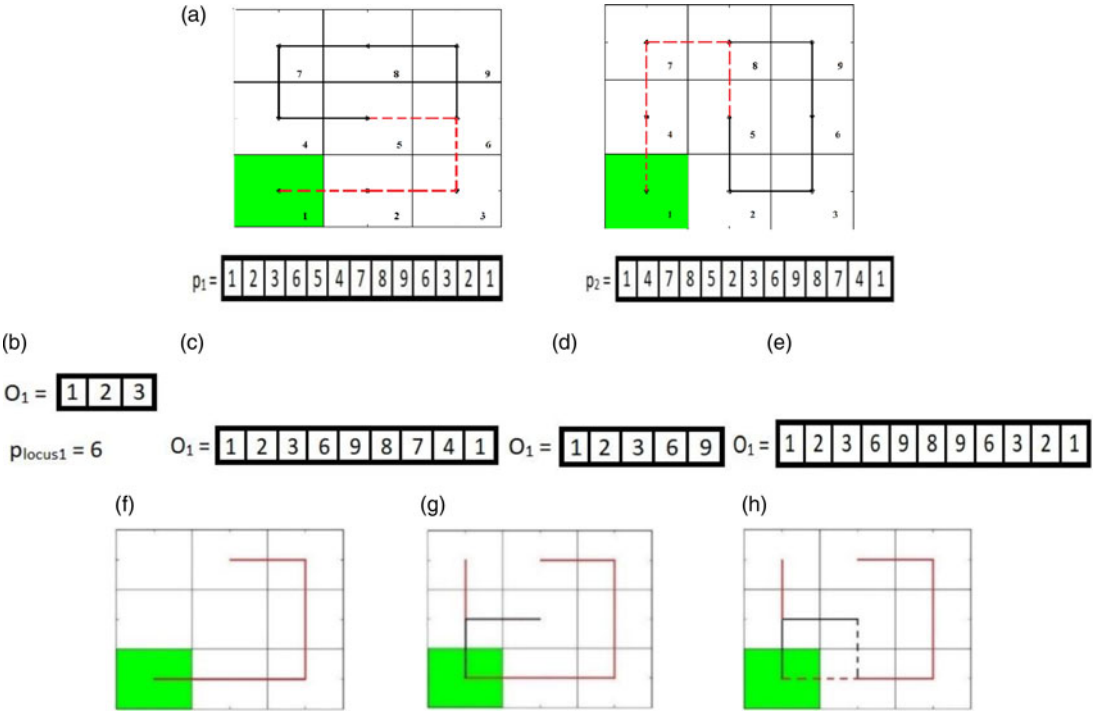

Figure 6. *Example of modified crossover.*

parents. Because, a regular two-point crossover of traditional genetic algorithm, TGA used in case of a continuous problem might not generate an offspring that consists of genes that are neighboring to each other and does not provide a feasible path due to the robot kinematics.

Fig. 6 depicts an example of a 3-by-3 simulation environment without any obstacles. Two parents P_1 and P_2 are chosen at random, and their paths and chromosomes are depicted in Fig. 6(a). Two random cutting points i and j where $i < j$ are selected, for example, $i = 4$ and $j = 6$. A locus point p_{locus1} is determined as the i^{th} element of p_1 and a vector O_1 is generated containing the elements from the first index to the $(i - 1)^{th}$ index of p_1, as shown in Fig. 6(b). The first occurrence of p_{locus1} is extracted from the second parent p_2 and the index k is assigned to it. As shown in Fig. 6(c), the vector O_1 is modified by appending the elements from the k^{th} index to the last element of P_2. The number of elements $(n = 9)$ in vector O_1 is determined and j is replaced with n if $n < j$; otherwise, the value of j is kept unchanged. Another locus point p_{locus2} is determined as the j^{th} element of O_1. The elements from index j to the last element of vector O_1 are deleted as shown in Fig. 6(d). The first occurrence of p_{locus2} from p_1 is extracted and its index is labeled as r. As shown in Fig. 6(e), the elements of p_1 are appended from the r^{th} index to the last element to the vector O_1. The path generated by the genetic offspring O_1 is depicted in Fig. 6(f), but the solution is not feasible.

Since such offspring members may not be feasible (i.e., coverage is not 100%), they are further appended by using TASP and A* algorithms. This entire crossover process is different from continuous optimization in the sense that here chromosome is a set of number of cells representing the path. The TASP and A* algorithms are employed to modify and transform it into a feasible solution O_1'. The TASP algorithm enables the robot to traverse all free cells, as shown in Fig. 6(g). A star algorithm is applied to return the robot to its charging station after it has traversed all accessible cells, as depicted in Fig. 6(h). The same procedure is applied to develop the second offspring O_2'. Table I depicts two parents and offspring produced by crossover and further modified with TASP and A* algorithms.

https://doi.org/10.1017/S0263574724000377 Published online by Cambridge University Press

Table I. *Example of a regular and improved crossover operator.*

Chromosome	Path
Parent P_1	1,2,3,6,5,4,7,8,9,6,3,2,1
Parent P_2	1,4,7,8,5,2,3,6,9,8,7,4,1
Offspring O_1	1,2,3,6,**9,8**,9,6,3,2,1
Offspring O_2	1,4,7,8,**9,6**,9,8,7,4,1
Modified offspring O_1'	1,2,3,6,**9,8**,9,6,3,2,1,4,7,4,5,2,1
Modified offspring O_2'	1,4,7,8,**9,6**,9,8,7,4,1,2,3,2,5,2,1

Table II. *Example of the mutation operator.*

Chromosome	Path
Parent	1, 2, 3, 6, 5, 4, 7, 8, 9, 6, 3, 2, 1
Offspring	1, 2, 3, 6, **9**, 8, 7, 4, 5, 2, 1

3.4.3. Modified mutation for discrete optimization

A probability factor called p_m determines whether a mutation can be used. A mutation is a process in which changes are made to the chromosome to create a new solution to make the population more diverse. Since the problem is based on discrete optimization, a regular mutation of GA, which randomly replaces a gene, does not provide a feasible path. In this paper, a modified mutation is implemented to generate a feasible offspring from a parent, which includes a regular mutation followed by an improvement using the TASP and A* algorithms. A gene with an index of m from a parent chromosome is chosen at random. The selected gene is replaced by another gene which is neighboring to the gene with index m-1. All the genes from the chromosome following the index m are removed. The TASP technique is applied to allow the robot to pass through the remaining unvisited cells, and the A* algorithm is applied to return the robot to the charging position.

Table II illustrates an example where $m = 5$. Gene (5) with index m is replaced at random by another neighboring gene (9) of gene (6) with index m-1 as shown in bold. The TASP algorithm is applied to generate the path to cover the remaining unvisited cells, which are represented by the underlined genes in Table II, to complete the CPP process. The A* algorithm is applied to move the robot to the charging position or gene (1), which is indicated in italics.

3.4.4. Merging

In this paper, the number of individuals in each generation, N_{par}, is selected as 150. Where the number of offspring generated through crossover, O_{nc}, and mutation, O_{nm}, depends on the p_c and p_m, respectively. The entire population of parents, N_{par}, and offspring through crossover, $O_{nc,}$ and mutation, O_{nm}, are combined to form a new population. This newly formed population is known as the merged population.

3.4.5. Sorting operators of NSGA-II

Non-dominated sorting (NS) is a key component of the proposed method that is used to find non-dominated solutions in merged populations based on the dominance concept and to rank them accordingly. The best solutions are represented in the first-rank fronts and they are non-dominated by any feasible solution. Likewise, the second-rank fronts represent the second-best solutions that are only dominated by the solutions in the first-rank fronts. The crowding distance sorting (CDS) method of the proposed algorithm is used to maintain a diverse set of solutions in the population and enable the algorithm to efficiently explore the multi-objective search space. The concepts of NS and CDS were explained with examples in ref. [34].

https://doi.org/10.1017/S0263574724000377 Published online by Cambridge University Press

Table III. *Motion types and energy.*

Motion	Energy
Straight	$2J_s$
Turn	$J_{dec} + J_T + J_{acc}$
U-turn	$J_{dec} + J_{UT} + J_{acc}$
Starting	$J_{acc} + J_s$
Finishing	$J_{dec} + J_s$

Table IV. *Energy parameters and value.*

Energy Parameters	Energy Value (J)
J_s	7.83
J_{dec}	5.78
J_{acc}	17.09
J_T	38.57
J_{UT}	51.22

3.5. Termination

The termination condition of the proposed algorithm is either a predefined fixed number of generations or the existence of desired solutions. In this study, the termination condition is determined by a predefined number of generations. However, if all the solutions of three consecutive generations remain unchanged before the maximum generation, the process is terminated.

