

Article

Multi-Objective Combinatorial Optimization Using the Cell Mapping Algorithm for Mobile Robots Trajectory Planning

Efraín Grisales-Ramírez ^{1,*}  and Gustavo Osorio ² 

¹ Grupo de Automática, Departamento de Electrónica y Automatización, Universidad Autónoma de Manizales, Manizales 170002, Colombia

² Grupo de Percepción y Control Inteligente, Departamento de Ingeniería Eléctrica, Electrónica y Computación, Facultad de Ingeniería y Arquitectura, Universidad Nacional de Colombia–Sede Manizales, Manizales 170003, Colombia

* Correspondence: efrain.grisalesr@autonoma.edu.co

Abstract: The use of optimal control theory for motion planning is a challenging task. Cell mapping offers a way to formulate combinatorial optimization problems, allowing the inclusion of complex cost functions as well as multi-objective optimization problems. This paper presents a suboptimal solution for a trajectory planning problem in a workspace with obstacles, for a differential drive mobile robot. This method relies on the use of any linearization technique that allows the regularization of the combinatorial optimization problem. We explore some classical problems in optimal control, i.e., distance, control effort, and navigation time), as well as the multi-objective optimization problem (MOP). We also performed a comparison with two classical path planning algorithms, namely A^* and RRT^* , to validate the proposed method when the multi-objective optimization problem includes distance in the cost function, achieving a compromise of less than 2% for the worst-case scenario for our case study.

Keywords: cell mapping algorithm (CMA); combinatorial optimization; mobile robot; multi-objective optimization; trajectory planning



Citation: Grisales-Ramírez, E.; Osorio, G. Multi-Objective Combinatorial Optimization Using the Cell Mapping Algorithm for Mobile Robots Trajectory Planning. *Electronics* **2023**, *12*, 2105. <https://doi.org/10.3390/electronics12092105>

Academic Editors: Ionica Oncioiu, Stelian Brad and Fuji Ren

Received: 27 February 2023

Revised: 26 March 2023

Accepted: 2 April 2023

Published: 4 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Using optimal control theory for trajectory planning presents two main challenges. Firstly, formulating and solving the problem analytically can be difficult. Secondly, computational methods can be cumbersome, and in some cases, interpreting the solution can be obscure. Cell-mapping offers a way to pose combinatorial optimization problems that lead to suboptimal solutions regarding the continuous original problem. Some advantages of this approach include the ability to formulate complex cost functions and the natural evolution towards multi-objective optimization problems. Trajectory optimization algorithms for mobile robots, aim to minimize a cost function while considering system dynamics, boundary conditions, and additional constraints [1–4]. In most cases, the trajectory planning problem can be viewed as the open loop solution of an optimal control problem, where the cost function may include several criteria such as distance, time, control effort, energy, safety, or their combination. The boundary conditions represent the initial and final states, while the constraints may relate to the physical implementation (speed, acceleration, and jerk), the workspace (obstacles and boundaries), and the nature of the task (autonomy, regulations, and comfort) [5–7].

In the motion control stage, the trajectory planned earlier is used as a reference. This requires navigation systems capable of locating the mobile robot in relative or absolute coordinates over time. Ultimately, the control system is designed to allow the robot to follow the reference trajectory provided by the planner [8–12]. The literature on trajectory planners highlights several open problems, such as generalizing methods for known and partially

known or unknown environments; lack of adaptation and non-robust behavior of conventional techniques [13]; high computational costs [14,15]; long routes [16,17]; trajectory generation without considering vehicle kinematic and dynamic constraints [18–22]; and complex environments [23]. In [24], a comparative analysis of intelligent and conventional methods for path planning is presented.

Optimization problems in trajectory planning can be organized into continuous and combinatorial categories, based on the nature of the mathematical objects and numerical methods [25]. In the literature, there are different algorithms that solve the optimization problem for path planning, including Dijkstra's algorithm, *RRT**, and *A**. The first is an exhaustive method that finds the shortest path in a discretized space by searching for the shortest path in a graph. The second is a modification of the RRT algorithm, which is based on the random expansion of trees in continuous configuration spaces. As the tree is constructed, a cost function that considers distance is implemented, and when the algorithm finds the path that minimizes this function, it stops [26]. The third is a heuristic search method that finds the shortest route between two nodes in a discretized space. The quality of the found path depends on the selected heuristic function [27]. In this work, we propose a direct connection between combinatorial and cell mapping methods.

