Robotica: (2019) volume 37, pp. 1363–1382. © Cambridge University Press 2019
doi:10.1017/S026357471800156X

Evolutionary Algorithms-Based Multi-Objective Optimal Mobile Robot Trajectory Planning

V. Sathiya†* and M. Chinnadurai‡

† *Department of Electronics and Communication Engineering, E.G.S. Pillay Engineering College, Nagapattinam, Tamil Nadu 611002, India*
‡ *Department of Computer Science and Engineering, E.G.S. Pillay Engineering College, Nagapattinam, Tamil Nadu 611002, India*
E-mail: mchinna81@gmail.com

(Accepted December 24, 2018. First published online: March 7, 2019)

SUMMARY
In this research study, trajectory planning of mobile robot is accomplished using two techniques, namely, a new variant of multi-objective differential evolution (heterogeneous multi-objective differential evolution) and popular elitist non-dominated sorting genetic algorithm (NSGA-II). For this research problem, a wheeled mobile robot with differential drive is considered. A practical, feasible and optimal trajectory between two locations in the presence of obstacles is determined through the proposed algorithms. A safer path is obtained by optimizing certain objectives (travel time and actuators effort) taking into account the limitations of mobile robot's geometric, kinematic and dynamic parameters. Robot motion is represented by a cubic NURBS trajectory curve. The capability of the proposed optimization techniques is analyzed through numerical simulations. Results ensure that the proposed techniques are more desirable for this problem.

KEYWORDS: Wheeled mobile robot; Differential drive; Optimal trajectory; HMODE; NSGA-II; NURBS.

1. Introduction

Wheeled mobile robot (WMR) is increasingly employed in a great number of potential applications in indoor environments (e.g., cleaning, search and rescue operations, surveillance, helping the handicapped or senior people) and outdoor environments (e.g., material transport, security patrols). An important characteristic of robot–environment interaction is given by the ability of the control system to autonomously navigate through both unstructured and structured environments. A structured environment is fundamentally a predictable space that is evidently and precisely defined. This kind of environment is stationary and it has no variables (i.e., a robot knows what to anticipate when traversing through it at all times). The unstructured environment is an unpredictable and messy environment with unforeseen and infinite variables such as human intervention, moisture, temperature, lighting, etc. Navigation in this environment is an arduous task for a mobile robot since it must be accomplished by recognizing and adapting to these vicissitudes.

Trajectory planning of a mobile robot refers to the task of discovering an optimal path by optimizing certain objective functions like the travelling time, joints accelerations and actuator efforts while satisfying certain kinematic and dynamic constraints for avoiding obstacles. It is an unavoidable major issue for the purpose of designing and programming of the mobile robot, and, by virtue

* Corresponding author. E-mail: sathiyav2105@gmail.com

of trajectory planning, only the WMR completes its missions. In a mobile robot, the trajectory planning software module does trajectory planning. To increase the operational speed of the robot, which influences industrial productivity, it is essential to diminish the overall travelling time. Since most of the mobile robots are batteries powered, their operation times and energy are limited. Hence, in what way to minimize energy consumption and keep mobile robots to navigate toward the target becomes a significant issue. Energy conservation for robotic applications can be accomplished in many different ways, for instance, by means of energy-efficient devices, energy-efficient motion planning, etc. Therefore, the minimum energy consumption and the minimum travelling time of the drive are considered as the main criteria of trajectory planning. Moreover, during the navigation of the mobile robot, it has to avoid moving and stationary obstacles. For smooth and practical trajectory planning, both kinematic and dynamic limitations of the mobile robot are to be considered.

There have been many different decision criteria like the shortest trajectory or path length, kinetic energy, potential energy, loss of kinetic energy, reducing steering actions of mobile robot actuators, actuator efforts, the smoothness of the trajectory, motor energy consumption used by many researchers for energy optimal trajectory planning. In this work, both the minimum traveling time and the minimum actuators effort are considered together as objective functions. According to the nature of the environment present around the mobile robot, trajectory planning is done in off-line as well as online. If the environment around the robot is structured and well known to the robot, off-line motion planning is the best option. Conversely, for an unknown unstructured environment, online motion planning has to be carried out. Many implicit and explicit methods are proposed in the literature for solving energy-efficient trajectory planning problem. Likewise, many researchers have treated this complex problem using traditional and intelligent algorithms.[1–4]

Here is an account of various research in mobile robot trajectory planning. Kim and Kim[5] presented a trajectory planning approach for a differential drive WMR. They considered static obstacles to the robot path. They presented some simulation results using the bang-bang principle to minimize the overall travelling time by considering the dynamics of the robot. Walambe et al.[6] suggested a trajectory planning technique for the prototype of a car. They considered the non-holonomic system (i.e., its wheels assumed to be rolling without slipping and skidding conditions) of the car with sideways movements as constraints. They developed a spline-based framework to handle the problem of parametric singularities. Kim and Kim.[7] suggested an optimal energy trajectory planning method of a mobile robot using Pioneer 3-AT platform. Minimizations of robot velocity and battery voltage (energy) were considered as objective functions.

Sgorbissa and Zaccaria[8] presented a method of robot navigation in a partially unknown environment. They taught the robot to have both prior knowledges related to the environment and local perceptions. They conducted a few real-world experiments to train the robot to have safe movement. Another work focused on the time-optimal motion designing strategy for a car shape mobile robot was developed by Li and Shao[9] The authors used interior-point method and an optimality index based on Hamiltonian to find and evaluate the trajectories. Further, they considered few kinematic constraints (terminal constraints), physical parameters bounds and collision avoidance constraints.

Liu and Sun[10] employed a method for optimization of energy consumption of the WMR. A* algorithm was used. Time and velocity constraints were considered. Han et al.[11] applied an optimal trajectory planning algorithm for a WMR with a differential drive. They used a single curvature trajectory planning method and considered tracking error as a criterion to be minimized. The authors conducted few experiments using a two-wheeled mobile robot. Korayem et al.[12] introduced a methodology for motion designing for Scout mobile manipulator in chaotic environments. A collision avoidance criterion was considered as the main goal. All parts of mobile robot and obstacles were considered as ellipsoids. Using ellipsoid equations, they verified the obstacle avoidance algorithm in both simulations and experiments.

Boryga et al.[13] modeled and investigated the planning rectilinear-arc polynomial trajectory (PRAPT) approach for trajectory planning of a tomato harvesting mobile manipulator by considering few kinematic constraints like displacement, velocity and acceleration. To represent a trajectory, two rectilinear curve segments had been used. A trajectory planning strategy for a system composed of a WMR was studied by Korayem et al.,[14] where the robot is equipped with both rigid and elastic joints. They modeled WMR as a non-holonomic system and used a joint coordinate system for trajectory planning. To derive kinematic and dynamic equations of robot and to reduce computing complexity, the Lagrange multipliers and recursive Gibbs–Appell (G–A) formulation were used. Korayem et al.[15] also presented a mathematical modeling and trajectory planning method of a two-link flexible non-holonomic mobile manipulator using Pontryagin's minimum principle. They considered kinematic and dynamic parameters of the mobile robot. Mirzaeinejad and Shafei[16] proposed a new solution which combines some prediction-based and G–A-based approaches for trajectory modeling and tracking control of a WMR. Each method discussed in literature[1–16] has its own limitations. Further, the conventional optimization methods[3–16] presented in the literature are not powerful enough to solve high-complexity robot trajectory optimization problems. Thus, it is necessary to reinvestigate an efficient approach for the motion designing of WMR in cluttered environments, which encompasses a simple, yet powerful technique to avoid collision with obstacles.

The process of finding an optimal path includes consideration of decision criteria and constraints. The optimal path has to yield optimal objective functions and satisfy certain constraints. The optimal path has to be selected based on several aspects, viz., the short-time approaches, energy-saving needs, obstacle avoidance, collision avoidance, path smoothness, path safety, limiting parameters like changing velocity and direction, etc. An optimal path is a collision-free trajectory in between two locations. Hence, an optimal path finding is a multi-objective optimization problem.[17] The path-finding algorithm has to simultaneously optimize multiple objective functions and ensure that the constraints (i.e., limitations on certain geometrical and motion parameters) are satisfied. Also, the path-finding algorithm is associated with obstacle detection and collision avoidance.[18] To solve this high-complexity multi-model problem, evolutionary techniques are generally proposed. Nevertheless, the main drawbacks of nature-inspired algorithms are time-consuming and high computational cost in finding an optimal solution. In some cases, they yield locally optimal solutions. Consequently, several research attempts have been made to enhance the performance of evolutionary techniques.