3.6. Calculation of energy

The value of the objective functions: total distance traveled by the robot, f_1 and the total number of turns by the robot, f_2, which are obtained from the optimal solutions of the proposed algorithm, are used to determine the energy consumption by the robot. The total energy consumption is calculated using equation (4) and data obtained from [30] for a differential drive robot. Table III shows energy consumption for each type of motion. The energy parameters and energy values in joules are provided in Table IV [30].

$$J = (J_{acc} + J_s) + 2J_s n_s + (J_{dec} + J_t + J_{acc})\, n_t + (J_{dec} + J_{ut} + J_{acc})\, n_{ut} + (J_{dec} + J_t) \tag{4}$$

In equation (4), n_t and n_{ut} represent the number of turns (right and left) and number of U-turns taken by the robot respectively whereas $n_s = N_{gc} - 1 - n_t - n_{ut}$, is the number of straight motions traversed by the mobile robot. J_s represents half the energy consumption by the robot moving along a straight motion. J_{acc} and J_{dec} signify the energy required to accelerate and decelerate by the robot respectively. J_t and J_{ut} denote the energy required for (right and left) turn and U-turn, respectively.

4. Results

The environments considered for the case studies are rectangular and discretized into square cells. The parameters of the proposed method are listed in Table V. The robot is used to clean the entire accessible free area in four complex environments with a range of obstacle shapes. The proposed algorithm is then compared with the results of other CPP algorithms developed in previously published papers [21, 22, 30, 36].

4.1. Case studies in four complex environments

This section presents the comparisons of the CPP problem using the proposed method and the solutions developed by HGA [30] and traditional GA (TGA) [36] in four rectangular simulation environments.

https://doi.org/10.1017/S0263574724000377 Published online by Cambridge University Press

Table V. *Algorithm parameters and its value.*

Modified NSGA-II Parameters	**Value**
Mutation probability	0.4
Crossover probability	0.8
Number of generations	30
Population size	150

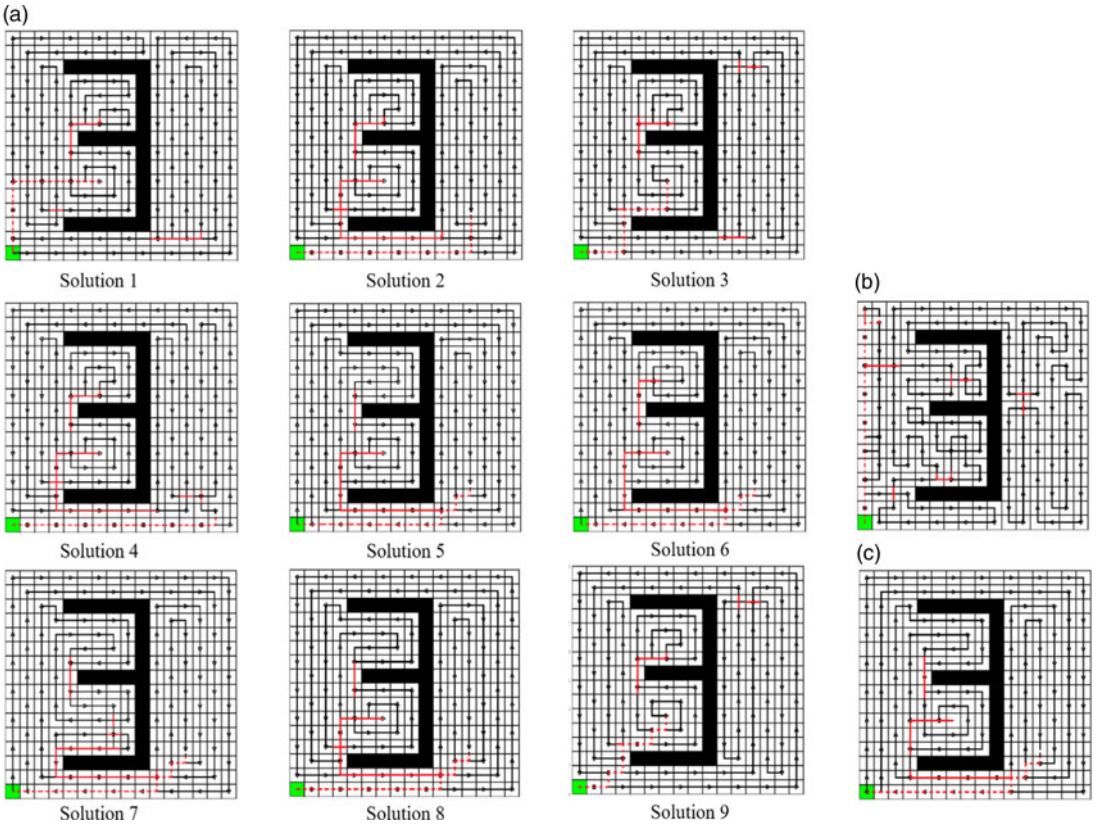

Figure 7. *Solutions generated with inverted E-shaped obstacle using (a) the proposed approach, (b) traditional genetic algorithm, and (c) hybrid genetic algorithm.*

Fig. 7 depicts the path developed for the mobile robot through an inverted *E*-shaped obstacle in a rectilinear environment, which is discretized into 256 cells.

The total number of turns, U-turns, distance traveled, and energy consumed by the mobile robot of the nine best solutions among the Pareto-optimal solutions obtained by the proposed method of the environment are shown in Table VI. Using higher-level information, an optimal solution with the lowest energy consumption (solution 1) is chosen from the Pareto-optimal set generated by the proposed method. Figs. 8 and 9 show *S*-shaped and *G*-shaped obstacles in the same setting as the inverted *E*-shaped obstacles. Fig. 10 depicts the results in a complex environment with 400 square cells.

The objective functions of the solutions generated by the proposed method of all shapes of obstacles are represented in the objective space as shown in Fig. 11. The solutions obtained by HGA and TGA are also represented in Fig. 11 with red and green colors respectively. Based on the comparison, it can be concluded that at least one solution derived from the proposed method generated less energy than HGA and TGA. Table VII shows that the proposed method outperformed HGA and TGA in the four

https://doi.org/10.1017/S0263574724000377 Published online by Cambridge University Press

Table VI. *Energy consumption of the mobile robot of the nine best solutions developed by the proposed method.*

Solution	Number of turns	Number of U-turns	Distance traveled (cells)	Energy consumption (J)	Computation Time (s)
1	42	0	250	5766.7	820
2	41	2	252	6119.6	845
3	41	1	266	5842	830
4	42	1	266	5919.1	834
5	41	2	262	6182.3	860
6	41	2	264	5837.8	853
7	43	0	262	5875.1	843
8	44	1	262	6198.5	837
9	46	0	248	6294.3	847

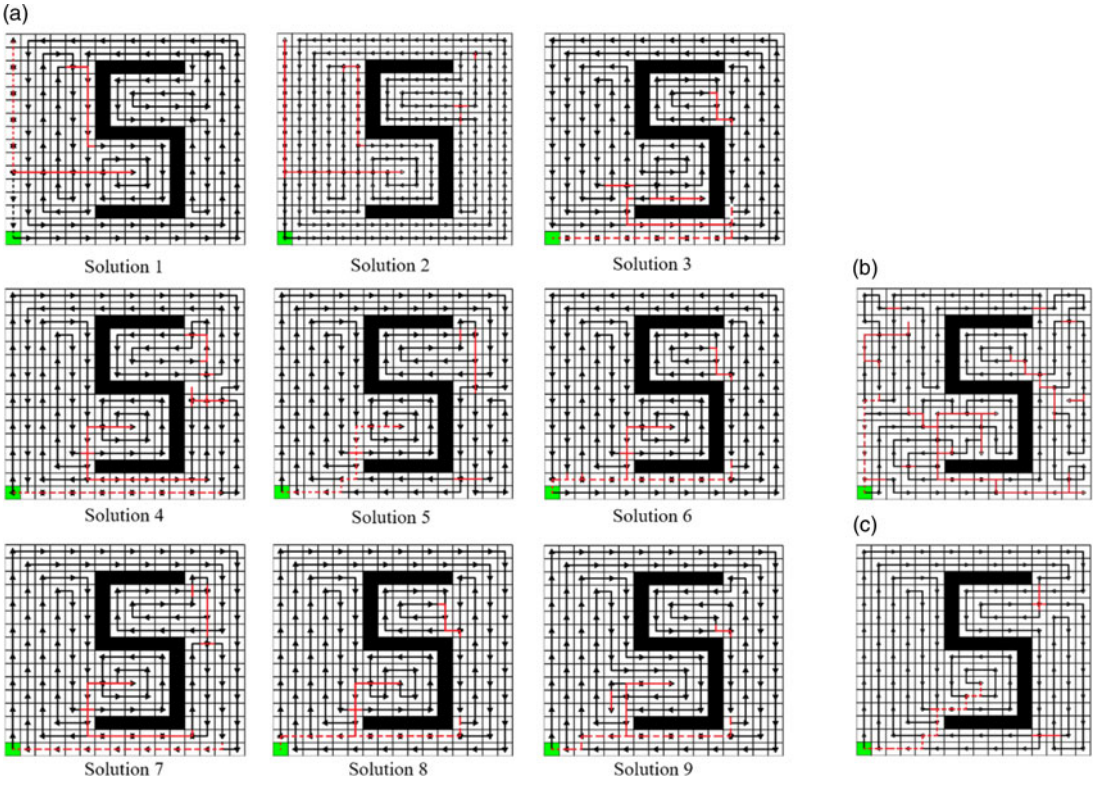

Figure 8. *Solutions generated with S-shaped obstacle, (a) using the proposed approach, (b) traditional genetic algorithm, and (c) hybrid genetic algorithm.*

environments. The energy consumption by the proposed method is less than HGA by 6%, 3%, 2.8%, and 6% while the proposed method generates less energy than TGA by 32%, 37%, 34.3%, and 27.2% in the given environments. The computation time of the proposed method is 1.5–2 times more than that of the TGA and HGA methods. Even though the suggested method requires more computation time than the existing methods, this is not a significant drawback because the study's application is to generate multiple sets of solutions in a static and known environment.