Cell mapping was proposed by C. S. Hsu in the 1980s [28], as an attempt to find efficient and practical ways to determine the global behavior of strongly nonlinear systems. The basic idea behind it, is to consider the state space as a collection of a large number of cells, where each cell is taken as a state entity. So far, two types of cell mappings have been investigated: simple and generalized. In simple cell mapping [29], each cell has a single image cell, whereas in generalized mapping [30,31], a cell has many image cells, with a given probability distribution between them. Each cell mapping has its own mathematical structure. For the case studies in this paper, simple cell mapping (SCM) is implemented, since there is a single image cell for the dynamical system, given a specific input with given initial conditions. Cell mapping methods are currently of interest to the scientific community due to recent developments in high-performance computing systems and the proliferation of large-scale data. In [32], a mathematical description of the cell mapping method and various engineering applications are presented. As outlined in the book, the cell mapping algorithm (CMA) is currently applied to solve nonlinear algebraic equations, multi-objective optimization problems [33,34], optimal controller design, global analysis of nonlinear dynamics, and other applications, such as the search for solutions in optimization problems, as shown in [35], and nonlinear controller design [36].

The use of optimal control theory for trajectory planning presents challenges in terms of formulating and solving problems analytically, as well as with computational methods and interpretation of solutions. However, cell-mapping offers a way to pose combinatorial optimization problems that lead to suboptimal solutions regarding the continuous original problem. This approach allows for complex cost function formulation and evolution towards multi-objective optimization problems. Trajectory optimization algorithms for mobile robots seek to minimize a cost function while considering system dynamics, boundary conditions, and additional constraints. The literature on trajectory planners highlights several open problems, including lack of adaptation and non-robust behavior of conventional techniques, high computational costs, and trajectory generation without considering vehicle kinematic and dynamic constraints, among others. In this paper, the authors propose a trajectory planner that uses a cell mapping algorithm to solve a multi-objective optimization problem with minimum control effort, distance, and navigation time, providing an alternative solution to these challenges. The proposed method is detailed in Section 2, with results presented in Section 3, and a comparison in distance of the paths generated by four different methods: combinatorial optimization using CMA for minimum distance, the multi-objective combinatorial optimization problem using CMA, the shortest path search algorithm *A**, and the optimized version of the rapidly exploring random tree *RRT** sampling method. Finally, Section 5 presents some conclusions and future work.

2. Materials and Methods

2.1. Optimal Trajectory Planning Problem

The optimal trajectory planning problem for mobile robots can be defined as:

$$\begin{cases} \min J \\ \text{subject to: } \dot{x} = f(x, u, t), \quad x(t_0) = x^0, \quad x(t_1) = x^1, \quad u(t) \in U \end{cases} \quad (1)$$

in this equation, $u(t)$ is a control within an action space U , that is required to take the mobile robot from the initial state x^0 to a target state x^1 . x^1 belongs to a subset of target states $X_G \subset X$. f represents the state transition equation, and J is the cost function that needs to be minimized [24]. Obtaining an analytical solution for the problem in Equation (1) is often difficult, so it is convenient to use a suitable discretization technique to obtain suboptimal solutions using computational algorithms. The following subsection presents a numerical approximation to the optimal trajectory planning problem.

2.2. Cell Mapping Algorithm for Trajectory Planning

In this section, the CMA is adapted, to discretize the state space of the trajectory planning problem. For this method, it is convenient to start with a linear dynamic model, which can be obtained using analytical or machine learning methods. To employ the CMA in the construction of reference paths for robot motion, we propose the following steps:

1. Generate an equally spaced grid for the state space, i.e., position and velocity. Note that the position and velocity constraints of the mobile robot define the boundaries of the grid. At this point, it is also possible to include obstacles or forbidden cells in the grid.
2. Define a set of constant values that can be applied to the dynamic model of the mobile robot. These elements represent constant control actions when applied for the cell mapping method.
3. Decide on the apply time of the control actions that best aligns with the construction of the grid.
4. Build and store the cell mapping weighted graph from a cost function, by applying every control action for each cell in the grid.

In the proposed CMA algorithm, all possible images of a cell are linked in the graph, and in this sense, they are considered neighbors of a cell. If the destination cell is outside the initial state space grid, or is a forbidden cell, then this image is automatically discarded. The cell-to-cell mapping, or simply the cell mapping problem, can be defined mathematically by Equation (2)

$$x(k+1) = C(x(k), \mu), \quad (2)$$

where $x(k+1)$ is the image of $x(k)$, C represents the cell mapping which embeds the dynamical model of the system, and the parameter vector μ is applied. The cell mapping may depend on the step k , but in this work, we consider a stationary dynamical system where C does not explicitly depend on k , as is done in [28].

The evolution of the system given in Equation (2), is applied for each cell in the state space, using different control signals. Upon completion of the evolution, data structures are generated with distance information between node $x(k)$ and $x(k+1)$, as well as the time spent in the evolution and the control signal applied. Any other performance index, such as control effort, energy, etc., can be naturally included in the computation, in order to approximate the solution for different optimization problems.