At present, evolutionary techniques are the most salient and promising research areas for multimodal and multi-objective optimization problems,[19–21] which have experienced a hasty development and have become powerful tools in a wide range of applications. They are very effective and robust in finding trade-off solutions. But, conventional techniques can handle a single criterion or a single objective function at a time. They cannot deal with optimization problems involving two or more objectives to be optimized concurrently. So, a combined objective function is formed by mathematically aggregating two or more individual objective functions. A weightage value is introduced into the combined objective function equation to reflect its relative importance. Evolutionary techniques consider all the objective functions and the necessary constraints and treat each objective function separately. Instead of finding one global optimum, multi-objective optimization methods must find a set of solutions, called as the Pareto optimal frontier, which gives more choice of user selection. Improved evolutionary algorithms find the best Pareto fronts with less execution time and less number of iterations. Further, they may discover a global optimal solution.

In this research, a new variant of differential evolution (DE), namely heterogeneous multi-objective differential evolution (HMODE) strategy is implemented for trajectory planning of WMR. Additionally, another popular multi-objective optimization algorithm, called elitist non-dominated sorting genetic algorithm (NSGA-II) is implemented in this study. A WMR with the differential drive is considered. The proposed algorithms determine a feasible and optimal trajectory between two locations by avoiding few obstacles and optimizing certain criteria such as travelling time and actuators effort. Also, the algorithms yield a safer path by considering geometric, kinematic and dynamic limitations of mobile robot. Five applications of WMR are considered in this study. A cubic Non Uniform Rational B-Spline trajectory curve is employed to represent the robot motion. Average fuzzy membership function (AFMF) method is used to select the best optimal trade-off solution from the Pareto optimal front. The strength and weakness of the solutions (i.e., Pareto optimal fronts) are determined by the metrics such as ratio of non-dominated individuals (RNI) and solution spread measure (SSM). Also, algorithm effort (AE) and optimizer overhead (OO) are applied to analyze the computational effort of the algorithms.

The remaining part of this article is structured as follows: Section 2 defines the multi-objective optimal motion planning problem addressed in this work. Section 3 formally describes the proposed algorithms viz., NSGA-II and HMODE and their implementations. Section 4 enumerates the performance measures used for analyzing the effectiveness of the proposed methods. Section 5 details five numerical simulations for a WMR with a differential drive in the presence of static and moving obstacles and also presents evaluation criteria supporting our statements. Section 6 summarizes the conclusions derived from this research work.

2. Problem Formulation

2.1. Optimal trajectory planning

The goal of this research is to find the best optimal trajectory planning for a WMR in five different applications. WMR starts its motion from the start point (S) and reaches the goal point (G). At the same time, it has to avoid the collision risk with obstacles in the environment. We consider a robot with five different applications. Application 1 has two fixed vertical obstacles. Application 2 has WMR motion within a room which has four walls and two fixed obstacles. Application 3 has nine fixed obstacles. Application 4 has three moving obstacles. In Application 5, the goal point is not fixed. It is moving.

2.2. Multi-objective optimization problem

The focus of the paper is to move the mobile robot from initial position to target point through stationary and moving obstacles, and finding the optimal path to reach the final position without collision. The mobile robot motion has to be accomplished by minimizing the travelling time and actuator efforts and with obstacle avoidance. To get a feasible and smooth motion, the geometric, kinematic and dynamic constraints of the mobile robot have to be satisfied. The mathematical model of problem formulation is presented here. Two objective optimization functions are proposed for minimizing travelling time $F_1(P)$ and actuators effort $F_2(P)$.

$$F_1(P) = T = \sum_{i=1}^{n} dt$$

$$F_2(P) = \int_0^T \sum_{i=1}^{2} \left[\frac{\tau_i(t)}{\tau_i^{max}} \right]^2 dt$$

Here, τ is actuator torque (i.e., DC motor torque), T is travelling time of the robot.

Constraints considered are as follows:

1. Geometric constraints:
   (a) Initial and final configurations, that is, positional values are as defined by the user.
   (b) Obstacle avoidance constraint, that is, the distance between the driving wheels and the obstacle are to be greater than zero.
2. Kinematic constraints:
   (a) Initial and final velocities are zero.
   (b) The velocity of the robot $<$ maximum velocity.
   (c) Acceleration of the robot $<$ maximum acceleration.
   (d) Jerk of the robot $<$ maximum jerk.
3. Dynamic constraint:
   The torque of the robot actuators $<$ maximum torque of actuators.

2.3. Description of the mobile robot

We consider herein is a WMR with a differential drive. It has two autonomously controlled active wheels and two supporting passive wheels. The geometrical representation of WMR with a differential drive is given by Fig. 1. The moving configuration of the mobile robot (q) in the frame (O, X, Y, Z) is given by the following equation. Here (x, y) is the centre position of the robot and θ is the rotation angle about the Z-axis:

$$q = [x, y, \theta]^T \tag{1}$$

2.4. WMR kinematics[22]

The kinematic model of a WMR is detailed as below:

$$\dot{q} = S(q)v \tag{2}$$

https://doi.org/10.1017/S026357471800156X Published online by Cambridge University Press

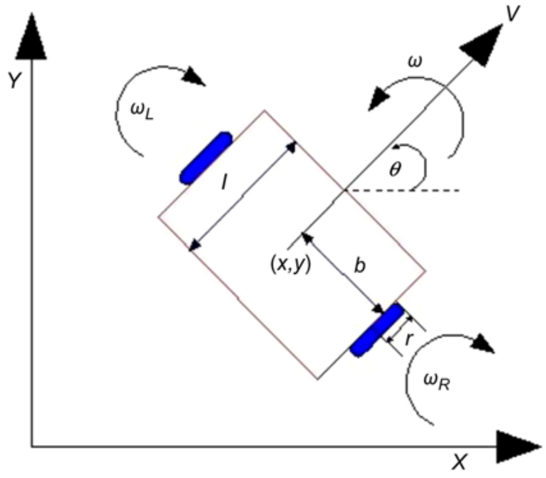

Fig. 1. Geometrical configuration of wheeled mobile robot.

Here,

$$v = [v\omega]^T,\ S(q) = \begin{bmatrix} \cos\theta & 0 & 0 \\ \sin\theta & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

v and ω are the linear and angular velocity of the WMR, respectively. Then, Eqs. (2) and (3) can be rewritten as

$$\dot{x} = v\cos\theta,\ \dot{y} = v\sin\theta,\ \dot{\theta} = \omega \tag{4}$$

Owing to the non-holonomic nature of the system, the constraint equation satisfies the ideal no-slip condition as given below:

$$C(q)\dot{q} = 0 \tag{5}$$

Here,

$$C(q) = \lfloor \sin\theta \quad -\cos\theta \quad 0 \rfloor \tag{6}$$

From Eqs. (3) and (4), the important kinematic parameters v, $\dot{v}$, θ, $\dot{\theta}$ and $\ddot{\theta}$ are derived as follows:

$$v = \sqrt{\dot{x}^2 + \dot{y}^2} \tag{7}$$

$$\theta = arctan\,(\dot{y}/\dot{x}) \tag{8}$$

$$\dot{v} = (\dot{x}\ddot{x} + \dot{y}\ddot{y})/\left(\sqrt{\dot{x}^2 + \dot{y}^2}\right) \tag{9}$$

$$\dot{\theta} = (\dot{x}\ddot{y} + \dot{y}\ddot{x})/(\dot{x}^2 + \dot{y}^2) \tag{10}$$

$$\ddot{\theta} = \left((\dot{x}^2 + \dot{y}^2)(\dot{x}\dddot{y} - \dot{y}\dddot{x}) - 2(\dot{x}\ddot{x} + \dot{y}\ddot{y})(\dot{x}\ddot{y} + \dot{y}\ddot{x})\right)/(\dot{x}^2 + \dot{y}^2)^2 \tag{11}$$

2.5. *WMR dynamics*[22]

The dynamic model of a WMR is obtained from the Lagrange's equations as below and the model satisfies the robot's kinematic constraints (Eq. (5)):

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}}\right) - \frac{\partial T}{\partial q} + C^T\lambda = Q \tag{12}$$

where $q =$ generalized coordinate vector, $\dot{q} =$ a vector of longitudinal and angular velocities of generalized coordinates, KE = the kinetic energy of the robot, $\lambda =$ a vector of a Lagrange multiplier and C is derived from Eq. (6). Equation (12) is rewritten as below:

$$M(q)\ddot{q} + V(q, \dot{q}) = E(q)\tau - C^T(q)\lambda \tag{13}$$

https://doi.org/10.1017/S026357471800156X Published online by Cambridge University Press