https://doi.org/10.1017/S0263574724000377 Published online by Cambridge University Press

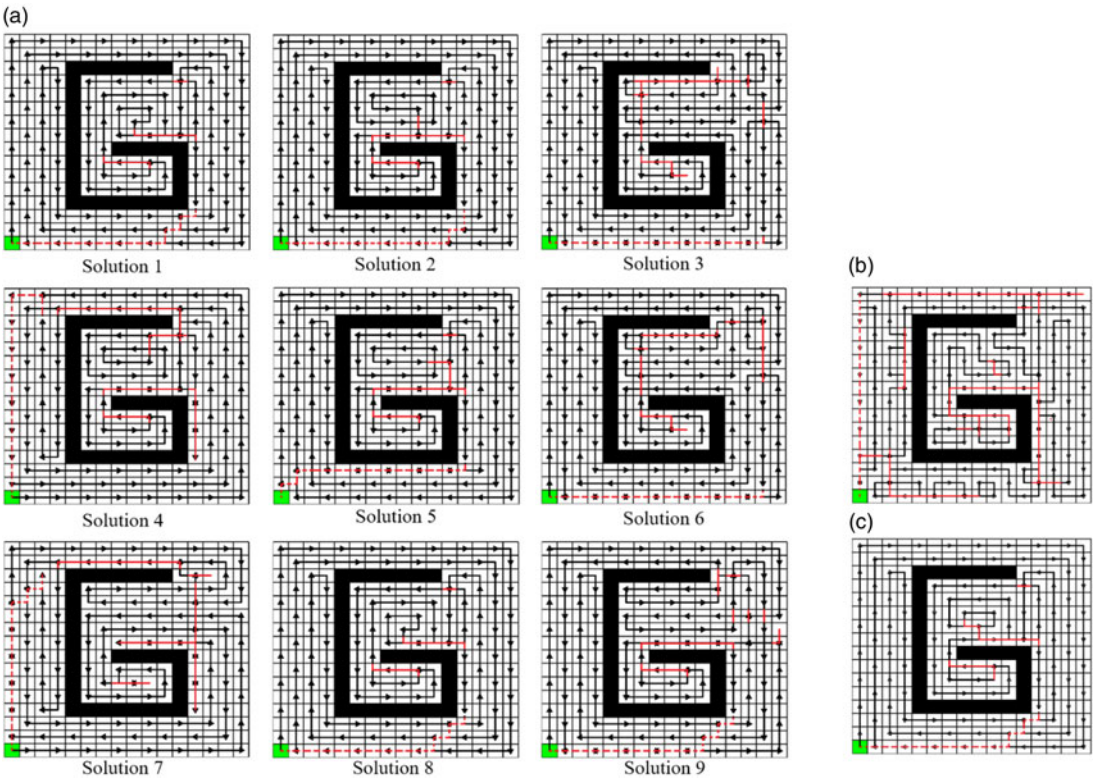

Figure 9. *Solutions generated with G-shaped obstacle, (a) using the proposed approach, (b) traditional genetic algorithm, and (c) hybrid genetic algorithm.*

4.2. Comparison of the proposed method with HGA for the same environment as [30]

Schafle et al. [30] published a study in which the HGA algorithm was used to solve the CPP problem with energy consumption as the objective. Fig. 12 depicts a comparison between the results of the proposed method and their paper. The nine best solutions among the Pareto set are shown in Fig. 12(a) which is developed by the proposed method. Fig. 12(b) represents the path generated by the HGA approach. Fig. 13 depicts the nine best solutions of the proposed method in objective space for clarity. The solution denoted by a red star is identical to the solution produced by the HGA approach. As shown in Table VIII, the energy consumed by solutions 1, 2, 4, and 5 is less than the energy consumed by the solution generated by HGA. It is observed from Table VIII that the number of turns developed by solutions 1, 2, and 4 is less than that of the HGA approach which decreases the energy consumption by 6.23%.

4.3. Comparison of the proposed method with TGA for the same environment as [36]

Sumit Gajjar et. al. [36] proposed work on the CPP problem of the mobile robot that traversed the accessible working environment area with various types of obstacles using TGA. Fig. 14 displays a comparison between the paths generated by the proposed approach and TGA in a working environment with square shape of obstacles. The comparison of the proposed approach and TGA with circular and irregular shapes of obstacles in the same environment is represented in Fig. 15 and Fig. 16, respectively. An optimal solution with the lowest energy consumption is selected among Pareto set, which is generated by the proposed algorithm using higher-level information. As shown in Fig. 17, the objective functions of the solutions generated by the proposed algorithm for all types of obstacles are represented in the objective space. The solution produced by TGA is denoted in green color in Fig. 17. The energy consumed by the mobile robot using the TGA and the proposed method to complete the CPP task in a

https://doi.org/10.1017/S0263574724000377 Published online by Cambridge University Press

Table VII. *Comparison of energy consumption developed by the proposed method, traditional genetic algorithm (TGA), and hybrid genetic algorithm (HGA).*

Environment	Method	Number of turns	Number of U-turns	Distance traveled (cells)	Energy consumption (J)	Computation Time (s)
E-shape obstacle	TGA	88	2	278	8537.5	330
	HGA	41	2	262	6135.3	460
	Proposed method	42	0	250	5766.7	820
S-shape obstacle	TGA	97	5	288	9281.4	376
	HGA	45	0	246	5951	520
	Proposed method	37	2	274	5779.9	874
G-shape obstacle	TGA	88	1	296	8761	324
	HGA	43	0	250	5922.1	452
	Proposed method	41	0	248	5752.2	814
Complex shape obstacle	TGA	123	7	390	12,186	460
	HGA	67	2	396	9424	543
	Proposed method	61	4	396	8874.7	932

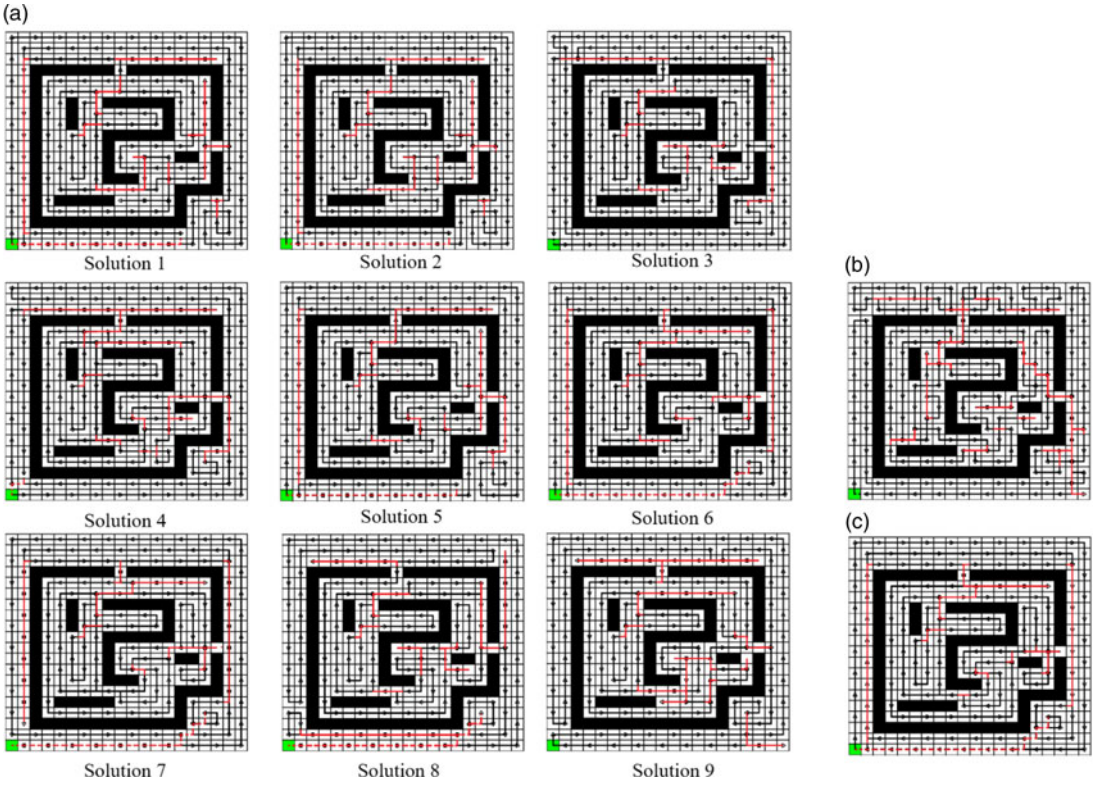

Figure 10. *Solutions generated in complex-shaped obstacle, (a) using the proposed approach, (b) traditional genetic algorithm, and (c) hybrid genetic algorithm.*

https://doi.org/10.1017/S0263574724000377 Published online by Cambridge University Press