Here,

$$M = \begin{pmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_z \end{pmatrix}, \quad V = 0 \tag{14}$$

$$E = \frac{1}{r}\begin{pmatrix} \cos\theta & \sin\theta \\ \sin\theta & \cos\theta \\ b & -b \end{pmatrix}, \quad \tau = \begin{pmatrix} \tau_r \\ \tau_l \end{pmatrix} \tag{15}$$

where $M =$ inertia matrix of the system, $m = m_1 + 2m_2$, $m_1 =$ the mass of the robot's platform, $m_2 =$ the mass of one robot's wheel, $I_z =$ the inertia moment of robot's platform about (O, Z), $r =$ the radius of the wheels, $b =$ half of the distance between the two wheels. τ_l and τ_r are torques provided by the actuators acting on left and right wheels, respectively. This model assumes that the inertia moments of wheels are very small and negligible. The differentiation of Eq. (2) gives the acceleration:

$$\ddot{q} = \dot{S}\,(q)\,v + S(q)\dot{v} \tag{16}$$

By replacing $\ddot{q}$ in Eq. (13), and by doing pre-multiplication of S^T, the expression for $\dot{v}$ is

$$\dot{v} = f\tau \tag{17}$$

Here,

$$f = (S^T M S)^{-1} S^T E, \quad \dot{v} = \begin{bmatrix} \dot{v} & \dot{\omega} \end{bmatrix}^T \tag{18}$$

Then, the DC motors torque is described as

$$\begin{pmatrix} \tau_r \\ \tau_l \end{pmatrix} = f^{-1}\begin{pmatrix} \dot{v} \\ \dot{\omega} \end{pmatrix} \tag{19}$$

$$f^{-1} = \begin{pmatrix} \dfrac{mr}{2} & \dfrac{rI_z}{2b} \\ \dfrac{mr}{2} & \dfrac{-rI_z}{2b} \end{pmatrix} \tag{20}$$

2.6. *Path representation*

NURBS curve has various highly desirable properties like smoothness and possibility of local modifications. These properties are crucial for curve designing because one can change a curve locally without modifying their underlying geometry in a global way. Therefore, the NURBS curve is widely employed in computer graphics, CAD, CAM, CIM and robotics. In this work, a cubic NURBS function is used to represent the trajectory of the autonomous mobile robot. The parametric representation of a general NURBS curve is given below:

$$P(u) = \frac{\sum_{i=0}^{n} N_{i,k}(u) W_i V_i}{\sum_{i=0}^{n} N_{i,k}(u) W_i} = \sum_{i=0}^{n} V_i R_{i,k}(u) \tag{21}$$

and

$$R_{i,k}(u) = \frac{N_{i,k}(u) W_i}{\sum_{i=0}^{n} N_{i,k}(u) W_i} \tag{22}$$

In the above formulae, *P(u)* represents robot displacement, $N_{i,k}(u)$ denotes kth-order basis function, V_i represents the control point, W_i represents weighting factor of the control point, n is the number of the control points, k is the order of NURBS curve and $R_{i,k}(u)$ denotes the kth order rational basis function.

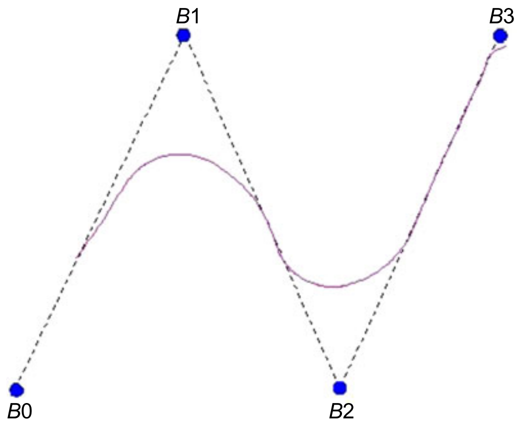

Fig. 2. A segment of cubic NURBS curve.

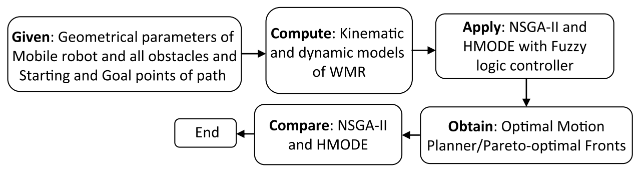

Fig. 3. Flowchart for the proposed method of doing optimal motion planning.

Totally four segments are considered in the path of initial and ending points. Three intermediate points are considered for each segment. As mentioned above, each segment is defined by a cubic NURBS curve with four control points as given in Fig. 2. $W_i = \{1, 2, 2, 1\}$ is the weight vector assumed for the control points in all curve segments. A NURBS curve is completely determined by its control point (i.e., the curve change in an anticipated way along with the movement of control points). The weighting factor W_i plays a vital role in the fitting process since these factors determine how much a control point influences locally the geometry of the curve. If the value of the weight increases, it pulls the NURBS curve toward the control point. If the weight value decreases, it pushes the NURBS curve far from the control point. If the value of W_i is infinity, the curve will go through the control point. If the value of W_i is zero, then control points do not have an impact on the curve. If the number of control point increases, more accurate and better curve fit is achieved. But, the number of control points increasing leads to the need for more computational effort and time. The positions of control points V_i are considered as decision variables in this multi-objective optimal robot motion planning problem.

3. Proposed Methods

Generally, traditional optimization methods yield sub-optimal results. But intelligent optimization algorithms yield optimal solutions. Especially evolutionary algorithms, viz., genetic algorithm (GA), NSGA-II, DE, etc., give optimal solutions. Then, these optimal solutions can be converted into a global optimum. The effective and robust nature of these evolutionary algorithms makes them the best fit for the multi-objective optimization problems. The outcome of the multi-objective optimization algorithm is the Pareto-optimal front. It gives more trade-off solutions to users' choice. A number of performance measures are available to find the effectiveness of a multi-objective optimization algorithm.[19] In this research work, both HMODE and NSGA-II algorithms are used for multi-objective motion planning of a WMR. Figure 3 explains the procedure involved in this multi-objective optimal motion planning problem.

3.1. NSGA-II

NSGA-II is a variant of GA and it is a most popular algorithm for an optimization problem with multi-objectives.[23] Major differences between NSGA and NSGA-II are as follows: (1) Elitism principle is implemented in the NSGA-II for preserving good solutions found in previous iterations. (2) As

https://doi.org/10.1017/S026357471800156X Published online by Cambridge University Press

Table I. CP vs. MP.

	CP				
MP	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
0.01	22.6	22.5	**22.4**	22.5	22.6
0.02	22.6	22.5	22.5	22.6	22.6
0.03	22.7	22.6	22.5	22.6	22.6
0.04	22.7	22.6	22.5	22.6	22.7
0.05	22.8	22.6	22.5	22.6	22.7
0.06	22.9	22.7	22.5	22.7	22.7
0.07	23	22.7	22.5	22.7	22.7
0.08	23.1	22.7	22.6	22.7	22.7
0.09	23.1	22.7	22.6	22.7	22.8
0.1	23.1	22.8	22.6	22.7	22.8

Note: The least value is given in bold.

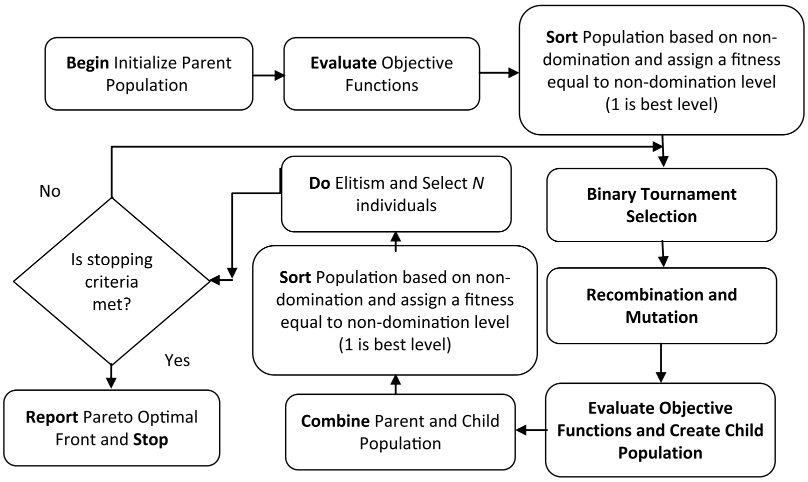

Fig. 4. The step by step procedure of NSGA-II.

compared to NSGA, the sorting procedure of the NSGA-II is faster. (3) The tunable parameter is not needed in NSGA-II. So, NSGA-II algorithm does not dependent on the programmer. Figure 4 shows the step-by-step procedure in NSGA-II algorithm.

3.1.1. NSGA-II operators. Deb et al.[23] recommended that crossover probability (CP) can be fixed between 0.5 and 0.9, mutation probability (MP) can be fixed between 0.01 and 0.1, real-parameter mutation parameter is 100 and simulated binary crossover (SBX) value of real parameter is 10. So to fix the CP and MP, an analysis was made by varying CP value from 0.5 to 0.9 and MP value from 0.01 to 0.1, the optimal travel time (F_1) in Application 1 was found. The results are given in Table I.