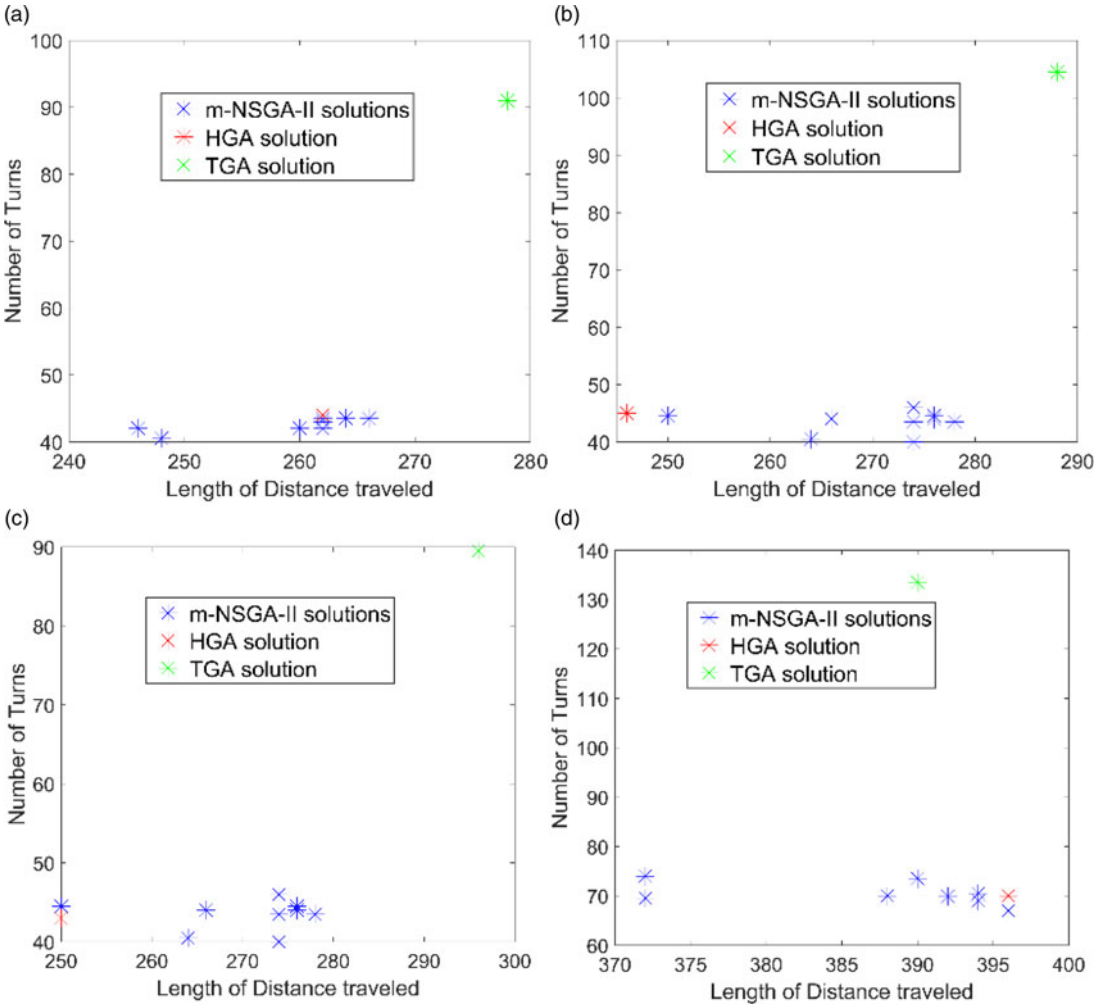

Figure 11. *Solutions generated by the proposed method, hybrid genetic algorithm and traditional genetic algorithm in (a) E-shaped, (b) S-shaped, (c) G-shaped, and (d) complex-shaped obstacles.*

rectilinear environment with various shapes of obstacles are shown in Table IX. The number of turns of the optimal solution generated by the proposed method is much less than the TGA approach in all types of obstacles, hence the energy obtained is reduced by 47.2%, 43.8%, and 43.8% in environments with square, circular, and irregular shapes of obstacle, respectively.

4.4. Comparison of the proposed method with SCD and ORD methods for the same environment given in [21, 22]

Prado et al. [22] presented work on the CPP problem using the SCD method on a mobile robot that covered an entire accessible environment. Batsaikhan et al. [21] proposed the same problem using ORD. Fig. 18 illustrates a comparison between the paths generated by the proposed method and those generated by SCD and ORD. Using higher-level information based on minimum energy consumption, solution 1 from the optimal solutions generated by the proposed algorithm is selected.

Fig. 19 illustrates the objective functions of the solutions generated by the proposed algorithm in the objective space. In Fig. 19, the solutions obtained by SCD and ORD are depicted in red and green colors, respectively. On the basis of the comparison, it can be concluded that at least one solution generated by the proposed method is less energy than SCD and ORD.

https://doi.org/10.1017/S0263574724000377 Published online by Cambridge University Press

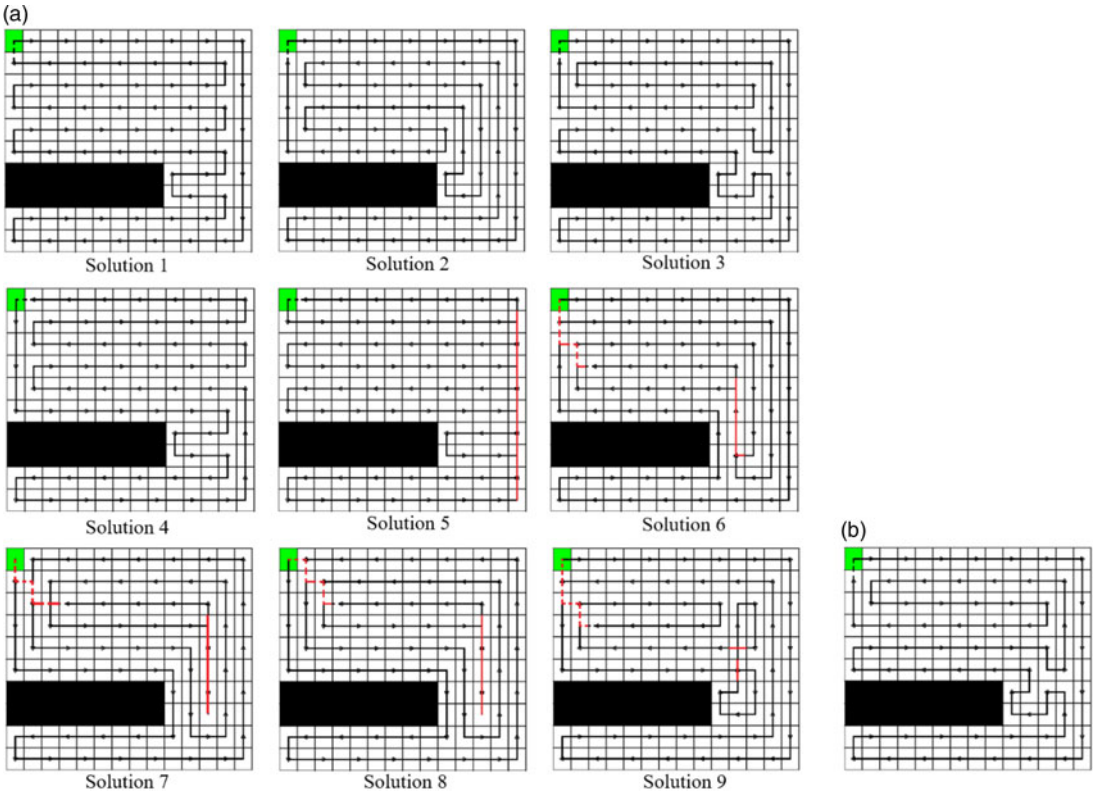

Figure 12. *Solutions generated by (a) the proposed algorithm, and (b) hybrid genetic algorithm in [30].*

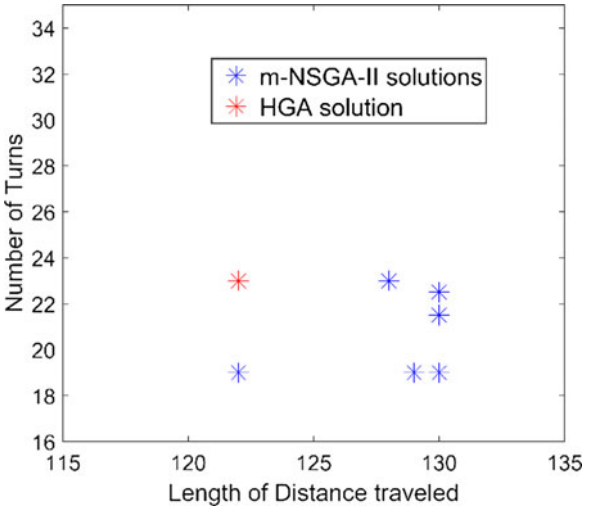

Figure 13. *Solutions with blue and red marks are generated by modified Non-dominated Sorting Genetic Algorithm-2 and hybrid genetic algorithm, respectively.*

Table X depicts a comparison of the energy consumption of the mobile robot to complete the CPP task generated by the proposed algorithm with SCD and ORD. Although the distance traveled generated by the proposed method is more than SCD and ORD, the number of turns developed by the proposed method is less than SCD and ORD by 65% and 60.4%. As a result, the energy consumption by the proposed method is decreased by 41% and 35.3%, respectively.

https://doi.org/10.1017/S0263574724000377 Published online by Cambridge University Press

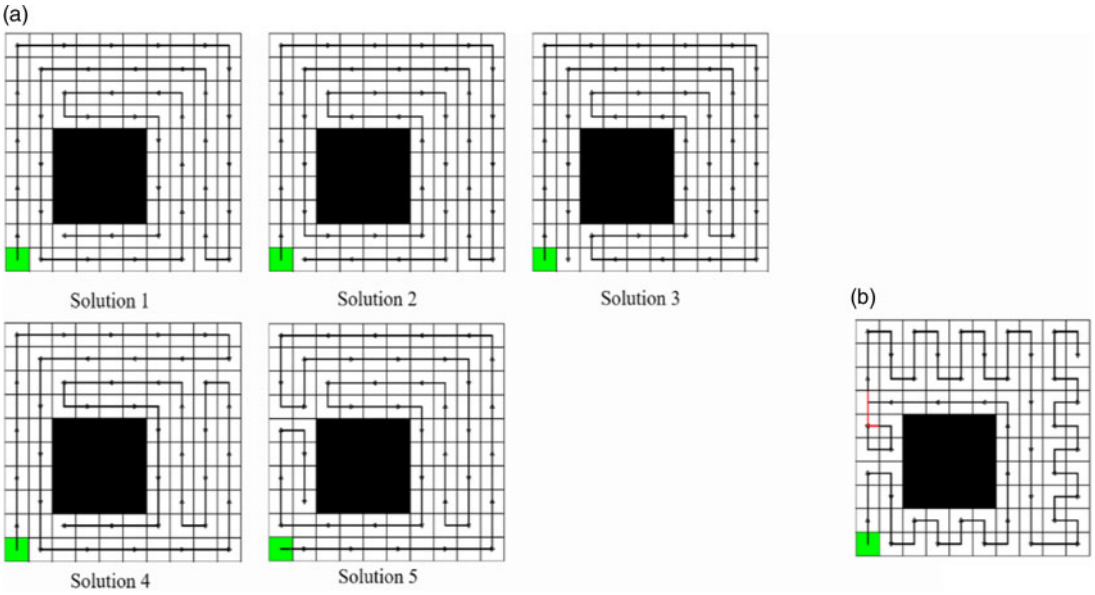

Figure 14. *The solutions generated by the proposed method (a) and traditional genetic algorithm (b) in [36] in square shape of obstacles.*

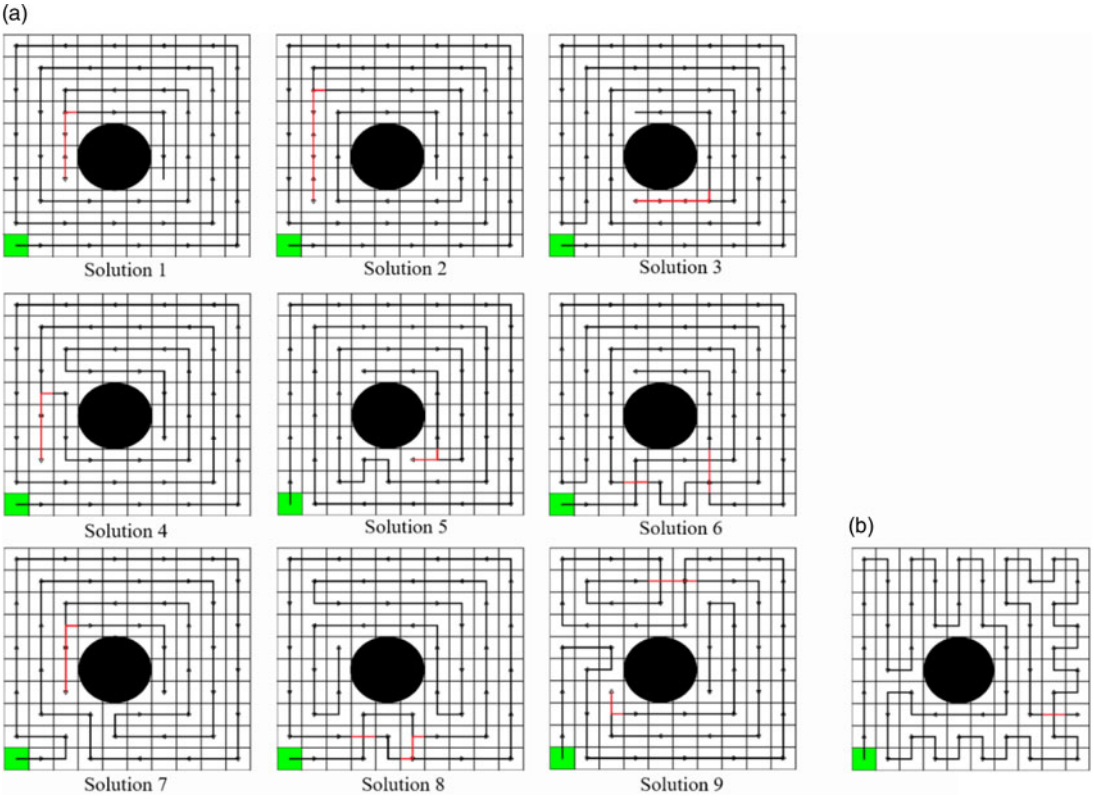

Figure 15. *The solutions generated by the proposed method (a) and traditional genetic algorithm (b) in [36] in circular shape of obstacles.*

https://doi.org/10.1017/S0263574724000377 Published online by Cambridge University Press

Table VIII. *Comparison of energy consumption by the mobile robot developed by the proposed method and hybrid genetic algorithm (HGA) in [30].*

Method	Number of turns	Number of U-turns	Distance traveled (cells)	Energy consumption (J)	Computation Time (s)
HGA and Proposed method solution 3	23	0	122	2986.3	270 (HGA) 385 (Solution 3)
Proposed method solutions 1, 2, and 4	19	0	122	2803.2	383 (Solution 1) 381 (Solution 1) 384 (Solution 1)
Proposed method Solution 5	19	0	130	2928.5	389 (Solution 5)

Figure 16. *The solutions generated by the proposed method (a) and traditional genetic algorithm (b) in [36] in irregular shape of obstacles.*

4.5. Comparison of the proposed method with the family of STC algorithms

The results from the proposed algorithm are compared with the family of STC algorithms – the spiral and full STC [23]. The comparison is conducted in three simulation environments, with two of them being taken from [24, 25], while the third environment is a new case study with complex shapes of obstacles. The proposed method generates optimal solutions for the first environment as illustrated in Fig. 20. The two best paths in terms of energy consumption among these solutions are shown in Fig. 20(a) and

https://doi.org/10.1017/S0263574724000377 Published online by Cambridge University Press

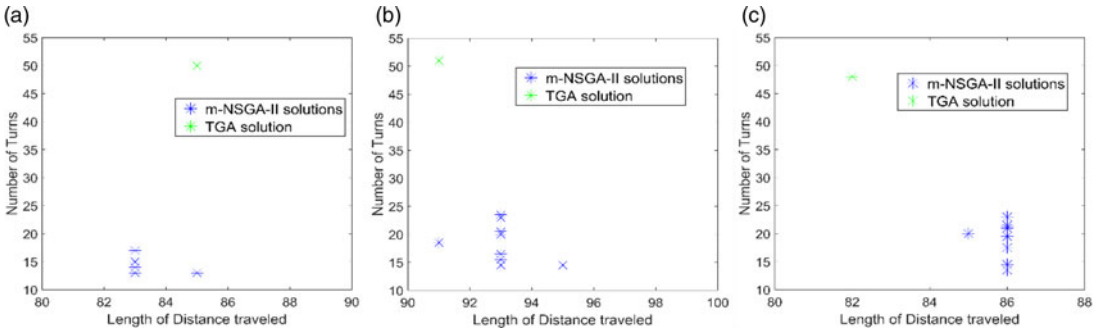

Figure 17. *Solutions generated by the proposed method and traditional genetic algorithm with (a) square, (b) circular, and (c) irregular shapes of obstacles.*