From Table I, it is observed that when the value of CP is 0.7 and MP is 0.01, we have got the minimum value for F_1. So, CP value is fixed as 0.7 and MP is fixed as 0.01. NSGA-II parameters value are: population size is 100, variable type is real, CP is 0.7, real-parameter mutation parameter and probability are 100 and 0.01, respectively, SBX value of real-parameter is 10 and maximum number of generations is 100.

3.2. HMODE

MODE algorithm[24] is a variant of GA for multi-objective function optimization. It is an extended version of DE algorithm. MODE is an evolutionary algorithm with floating-point-encoded scheme. Since the context of the MODE is very simple, it needs less computational work, less storage space and less CPU run time. Due to its simple configuration, robustness, speed and ease of use, MODE gained much attention. Also, it is used to resolve a multi-objective optimization problem of all

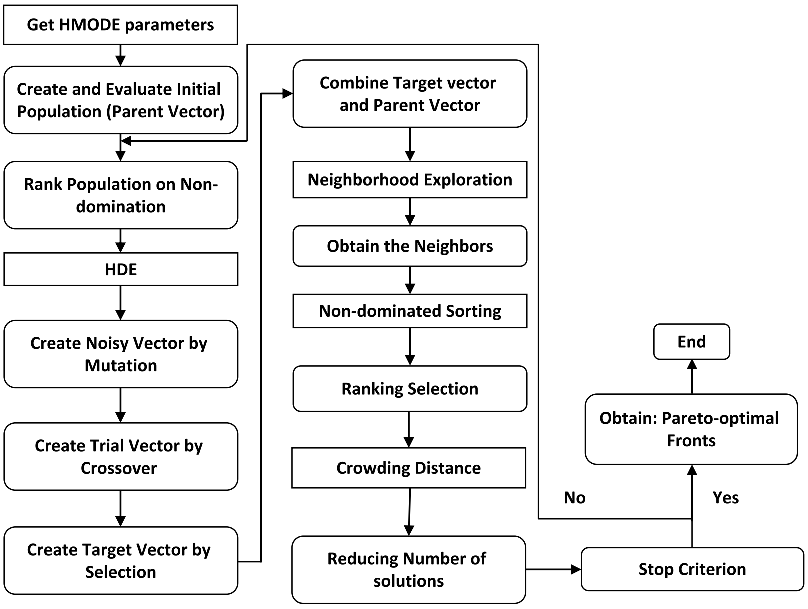

Fig. 5. Flowchart of HMODE procedure.

fields. MODE algorithm provides the best Pareto-optimal frontier for more choice of users. Babu and Anbarasu[24] used MODE for solving chemical engineering problems. But, the limitations of their work are: (1) They used only one variant of DE (i.e., DE/rand/1/bin). They have not considered important and best-performing variants, namely DE/current-to-best/1, DE/rand-to-best/1 and DE/TSDE. (2) Further, the Scaling factor F is fixed one. However, flexible F-value leads to the best solutions.

With multiple forms of schemes for generating noisy vector by differential mutation, there exist new variants of DE. The DE algorithm has been employed in several fields of science and engineering. But, no single DE variant can solve multi-model, complex and real-world problems. Each DE variant has its own preferences than another. To give the global best solution to the real-world problem, a DE variant needs to have all the necessary advantages and the best possible solution strategies. One simple idea to have such a DE variant is that the creation of the best heterogeneous differential evolution (HDE) variant. Two to five different best performing DE variants can be integrated. It creates an environment which has HDE variants that work on an island-based distributed structure. Thangavelu and Shunmuga Velayutham[25] carried out a study about the consequences of integrating different DE variants. They have conducted an investigation in a distributed framework and proved that HDE works well. However, the limitations of their work are: (1) They considered only four basic DE variants such as DE/rand/1/bin, DE/best/2/bin, DE/best/1/bin and DE/rand/2/bin, and they have not considered important and best-performing variants, namely DE/current-to-best/1, DE/rand-to-best/1 and DE/TSDE. (2) Further, they have not tested their method in a real-world situation. (3) They have not verified their findings for a constrained problem. (4) They have not tested their method for a multi-objective optimization problem.

To overcome the above barriers, the improvements made in the proposed HMODE technique over the method proposed by Thangavelu and Shunmuga Velayutham[25] are: (1) Best-performing variants, namely DE/current-to-best/1, DE/rand-to-best/1 and DE/TSDE, are considered. (2) The values of the scaling factors K and F are not same. K and F values are randomly generated in each iteration. (3) An improved crossover scheme is used. (4) The proposed HMODE variant is tested by applying it to the real-world and constrained problem with multi-objective functions. The proposed HMODE technique's pseudo code is detailed in the Appendix. Figure 5 describes the procedure of implementing HMODE algorithm for a multi-objective optimization problem.

3.2.1. HMODE operators. To fix the crossover constant (CC), an analysis was made by varying CC value from 0.1 to 0.95. The optimal travel time (F_1) in Application 1 was found. The results are given in Table II.

https://doi.org/10.1017/S026357471800156X Published online by Cambridge University Press

Table II. CC vs. F1.

CC	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
F_1	25.4	24.5	23.2	23.1	22.8	22.6	22.3	21.8	21.8
CC	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95
F_1	21.8	21.7	21.7	21.5	21.4	21.4	21.3	21.3	**21.2**

Note: The least value is given in bold.

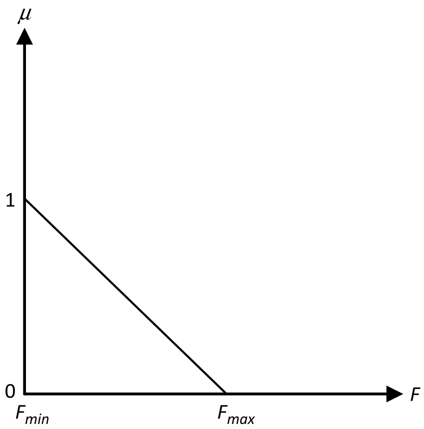

Fig. 6. Fuzzy membership function for a minimization objective function.

From Table II, it is observed that when CC value is 0.95, we got the minimum value for F_1. Therefore, the CC of 0.95, number of population (NP) of 100 and the maximum generations of 100 were found suitable for the HMODE algorithm.

4. Performance Measures for Multi-Objective Optimization

In all multi-criteria optimization processes, there are three important and major goals: (1) to find solutions that are as close to all the Pareto-optimal solutions in the non-dominated front as possible, (2) to discover solutions which are diverse and cover all the regions of the front and (3) to discover the superior behavior of the optimization algorithm. Therefore, to made a comparison between two or more multi-criteria optimization algorithms, at least three performance metrics (first one for testing the progress of the optimization toward a desired Pareto-optimal front, the second one for testing the solution spread of Pareto-front and the third one is to calculate computational ability of optimization algorithm) are required to be utilized. Moreover, the exact usage of these three performance measures is imperative. For the above-stated reasons, RNI, SSM and OO metrics are selected. To select the best solution from the Pareto-optimal front, AFMF method is used. The strength or weakness of a Pareto-optimal front is assessed by RNI and SSM. The computational complexity of multi-objective optimization algorithms is analyzed by the performance measures AE and OO.

4.1. AFMF (μ_{avg}) method

AFMF method is the best choice if a decision maker does not know a suitable value of weight to be assigned to each objective function. Normally, the decision maker may not know the weightage to be assigned to all the objective functions. Moreover, the weightage given to any objective function is not deterministic in nature. They are fuzzy type. To handle this situation, an efficient method such as AFMF is used here for multi-objective performance evaluation.

Figure 6 shows the fuzzy membership function (μ) value of an input (i.e., an objective function value). Fuzzy membership function (μ) depends on weightage value of each input. Fuzzy membership function (μ) value is one at F_{min} and zero at F_{max} for a minimization objective function. Fuzzy membership function (μ) value is zero at F_{min} and one at F_{max} for a maximization objective function.

Fuzzy membership function for a minimization objective function is calculated as below:

$$\mu_i = (F_{imax} - F_i)/(F_{imax} - F_{imin})$$

https://doi.org/10.1017/S026357471800156X Published online by Cambridge University Press

F_i = value of objective function i (i = objective function number = 1, 2 for this problem),
F_{min} = minimum value of objective function i,
F_{max} = maximum value of objective function i.

The best algorithm will have the highest AFMF value (i.e., μ_{avg}).

For this problem, μ_{avg} is calculated as below:

$$\mu_{avg} = (\mu_1 + \mu_2)/2.0 \tag{23}$$

where

$$\mu_{11} = (F_{1max} - F_1)/(F_{1max} - F_{1min}),$$
$$\mu_2 = (F_{2max} - F_2)/(F_{2max} - F_{2min}).$$

4.2. Ratio of non-dominated individuals

The necessary conditions for a better Pareto-optimal front are (1) it needs to have more Pareto-optimal solutions. (2) They need to be evenly spread in all sector of the front. Then only, a decision maker will get more trade-off solutions in all regions or domains. RNI checks the first necessary condition for the best Pareto-optimal front.