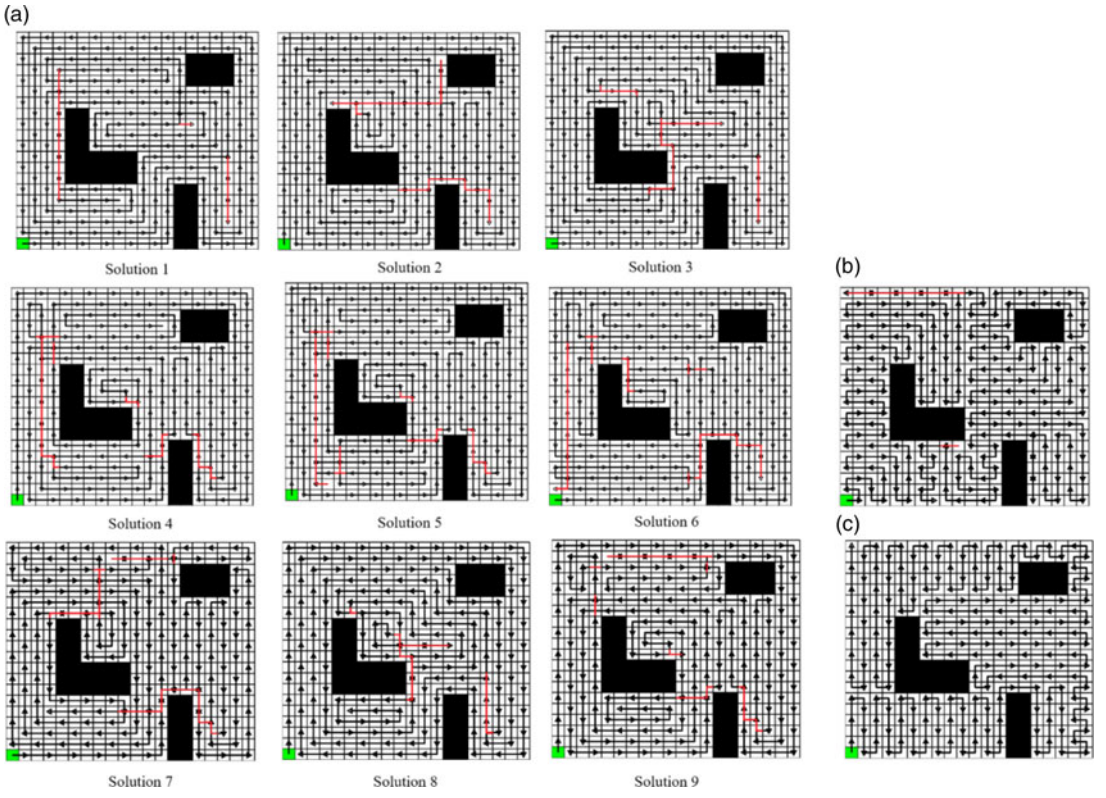

Figure 18. *The paths generated by (a) the proposed method, (b) spatial cell diffusion in [22], and (c) oriented rectilinear decomposition in [21] in an environment.*

Fig. 20(b). Further, Fig. 20(c) and Fig. 20(d) display the paths formed by spiral STC and full STC, respectively, and the spanning tree graph is represented by a thick blue line. The comparison of the solutions produced by the proposed technique, spiral STC, and full STC for the second and third environments is demonstrated in Fig. 21 and Fig. 22, respectively.

The algorithm in the first environment generates multiple-optimal solutions, which are indicated by blue asterisks in the objective space, as depicted in Fig. 23(a). The solution generated by the full STC approach is represented by a red asterisk in the figure. The optimal multiple solutions in the objective space produced by the proposed technique and the solution obtained by the full STC method for the second and third environments are illustrated in Fig. 23(b) and Fig. 23(c), respectively.

https://doi.org/10.1017/S0263574724000377 Published online by Cambridge University Press

Table IX. *Comparison of energy consumption developed by the proposed method and traditional genetic algorithm (TGA) approach in [36].*

Environment	Method	Number of turns	Number of U-turns	Distance traveled (cells)	Energy consumption (J)	Computation Time (s)
Square shape obstacle	TGA	50	0	85	3658.6	173
	Proposed method	13	0	83	1917.8	287
Circular shape obstacle	TGA	51	0	91	3798.4	176
	Proposed method	13	1	93	2132.8	289
Irregular shape obstacle	TGA	48	0	82	3520.1	186
	Proposed method	12	1	86	1977.4	307

Table X. *Comparison of energy consumption developed by the proposed method, spatial cell diffusion (SCD) in [22], and oriented rectilinear decomposition (ORD) in [21].*

Method	Number of turns	Number of U-turns	Distance traveled (cells)	Energy consumption (J)
Proposed method Solution 1	42	2	370	7778.4
SCD	120	3	353	13,162
ORD	106	0	349	12,017

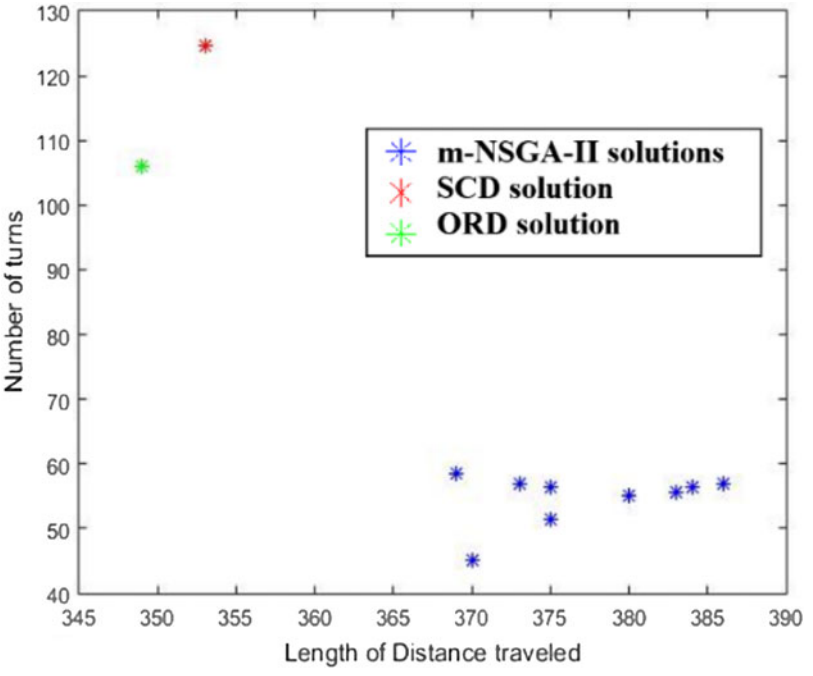

Figure 19. *Solutions generated by the proposed method, spatial cell diffusion [22] and oriented rectilinear decomposition [21].*

https://doi.org/10.1017/S0263574724000377 Published online by Cambridge University Press

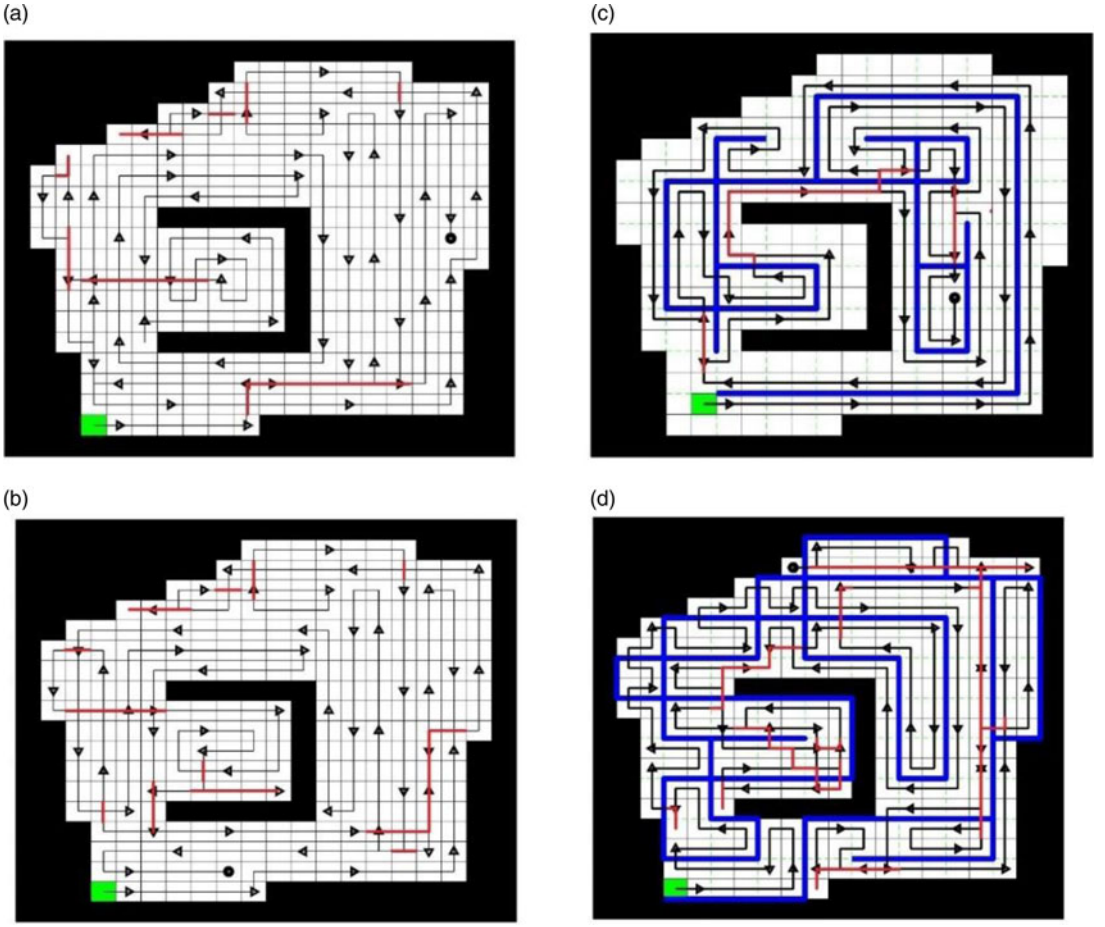

Figure 20. *(a and b) Paths generated by the modified Non-dominated Sorting Genetic Algorithm-2, (c) Spiral spanning tree coverage algorithm (STC), and (d) Full STC methods for the first environment in [24].*

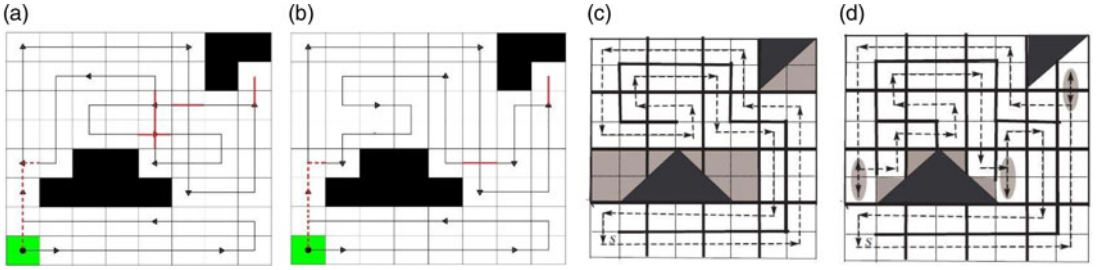

Figure 21. *(a and b) Paths generated by the modified Non-dominated Sorting Genetic Algorithm-2, (c) Spiral spanning tree coverage algorithm (STC), and (d) Full STC methods for the second environment in [25].*

Table XI presents a comparison of the proposed method with spiral and full STC algorithms in terms of energy consumption and coverage area by the mobile robot when completing the CPP task in three different environments. The proposed method reduces the energy consumption of the mobile robot in the first, second, and third environments by 33.8%, 5.8%, and 12.44%, respectively, compared to the full STC method. While the spiral STC method has the minimum energy usage across all the provided

https://doi.org/10.1017/S0263574724000377 Published online by Cambridge University Press

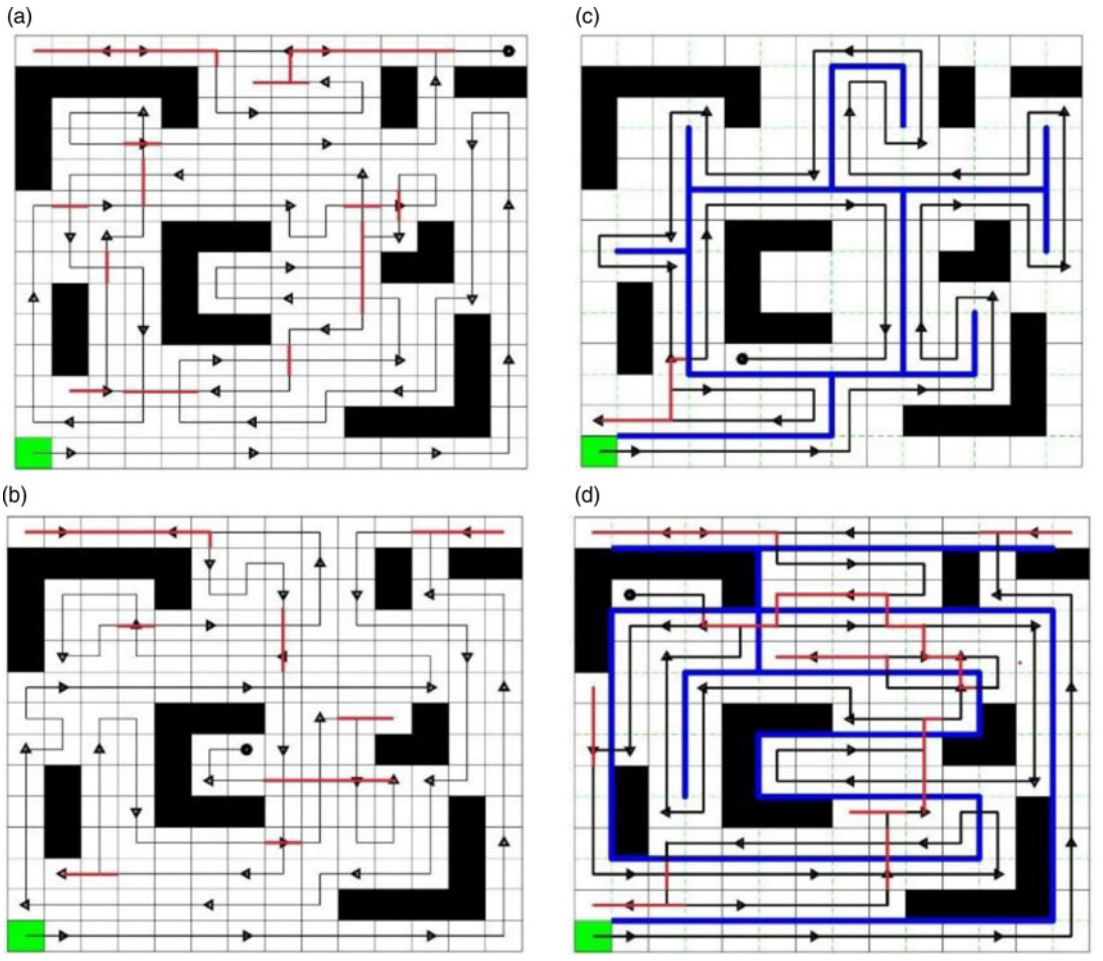

Figure 22. *(a and b) Paths generated by the modified Non-dominated Sorting Genetic Algorithm-2, (c) Spiral spanning tree coverage algorithm (STC), and (d) Full STC methods for the third environment.*

environments, its coverage rate is only 73.3%, 87.27%, and 71.6% in the first, second, and third environments, respectively. The proposed approach and full STC algorithm allow the robot to achieve complete coverage of 100%.

5. Conclusion and future scope

This paper proposes a modified NSGA-II approach for solving the MOO-CPP problem for an autonomous mobile robot in a variety of static rectilinear known environments. Using a grid-based approach, the working environment is discretized into cells, with the size and shape of each cell depending on the mobile robot. Unlike a SOO technique, a MOO method generates Pareto-optimal solutions that can be used based on the preference of user needs. The minimization of the total distance traveled and the total number of turns made by the mobile robot are the two objective functions employed in the proposed algorithm. The two objective functions are used to calculate the amount of energy the mobile robot requires to complete the CPP task. The energy consumption generated by the proposed algorithm is compared to that of the HGA, TGA, ORD, SCD, and family of STC methods. It is found that at least one solution among the multiple-optimal solutions generated by the proposed method consumes less energy than the solution generated by mentioned methods. It is observed that energy consumption by the mobile robot generated by the proposed method is approximately decreased by the range of 5%–60%.

https://doi.org/10.1017/S0263574724000377 Published online by Cambridge University Press

Table XI. *Comparison of energy consumption developed by the proposed method, spiral spanning tree coverage algorithm (STC), and full STC methods.*

Environments	Method	Number of turns	Number of U-turns	Distance traveled (cells)	Energy consumption (J)	Coverage in %
First environment [24]	Modified NSGA-II Solution 1	53	3	277	6978	100
	Modified NSGA-II Solution 2	52	2	280	6920	100
	Spiral STC	60	0	207	6026.9	73.3
	Full STC	116	4	311	10,453	100
Second environment [25]	Modified NSGA-II Solution 1	17	1	64	1877.5	100
	Modified NSGA-II Solution 2	19	1	60	1906	100
	Spiral STC	15	0	48	1712	87.27
	Full STC	19	3	58	1991.9	100
Third environment	Modified NSGA-II Solution 1	48	3	186	5324	100
Case study	Modified NSGA-II Solution 2	46	4	178	5165.6	100
	Spiral STC	41	0	120	3794.7	71.6
	Full STC	54	6	194	5899.3	100

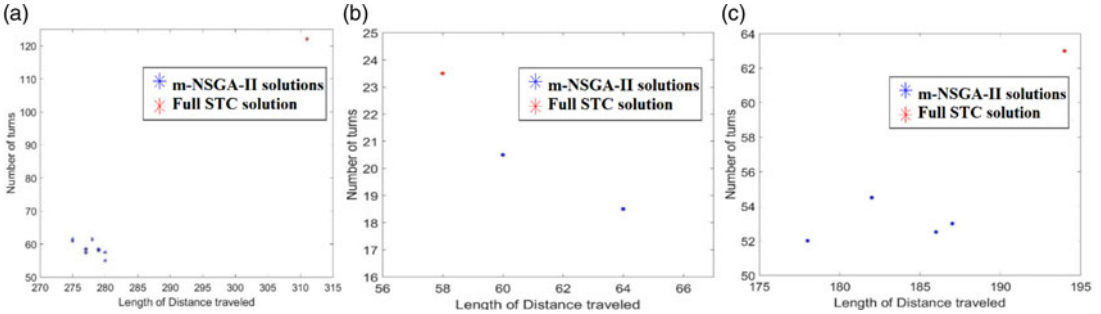

Figure 23. *Solutions with blue and red colors are generated by the modified Non-dominated Sorting Genetic Algorithm-2 algorithm and full spanning tree coverage algorithm method, respectively, of the first and the second environments (a and b) in [24, 25] and (c) case study.*

The conclusion is supported by previously published papers as well as case studies that were carried out in rectilinear environments with various shapes of obstacles. As the proposed method (modified NSGA-II) provides a diverse set of solutions (Pareto front results) compared to existing methods (TGA, HGA, ORD, SCD and STC), it (m-NSGA-II) takes higher computational time which is justified by the advantage and flexibility (multiple optimal solutions) it offers. It is to be noted that computational times are not a limitation as the proposed method is for offline path planning. Further, validation of an experiment can be conducted using a robot in a real-time environment. This research can be extended to environments of various non-rectilinear shapes and sizes. In addition to static obstacles, the environment can also contain moving obstacles. The studies can be extended with multi-agent robots with a larger scale of environments.

https://doi.org/10.1017/S0263574724000377 Published online by Cambridge University Press

Author contributions. Conceptualization, methodology, and simulation environment design, Hari Kumar Vorugnati; methodology, develop MATLAB code, and analysis, Monex Sharma.