RNI is found from the following formula:

$$RNI(X) = nondom\_indiv/P \tag{24}$$

Here, $nondom\_indiv$ = number of non-dominated solutions present in population X, P = population size. If $RNI = 1$, then all the solutions are non-dominated. If $RNI = 0$, then there is no non-dominated solution present in the population. The desired RNI value should be in the range above zero and below one. The desired is that there should be at least one non-dominated solution present. Further, all solutions are to be non-dominated solutions is a rare case. So, an algorithm needs to offer a good RNI range of $0 < RNI < 1$. The best algorithm will have a high RNI value.

4.3. Solution spread measure

The second necessary condition for the best Pareto-optimal front is that Pareto-optimal solutions need to be well spread in all regions of the front. Then only, a decision maker will get more trade-off solutions in all regions or domains. SSM checks the second necessary condition for the best Pareto-optimal front:

$$\mathrm{SSM} = \frac{d_f + d_l + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_l + (N-1)\bar{d}} \tag{25}$$

Here, N = number of solutions in the Pareto frontier. There are $(N-1)$ consecutive distances, d_i = the distance (in objective function space) between each solution, $\bar{d}$ = an arithmetic mean of all d_i. d_l = Euclidean distance between extreme solutions of Pareto-optimal. df = Euclidean distance between the boundary solutions and If SSM value is small, there is an excellent distribution of the solutions in Pareto-optimal frontier. If SSM value is high, then there is no good spread of solutions in Pareto-optimal set. So, SSM finds distribution capacity of an algorithm. The best algorithm will have a low SSM value.

4.4. Optimizer overhead

Two computer-related parameters such as the total number of functions assessments and the total time complexity are used to test the effectiveness of an algorithm. By these parameters, the simplicity, the number of calculations to be made, memory requirements and the time required to solve can be tested for an algorithm. The best algorithm will have a simple structure, less number of calculations, less running time and less memory space requirement. OO is computed as

$$\mathrm{OO} = (T_{Total} - T_{PFP})/T_{PFP} \tag{26}$$

Here, T_{Total} = total time taken for giving the results. T_{PFP} = time spent for the pure function assessments. If the value of OO is zero, then the algorithm is an efficient one. Also, the algorithm is not having any overhead. Practically, this is an ideal case and not practicable.

Table III. WMR geometrical parameters.

Wheel radius (*r*)	Distance between wheels (*r*)	Mass (*m*)	Moment of inertia (*I*)	τ_i^{max}
0.15 cm	0.1 m	5 kg	26 kg.m^2	1 N.m

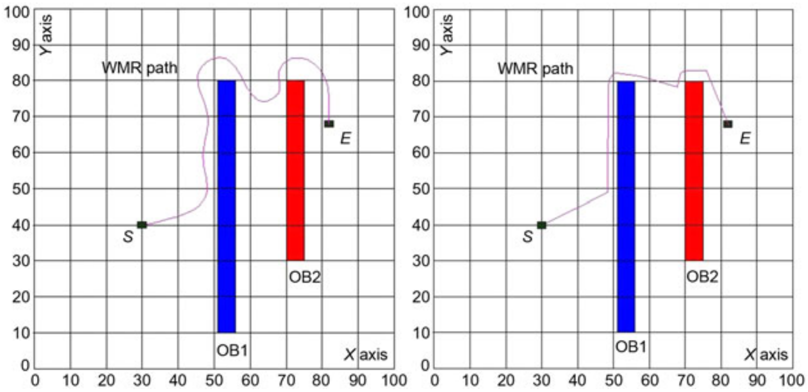

Fig. 7. Optimal path of WMR in Application 1 from NSGA-II and HMODE.

4.5. Algorithm effort

AE calculates the computational efficiency of an algorithm to find non-dominated solutions. In a fixed sample time, how many functions are evaluated by the algorithm is called as AE. AE is calculated as below:

$$\begin{aligned} &\text{AE} = T_{run}/N_{eval}, \\ &(T_{run} > T_{1stgen}) \cap (T_{eval} \propto N_{eval}) \end{aligned} \tag{27}$$

Here, N_{eval} is the total number of functions evolutions. T_{run} is a fixed period of simulation time. If an algorithm is best in terms of computational effort, then it will evaluate a number of functions in a fixed time (T_{run}). So, for a good algorithm which has high computational effort, AE value will be small. The condition to set the fixed time is $T_{run} > T_{1stgen}$. Here, T_{1stgen} is the computation time of first generation. T_{run} and $T_{run} > 0$. Generally, AE value is in the range of $(0, \infty)$.

5. Simulation Results

The mobile robot has two DC motors for driving wheels in the backside. Arduino board and other electronic components are used for control and programming. All the geometrical parameters like robot mass, wheel radius, the moment of inertia, etc., are noted for the custom-developed mobile robot. The geometrical parameters of WMR are detailed in Table III. The kinematic and dynamic models of a differential drive WMR are derived as detailed Sections 2.4 and 2.5. Five different applications of WMR have been simulated. The optimal path which has the highest AFMF (μ_{avg}) is selected as the best optimal path.

In Application 1, the starting point of the WMR is (30, 40) cm and goal point is (83, 67) cm. First obstacle size is 5 cm × 70 cm. Second obstacle size is 5 cm × 50 cm. The best optimal path of WMR obtained from NSGA-II and HMODE is given in Fig. 7.

In Application 2, the WMR start point is (14, 8) m and goal point is (14, 18) m. WMR moves inside a room. So the room has four walls and two fixed obstacles as shown in Fig. 8. The best optimal path of WMR obtained from NSGA-II and HMODE is given in Fig. 8.

In Application 3, the WMR start point is (5, 5) m and goal point is (30, 25) m. WMR has nine fixed obstacles to its path as shown in Fig. 9. The best optimal path of WMR obtained from NSGA-II and HMODE is given in Fig. 9.

In Application 4, the WMR start point is (200, 0) cm and goal point is (0, 250) cm. WMR has three moving obstacles to its path as per Fig. 10. Obstacle 1 is a circle with diameter of 20 cm. Its initial point is (−50, 250) cm. It moves with a speed of 10 cm/s from left to right as in Fig. 10. Obstacle 2 is a square with 10-cm side length. Its initial point is (100, 250) cm. It moves with a speed of 3 cm/s

https://doi.org/10.1017/S026357471800156X Published online by Cambridge University Press

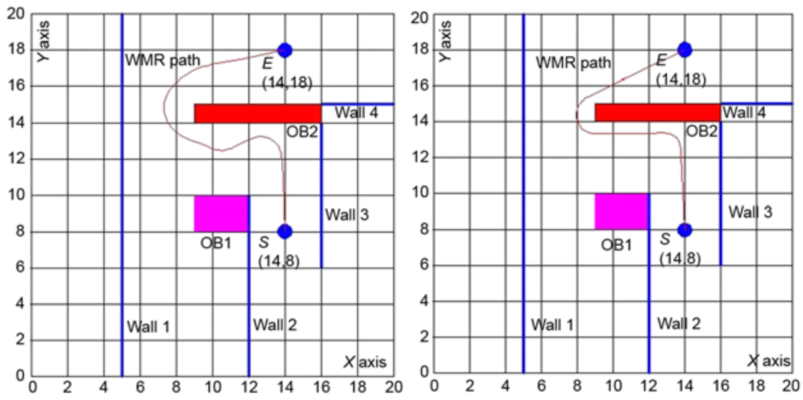

Fig. 8. Optimal path of WMR in Application 2 from NSGA-II and HMODE.

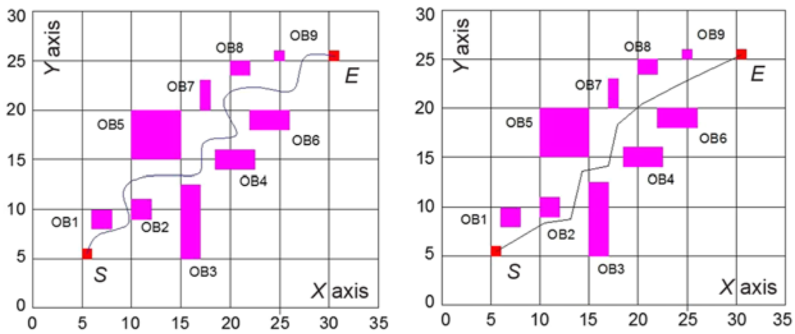

Fig. 9. Optimal path of WMR in Application 3 from NSGA-II and HMODE.