Financial support. This research received no specific grant from any funding agency, commercial, or not-for-profit sectors.

Competing interests. The authors declare no Competing interests exist.

Ethical approval. None.

References

[1] V. V. M. J. S. Chembuly and H. K. Voruganti, "Trajectory planning of redundant manipulators moving along constrained path and avoiding obstacles," *Procedia Comput Sci* **133**, 627–634 (2018).
[2] X. Luo, S. Li, S. Liu and G. Liu, "An optimal trajectory planning method for path tracking of industrial robots," *Robotica* **37**(3), 502–520 (2019).
[3] F. O. Coelho, M. F. Pinto, P. Carvalho and A. L. M. Marcato, "Hybrid methodology for path planning and computational vision applied to autonomous mission: A new approach," *Robotica* **38**(6), 1000–1018 (2019).
[4] H. V. Pham, P. Moore and D. X. Truong, "Proposed smooth-STC algorithm for enhanced coverage path planning performance in mobile robot applications," *Robotics* **8**(2), 44 (2019).
[5] K.-C. Huang, F.-L. Lian, C.-T. Chen, C.-H. Wu and C.-C. Chen, "A novel solution with rapid voronoi-based coverage path planning in irregular environment for robotic mowing systems," *Int J Intell Robot Appl* **5**(4), 558–575 (2021).
[6] J. Hong, S. Yoo, I. Joo, J. Kim, K. S. Hwa and T. Seo, "Optimal parameter design of a cleaning device for vertical glass surfaces," *Int J Precis Eng Man* **20**(2), 233–241 (2019).
[7] M. Endo, S. Endo, K. Nagaoka and K. Yoshida, "Terrain-dependent slip risk prediction for planetary exploration rovers," *Robotica* **39**(10), 1883–1896 (2021).
[8] N. Karapetyan. *Robot Area Coverage Path Planning in Aquatic Environments, Diss* (University of South Carolina, Columbia, South Carolina, 2021).
[9] C. S. Tan, R. Mohd-Mokhtar and M. R. Arshad, "A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms," *IEEE Access* **9**, 119310–119342 (2021).
[10] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robot Auton Syst* **61**(12), 1258–1276 (2013).
[11] S. M. Ahmadi, H. Kebriaei and H. Moradi, "Constrained coverage path planning: Evolutionary and classical approaches," *Robotica* **36**(6), 904–924 (2018).
[12] T. Oksanen and A. Visala, "Coverage path planning algorithms for agricultural field machines," *J Field Robot* **26**(8), 651–668 (2009).
[13] H. Choset and P. Philippe, "Coverage Path Planning: The Boustrophedon Cellular Decomposition," In*Field and Service Robotics* (Springer, London, 1998) pp. 203–209.
[14] C. Howie, E. Acar, A. A. Rizzi and J. Luntz, "Exact cellular decompositions in terms of critical points of Morse functions," *IEEE Int Conf Robot* **3**, 2270–2277 (2002).
[15] R. Bähnemann., N. Lawrance, J. Chung., M. Pantic, R. Siegwart and J. Nieto, "Revisiting Boustrophedon Coverage Path Planning as a Generalized Traveling Salesman Problem," **In:** *12th International Conference-In Field and Service Robotics*,**16**, (2019) pp.277–290.
[16] C. Huang, W. Li, C. Xiao, B. Liang and S. Han, "Potential field method for persistent surveillance of multiple unmanned aerial vehicle sensors," *Int J Distrib Sens N* **14**(1), 155014771875506 (2018).
[17] H. V. Pham and T. N. Lam, "A new method using knowledge reasoning techniques for improving robot performance in coverage path planning," *Int J Comput Appl T* **60**(1), 57–64 (2019).
[18] Z. Cai, S. Li, Y. Gan, R. Zhang and Q. Zhang, "Research on complete coverage path planning algorithms based on A*Algorithms," *Open Cybernet Syst J* **8**(1), 418–426 (2014).
[19] E. P. Gonzalez, O. Alvarez, Y. Díaz, C. Parra and C. Bustacara, "BSA: A Complete Coverage Algorithm," **In:** *IEEE International Conference on Robotics and Automation*, (2005) pp. 2040–2044.
[20] M. Mitschke, N. Uchiyama and O. Sawodny, "Online Coverage Path Planning for a Mobile Robot Considering Energy Consumption," **In:** *IEEE 14th International Conference on Automation Science and Engineering (CASE)*, (2018) pp. 1473–1478.
[21] B. Dugarjav, S.-G. Lee, D. Kim, J. H. Kim and N. Y. Chong, "Scan matching online cell decomposition for coverage path planning in an unknown environment," *Int J Precis Eng Man* **14**(9), 1551–1558 (2013).
[22] J. Prado and L. Marques, "Energy Efficient Area Coverage for an Autonomous Demining Robot, ROBOT2013," **In:** *First Iberian Robotics Conference*, **2**, (2014) pp. 459–471.
[23] Y. Gabriely. and E. Rimon, "Spanning-tree based coverage of continuous areas by a mobile robot," *IEEE Int Conf Robot Auto* **31**, 77–98 (2001).
[24] K. R. Guruprasad, "X-stc: An Extended Spanning Tree-Based Coverage Algorithm for Mobile Robots," **In:** *Proceedings of the Advances in Robotics*, (2019) pp. 1–6.

https://doi.org/10.1017/S0263574724000377 Published online by Cambridge University Press

[25] K. R. Guruprasad and T. D. Ranjitha, "CPC algorithm: Exact area coverage by a mobile robot using approximate cellular decomposition," *Robotica* **39**(7), 1141–1162 (2021).
[26] V. G. Nair and K. R. Guruprasad, "GM-VPC: An algorithm for multi-robot coverage of known spaces using generalized voronoi partition," *Robotica* **38**(5), 845–860 (2019).
[27] A. E. F. Ryerson and Q. Zhang, "Vehicle path planning for complete field coverage using genetic algorithms," *Agri Eng Int: CIGR Ejournal* **9**, 1–11 (2007).
[28] M. A. Yakoubi and M. T. Laskri, "The path planning of cleaner robot for coverage region using genetic algorithms," *J Innov Digi Ecosyst* **3**(1), 37–43 (2016).
[29] T. R. Schafle, S. Mohamed, N. Uchiyama and O. Sawodny, "Coverage Path Planning for Mobile Robots Using Genetic Algorithm with Energy Optimization," **In:** *2016 International Electronics Symposium (IES)* (IEEE, Denpasar, Indonesia, 2016) pp. 99–104.
[30] T. R. Schäfle, M. Mitschke and N. Uchiyama, "Generation of optimal coverage paths for mobile robots using hybrid genetic algorithm," *JRobot Mechatr* **33**(1), 11–23 (2021).
[31] M. Sharma and H. K. Voruganti, "Preference Based Multi-Objective Optimization Techniques for Coverage Path Planning of a Mobile Robot," **In:** *Proceedings of the Advances in Robotics*, (2023) pp. 1–6.
[32] S. Gao-gao., G. Liu, P. Zhu and H. Jian, "Complete coverage path planning for horticultural electric tractors based on an improved genetic algorithm," *J Appl Sci Engi* **24**(3), 447–456 (2021).
[33] M. Sharma and H. K. Voruganti, "An effective genetic algorithm based multi-objective optimization approach for coverage path planning of mobile robot," *Comm Com Inf Sc* **1738**, 444–457 (2022).
[34] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans Evolut Comput* **6**(2), 182–197 (2002).
[35] H. Wang, S. Lou, J. Jing, Y. Wang, W. Liu and T. Liu, "The EBS-A* algorithm: An improved A* algorithm for path planning," *PLOS ONE* **17**(2), e0263841 (2022).
[36] S. Gajjar, J. Bhadani, P. Dutta and N. Rastogi, "Complete Coverage Path Planning Algorithm for known 2d Environment, 2017," **In:** *2nd IEEE International Conference on Recent Trends in Electronics*, (Information & Communication Technology (RTEICT), 2017) pp. 963–967.

Cite this article: M. Sharma and H.K. Voruganti, " Multi-objective optimization approach for coverage path planning of mobile robot", Robotica. https://doi.org/10.1017/S0263574724000377