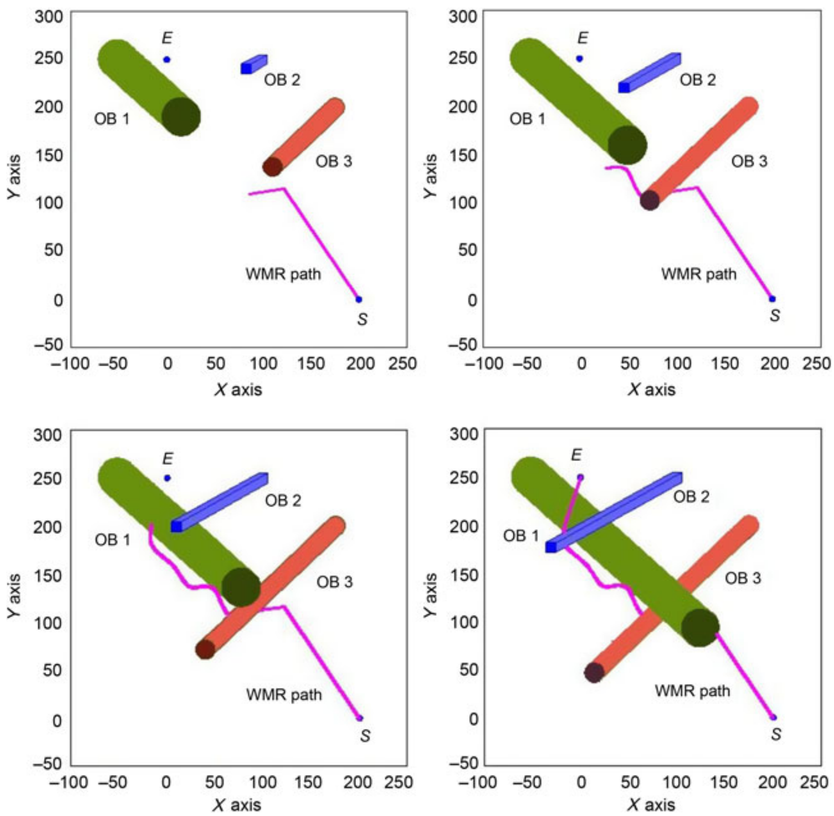

Fig. 10. Optimal path of WMR in Application 4 from HMODE.

https://doi.org/10.1017/S026357471800156X Published online by Cambridge University Press

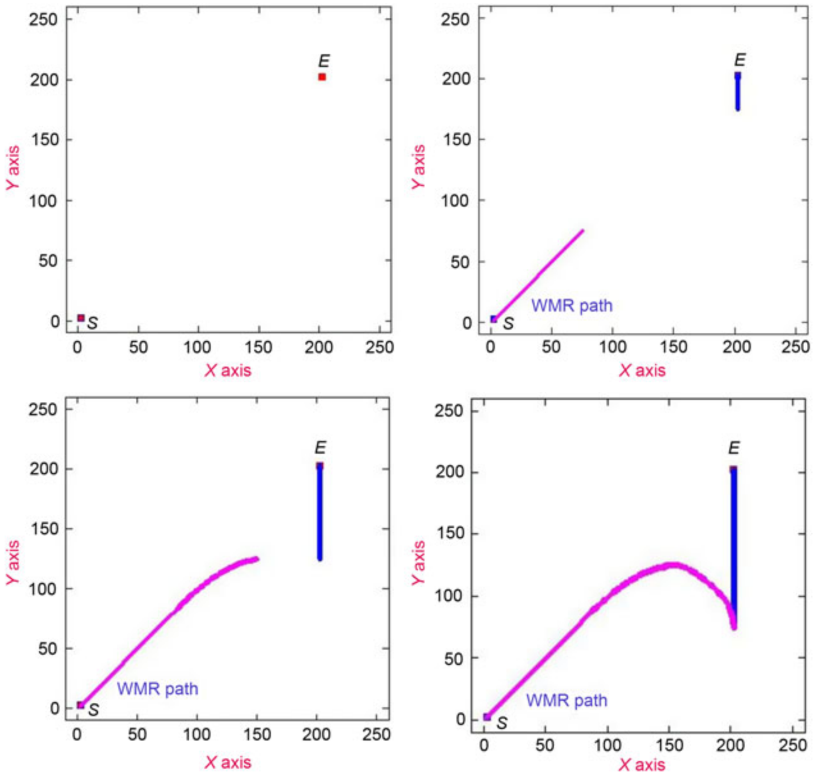

Fig. 11. Optimal path of WMR in Application 5 from HMODE.

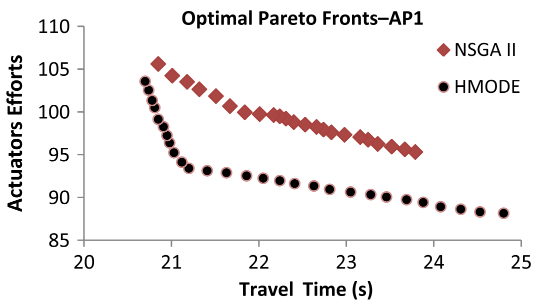

Fig. 12. Pareto optimal fronts of NSGA-II and HMODE for Application 1.

as in Fig. 10. Obstacle 3 is a circle of 10-cm diameter. Its initial point is (175, 200) cm. It moves with a speed of 10 cm/s from right to left as in Fig. 10. The best optimal path of WMR obtained from HMODE is given in Fig. 10.

In Application 5, the WMR start point is (0, 0) cm and goal point is (200, 200) cm. But the goal point moves with a speed of 3.5 cm/s as in Fig. 11. The best optimal path of WMR obtained from HMODE is given in Fig. 11.

Numerical experiments have been performed using evolutionary algorithms like NSGA-II and HMODE. Then the resultant optimal motions are verified through simulation. The results such as travelling time and actuator efforts are noted. Pareto-optimal frontiers and the optimal solutions obtained from the NSGA-II and HMODE algorithms are analyzed and compared.

The values of the multi-objective performance indicators such as RNI, SSM, AE and OO for NSGA-II as well as HMODE approaches are calculated. A comparison of results from both algorithms is made. Results proved that NSGA-II and HMODE approaches are good for creating optimal motion planning of differential drive WMR. The Pareto-optimal frontiers achieved from NSGA-II and HMODE are given in Figs. 12–16. It is observed from Figs. 12–16 that HMODE offers best Pareto-optimal fronts than those of NSGA-II. Tables IV– VIII illustrate the optimum value of objective functions obtained from NSGA-II and HMODE approaches.

https://doi.org/10.1017/S026357471800156X Published online by Cambridge University Press

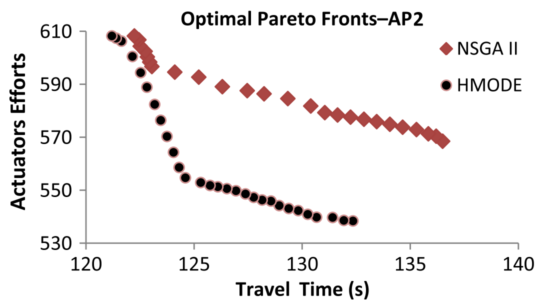

Fig. 13. Pareto optimal fronts of NSGA-II and HMODE for Application 2.

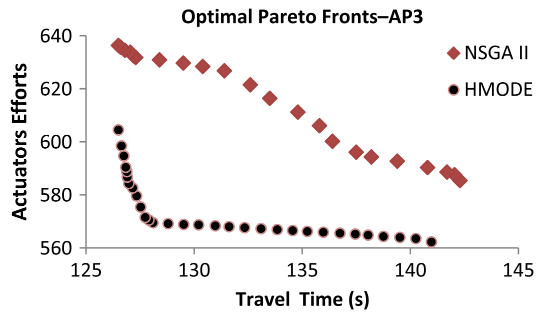

Fig. 14. Pareto optimal fronts of NSGA-II and HMODE for Application 3.

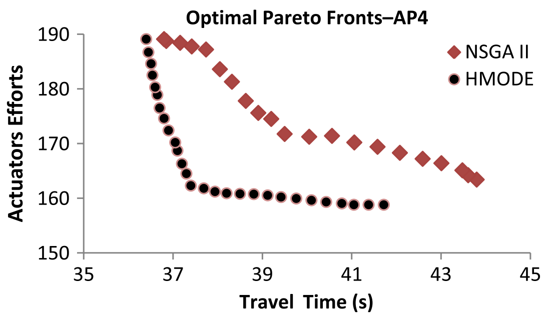

Fig. 15. Pareto optimal fronts of NSGA-II and HMODE for Application 4.

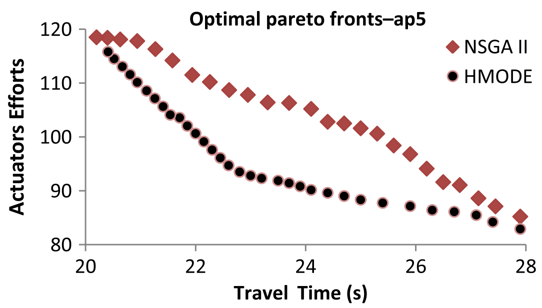

Fig. 16. Pareto optimal fronts of NSGA-II and HMODE for Application 5.

The approach which provides the highest μ_{avg}, minimum SSM, maximum RNI, minimum AE and minimum OO values is considered as the most appropriate approach for solving the multi-objective optimization problem. The solutions obtained from NSGA-II and HMODE are given in Table IX. It is observed from Table IX that (1) HMODE algorithm provides the maximum RNI value as compared to NSGA-II approach. (2) Also, it gives minimum SSM value than that of NSGA-II. (3) Also, HMODE technique is better than NSGA-II with regard to the highest μ_{avg}, minimum AE and

Table IV. Optimal results obtained from NSGA-II and HMODE algorithms for Application 1.

	F_1 (s)	F_2 (J)	F_{avg}
F_{max}	24.8	105.6	
F_{min}	20.7	88.15	
NSGA-II	22.4	98.8	0.4917
HMODE	21.2	93.4	0.79575

Table V. Optimal results obtained from NSGA-II and HMODE algorithms for Application 2.

	F_1 (s)	F_2 (J)	F_{avg}
F_{max}	136.5	608.25	
F_{min}	121.2	538.4	
NSGA-II	130.4	581.8	0.3887
HMODE	124.6	554.7	0.77225

Table VI. Optimal results obtained from NSGA-II and HMODE algorithms for Application 3.

	F_1 (s)	F_2 (J)	F_{avg}
F_{max}	142.3	636.35	
F_{min}	126.5	562.25	
NSGA-II	135.8	606.1	0.4098
HMODE	128.1	569.5	0.90045

Table VII. Optimal results obtained from NSGA-II and HMODE algorithms for Application 4.

	F_1 (s)	F_2 (J)	F_{avg}
F_{max}	43.8	189.1	
F_{min}	36.4	158.8	
NSGA-II	39.5	171.75	0.57685
HMODE	37.4	162.3	0.8747

Table VIII. Optimal results obtained from NSGA-II and HMODE algorithms for Application 5.

	F_1 (s)	F_2 (J)	F_{avg}
F_{max}	27.9	118.5	
F_{min}	20.2	82.9	
NSGA-II	24.4	102.8	0.44775
HMODE	22.6	94.7	0.6784

minimum OO values. (4) It is noted that HMODE approach finds better convergence than NSGA-II. (5) Likewise, the computational complexity of finding Pareto-optimal frontier in HMODE is less than that of NSGA-II. Subsequently, HMODE algorithm is significantly faster as compared to NSGA-II algorithm.

From Figs. 12–16, it is found that HMODE delivers large number of non-dominated solutions than NSGA-II. Accordingly, HMODE is the most suitable approach for the problem where the users need large number of solutions to their choice. The results gained from NSGA-II and HMODE are compared with the results obtained by methods proposed by Patle et al.,[26] namely fuzzy logic (FL), multi-objective genetic algorithm (MGA), ant colony optimization (ACO) and artificial neural networks (ANN) listed in Table X. From Table X, the points observed are: (1) Both NSGA-II and

Table IX. The performance metrics values of NSGA-II and HMODE algorithms.

Algorithm		T_{run} (s)	N_{eval}	AE	SSM	RNI	OO
NSGA-II	Application 1	2	105	0.019	0.815	0.23	0.165
	Application 2	2	108	0.0185	0.824	0.26	0.176
	Application 3	2	112	0.0178	0.861	0.22	0.181
	Application 4	2	117	0.017	0.872	0.21	0.182
	Application 5	2	122	0.0164	0.804	0.25	0.179
HMODE	Application 1	2	135	0.0148	0.973	0.28	0.0653
	Application 2	2	141	0.0142	0.981	0.29	0.0671
	Application 3	2	147	0.0136	0.984	0.31	0.0712
	Application 4	2	154	0.0129	0.986	0.28	0.0725
	Application 5	2	161	0.0124	0.978	0.33	0.0721

Table X. Optimal travel time (T in seconds) comparison with Patle et al.26 methods.

	FI[26]	ANN[26]	ACO[26]	MGA[26]	NSGA-II	HMODE
Application 1	25.96	25.17	24.4	23.62	22.4	21.2
Application 5	26.95	28.95	29.45	25.9	24.4	22.6

HMODE provide better performance than the methods proposed by Patle et al.[26] (2) HMODE gives superior solutions than NSGA-II. Hence, HMODE is superior than NSGA-II.

From the results, the following points are observed: (a) HMODE algorithm gives the best optimal fronts for user choice. The best optimal solution can be selected by using AFMF method. (b) The strength of optimal fronts is good. (c) The computational effort of HMODE is good. (d) Numerical experiments proved that NSGA-II and MODE are good algorithms for optimal motion designing of a differential-driven WMR than the methods proposed by Patle et al.[26] and (e) Both NSGA-II and HMODE do best obstacle avoidance.

The reasons behind the superior performance of HMODE algorithm are: (1) HMODE is an evolutionary algorithm with floating-point-encoded scheme. (2) The theoretical framework of the HMODE is very simple. So, it needs less computational work, less storage space and less CPU run time. HMODE has a simple structure, speed, robustness and ease of use. (3) Best-performing variants, namely DE/current-to-best/1, DE/rand-to-best/1 and DE/TSDE are included in HMODE. Furthermore, HMODE has two more variations, viz. DE/rand/1 and DE/rand/2. (4) In HMODE algorithm, the values of scaling factors K and F are not same. They are randomly generated in each iteration. (5) An improved crossover scheme is used. (6) HMODE provides better convergence speed, computational complexity, accuracy (near the global optimal solution), stability, success rate, final solution quality and robustness.

6. Conclusions

The proposed method finds multi-objective optimal motion planning using two intelligent optimization techniques namely HMODE and NSGA-II for a custom-made WMR with a differential drive. The conclusions made from this research work are: (1) The numerical results of the experiments proved that both NSGA-II and HMODE are good algorithms for optimal trajectory planning for differential-driven WMR. (2) All essential constraints like geometric, dynamic and kinematic restraints of the WMR are satisfied with all trade-off solutions obtained from HMODE and NSGA-II. (3) Both HMODE and NSGA-II give best moves for the autonomous mobile robot in terms of obstacle avoidance. (4) HMODE technique gives a minimum value for AE and OO, and maximum value for average fuzzy membership function than those of NSGA-II. (5) Also, HMODE technique gives more trade-off solutions, a minimum value for SSM and maximum value for RNI than those of NSGA-II. (6) The algorithm running time for HMODE is less than that of NSGA-II. (7) HMODE algorithm gives the best Pareto-optimal fronts than NSGA-II algorithm. (8) Both NSGA-II and HMODE provide better performance than the Patle et al.[26] methods, namely FL, MGA, ANN and ACO. (9) HMODE gives improved results than NSGA-II for optimization problem with multi-objective. (10) The proposed HMODE algorithm is a more suitable method for multi-objective optimal trajectory planning of WMR.

The future works are real-time experiments in indoor, outdoor environments. Another future work is the development of an intelligent WMR robot for all terrain conditions.

References

1. S. M. Ahmadi, H. Kebriaei and H. Moradi, "Constrained coverage path planning: Evolutionary and classical approaches," *Robotica* **36**(6), 904–924 (2018).
2. S. S. Paliwal and R. Kala, "Maximum clearance rapid motion planning algorithm," *Robotica* **36**(6), 882–903 (2018).
3. T. Kröger and F. M. Wahl, "Online trajectory generation: Basic concepts for instantaneous reactions to unforeseen events," *IEEE Trans. Rob*. **26**(1), 94–111 (2010).
4. F. Canali, C. Guarino Lo Bianco and M. Locatelli, "Minimum jerk online planning by a mathematical programming approach," *Eng. Optim*. **46**(6), 763–783 (2014).
5. Y. Kim and B. K. Kim, "Time optimal trajectory planning based on dynamics for differential-wheeled mobile robots with a geometric corridor," *IEEE Trans. Ind. Electron*. **64**(7), 5502–5512 (2017).
6. R. Walambe, N. Agarwal, S. Kale and V. Joshi, "Optimal trajectory generation for car-type mobile robot using spline interpolation," *IFAC-Papers On Line* **49**(1), 601–606 (2016).
7. C. H. Kim and B. K. Kim, "Minimum-energy translational trajectory generation for differential-driven wheeled mobile robots," *J. Intell. Rob. Syst*. **49**, 367–383 (2007).
8. A. Sgorbissa and R. Zaccaria, "Planning and obstacle avoidance in mobile robotics," *Rob. Auton. Syst*. **60**, 628–638 (2012).
9. B. Li and Z. Shao, "Simultaneous dynamic optimization: A trajectory planning method for nonholonomic car-like robots," *Adv. Eng. Software* **87**, 30–42 (2015).
10. S. Liu and D. Sun, "Minimizing energy consumption of wheeled mobile robots via optimal motion planning," *IEEE ASME Trans. Mech*. **19**(2), 401–411 (2014).
11. S. Han, B. S. Choi and J. M. Lee, "A precise curved motion planning for a differential driving mobile robot," *Mechatronics* **18**, 486–494 (2008).
12. M. H. Korayem, M. Nazemizadeh and V. Azimirad, "Optimal trajectory planning of wheeled mobile manipulators in cluttered environments using potential functions," *Sci. Iran B* **18**(5), 1138–1147 (2011).
13. M. Boryga, A. Graboś, P. Kołodziej, K. Gołacki and Z. Stropek, "Trajectory planning with obstacles on the example of tomato harvest," *Agric. Agric. Sci. Procedia* **7**, 27–34 (2015).
14. M. H. Korayem, A. M. Shafei and H. R. Shafei, "Dynamic modeling of nonholonomic wheeled mobile manipulators with elastic joints using recursive Gibbs–Appell formulation," *Sci. Iran B* **19**(4), 1092–1104 (2012).
15. M. H. Korayem, H. N. Rahimi and A. Nikoobin, "Mathematical modeling and trajectory planning of mobile manipulators with flexible links and joints," *Appl. Math. Model*. **36**, 3229–3244 (2012).
16. H. Mirzaeinejad and A. M. Shafei, "Modeling and trajectory tracking control of a two-wheeled mobile robot: Gibbs–Appell and prediction-based approaches," *Robotica* **36**(10), 1551–1570 (2018).
17. B. Wang, S. Li, J. Guo and Q. Chen, "Car-like mobile robot path planning in rough terrain using multi-objective particle swarm optimization algorithm," *Neurocomputing* **282**(22), 42–51 (2018).
18. E. Plaku, E. Plaku and P. Simari, "Clearance-driven motion planning for mobile robots with differential constraints," *Robotica* **36**(7), 971–993 (2018).
19. R. Saravanan, S. Ramabalan and C. Balamurugan, "Multiobjective trajectory planner for industrial robots with payload constraints," *Robotica* **26**(6), 753–765 (2008).
20. Y. Xue and J.-Q. Sun, "Solving the path planning problem in mobile robotics with the multi-objective evolutionary algorithm," *Appl. Sci*. **8**(9), 1425 (2018).
21. S. G. Radhakrishna Prabhu, R. C. Seals, P. J. Kyberd and J. C. Wetherall, "A survey on evolutionary-aided design in robotics," *Robotica* **36**(12), 1804–1821 (2018).
22. M. E. A. Boudjellel and T. Chettibi, "Optimal Trajectory Planning for a Mobile Robot in Presence of Obstacles Using Multi-objective Optimization Techniques," *8th International Conference on Modelling, Identification and Control (ICMIC-2016)*, Algiers, Algeria (2016), pp. 509–514.
23. K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput*. **6**(2), 182–197 (2002).
24. B. V. Babu and B. Anbarasu, Multi-Objective Differential Evolution (MODE): An Evolutionary Algorithm for Multi-Objective Optimization Problems (MOOPs), http://discovery.bitspilani.ac.in/discipline/chemical/BVb/publications/html, 2005.
25. S. Thangavelu and C. Shunmuga Velayutham, "An investigation on mixing heterogeneous differential evolution variants in a distributed framework," *Int. J. Bio. Inspir. Comput*. **7**(5), 307–320 (2015).
26. B. K. Patle, D. R. K. Parhi, A. Jagadeesh and Sunil Kumar Kashyap, "Matrix-binary codes based genetic algorithm for path planning of mobile robot," *Comput. Electr. Eng*. **67**, 708–728 (2018).

Appendix

HMODE-Pseudo code:

The steps are repeated up to all the selected best NP individuals have the rank number one.

The goal is to minimize all objective functions, f_q.

(1) For all N_p parent vectors generate box, P by a random-number algorithm, where P is a sequence of real variables.
(2) Vectors classification procedure by non-domination:
 a. Generate new (empty) box, P', with size, N_p.
 b. For $(i = 1 - N)$
 {Move ith vector from P to P',}
 c. Compare P with P' for each member
 d. If i, member of P dominates over j, member of P' in terms of all objective functions, remove the jth vector from P' and put it back in its original location in P and vice versa
 e. Maintain both i and j in P' (in sequence), if i and j are non-dominating
 f. Repeat the above procedure for all the vectors in P. Now P' contains a sub-box called first front. It is a subset of P contains non-dominated vectors. Assign a rank number, I_{rank}, of I
 g. Repeat Step 2b above (for the vectors remain in P), create subsequent fronts. Assign ranks for the subsequent fronts generated, $I_{rank} = 2, 3, \ldots$ Thus, the box P' is a sequence of fronts generated by all N_p vectors.
(3) Spreading out: procedure to calculate crowding distance, $I_{i,dist}$. It is a distance from ith vector which is from any front, j, of P':
 a. Arrange all vectors in front j sequentially to define the nearest neighbor of any vector in the front. The vectors are arranged in a ascending fashion based on objective functions like fitness functions.
 b. Find the cuboid of largest size which encloses vector i that nearly touches its very nearest neighbors in f-space.
 c. Calculate $I_{i,dist} = 1/2 \times$ (sum of all the sides of the cuboid)
 d. The values of largest I_i, dist are solutions at boundaries. It influences the convergence characteristics.
(4) Do the DE operation on NP target vectors present in the P' to generate a NP trial vectors, then save it in P''.
 a. Generate new (empty) box, P'' of size N_p
 b. For $(i = 1 - N)$, in P', allocate solutions to sub populations NP/5, where each node has NP/5 solutions
 c. Perform Mutation:

Assign a DE variant to all nodes. Create a noisy vector $(N_{i,G})$ in each node by using the below mentioned scheme:

Node 1: DE/rand/1: $N_{i,G} = X_{r1,G} + F(X_{r2,G} - X_{r3,G})$.
Node 2: DE/rand/2: $N_{i,G} = X_{r1,G} + K(X_{r2,G} - X_{r3,G}) + F(X_{r4,G} - X_{r5,G})$.
Node 3: DE/current-to-best/1: $N_{i,G} = X_{i,G} + K(X_{best,G} - X_{r1,G}) + F(X_{r2,G} - X_{r3,G})$.
Node 4: DE/rand-to-best/1: $N_{i,G} = X_{r1,G} + K(X_{best,G} - X_{i,G}) + F(X_{r2,G} - X_{r3,G})$.
Node 5: DE/TSDE: $N_{i,G} = X_{r1,G} + K(X_{best1,G} - X_{r2,G}) + F(X_{best2,G} - X_{r3,G})$.

Here, $X_{i,G}$ is the current solution of a vector, $X_{r1,G}, X_{r2,G}, X_{r3,G}, X_{r4,G}, X_{r5,G}$ are randomly chosen solutions, $X_{best1,G}, X_{best2,G}, X_{best,G}$ are the best solutions in the vector, K and F are scaling factors = rand[0.3, 0.9] = a random number in between 0.3 and 0.9 generated in every iteration.

If $N_{i,G} > X_{max}$ or $N_{i,G} < X_{min}$ then once again create a new $N_{i,G}$ by using above mutation schemes.

Combine all new nodes in the box P''.

 d. Calculate P'' = {trial vector}, where trial vector = {target vector (cross over) noisy random vector}′.
 e. Crossover procedure:
 (1) Generate random numbers = problem dimension;
 (2) For each of the dimensions: if random no. > CR;

https://doi.org/10.1017/S026357471800156X Published online by Cambridge University Press

Assign, trial vector value = the target vector value.
Else, trial vector value = the noisy random vector value.

(5) Elitism: Copy $\{N_p$ parent vectors$\}$ (P')^$\{N_p$ trial vectors$\}$ $(P'') \Rightarrow$ box PT. Now Box $PT = \{2N_p \text{vectors}\}$

a. Apply classification procedure mentioned in Step 2 above, to PT to generate $2N_p$ vectors inside the fronts (box PT') using non-domination principle only.

b. Box $P''' =$ best N_p from box PT'. The best N_p is identified as the one which has the best ranked chromosome.

If $I_{i,rank} \neq I_{j,rank}$: $I_{i,rank} < I_{j,rank}$;
If $I_{i,rank} = I_{j,rank}$: $I_{i,dist} > I_{j,dist}$.

Now one generation is completed. Stop if the appropriate criteria are satisfied, for example, the generation number > maximum no. of generations. Else, copy P''' into the starting box P. And Go to the Step 2 above.

https://doi.org/10.1017/S026357471800156X Published online by Cambridge University Press