Algorithm 1 details how CMA is used to construct the graphs of the entire state space. Here, $cell(x^0)$ corresponds to the cell containing x^0 . To avoid dimensional growth in the graph links, it is advisable to track parameters such as cell size, control actions, and time evolution of the dynamic model.

Algorithm 1: CMA for mobile robots in \mathbb{R}^2 .

Result: G_t, G_u, G_d
 $i_1^0 = cell(x^0);$
 $i_1^1 = cell(x^1);$
 $i_1 = i_1^0;$
 $t_s = 1;$
while $i_1 \leq n_{cellsx}$ **do**
 Compute and update links weights;
 $G_t(i_1^0, i_1^1) \leftarrow t_s;$
 $G_t(i_1^0, i_1^1) \leftarrow \frac{1}{2}U^2;$
 $G_t(i_1^0, i_1^1) \leftarrow d(p(i_1^0, i_1^1));$
 $i_1 = i_1^1;$
end

2.3. Combinatorial Optimization Problem Using CMA

The optimization problem is proposed in Equation (3), in which one seeks to minimize a cost function F , that may be subject to different parameters.

$$\begin{cases} \min & F, \\ \text{subject to:} & \bar{q}(k+1) = C(\bar{q}(k), U_i), \\ & \bar{q}(0) = [p(0), \dot{p}(0)]^T, \quad \bar{q}(n-1) = [p(n-1), \dot{p}(n-1)]^T, \\ & U_i = [u_x, u_y], \quad u_x, u_y = \{-\bar{u}, 0, \bar{u}\}, \end{cases} \quad (3)$$

where \bar{q} is the state vector consisting of position (p) and velocity (\dot{p}). $p(i)$ represents a path node with coordinates $(x(k), y(k))$, where $p(0)$ is the origin position of the robot with initial velocity $\dot{p}(0)$, and $p(n-1)$ is the coordinate of the destination with final velocity $\dot{p}(n-1)$. U_i is the vector of control signals applied at each node, with a length that depends on the motion dimensions, which in this case is two.

Now we implement the cost function, with the parameters to be optimized in the robot movement. We build a new data structure in the form of a directed graph with positive values, where the nodes are the cells of the state space and the edges represent the costs associated with displacement between two cells. To determine the path that minimizes the cost function, we perform a shortest path search in the graph, using Dijkstra’s algorithm. The following objective functions are used to solve the optimization problem in mobile robot path planning, and the results are shown in Section 3.

Shortest distance

$$F = F_d = d(p(0), p(n-1)) = \sum_{k=0}^{n-2} \int_{p_k}^{p_{k+1}} \sqrt{1 + [f(p)]^2} dp \quad (4)$$

Least control effort

$$F = F_u = \frac{1}{2} \sum_{k=0}^{n-2} U_k^2 \quad (5)$$

Least navigation time

$$F = F_t = \sum_{k=0}^{n-2} t_s \quad (6)$$

Multi-objective problem

$$F = F_{MOP} = \alpha_1 F_1 + \alpha_2 F_2 + \dots + \alpha_n F_n \quad (7)$$

where, $\alpha_1 + \alpha_2 + \dots + \alpha_n = 1, \quad \alpha_i > 0$

Despite the constraints on the control signals, the optimization problem has multiple solutions in the state space, that correspond to symmetries in the workspace or variations in speed at the intermediate nodes. The robot would take $t = (n - 1)t_s$ seconds to complete each of the possible paths. Formulating a multi-objective optimization problem (MOP) reduces the multiplicity of solutions, depending on the characteristics of the cost function and the problem dimension.

Algorithm 2 details how to solve the combinatorial optimization problem in a graph.

Algorithm 2: Shortest path algorithm in \mathbb{R}^2 .

Result: $shortestpath[]$
 $F \rightarrow \text{Convex hull}(G_t, G_u, G_d)$;
 $k^0 = cell(\bar{q}(0))$;
 $k^{n-1} = cell(\bar{q}(n - 1))$;
 $shortestpath = \min F(d(k^0, k^{n-1}))$;

3. Results

This paper addresses the problem of optimal path planning for differential drive mobile robots. To achieve this, the authors employ the mathematical model of the dynamics of the control point P_r of the robot, as developed in [37]. The mobile robot is depicted in Figure 1, which provides details of the various physical parameters of the platform. The dynamic model is presented in Equation (8), where the state vector q , comprises the linear positions and velocities of the robot, as well as the angular positions of the robot wheels

$$M(q)\ddot{q} + V(q, \dot{q}) = E(q)\tau - A^T(q)\lambda \tag{8}$$

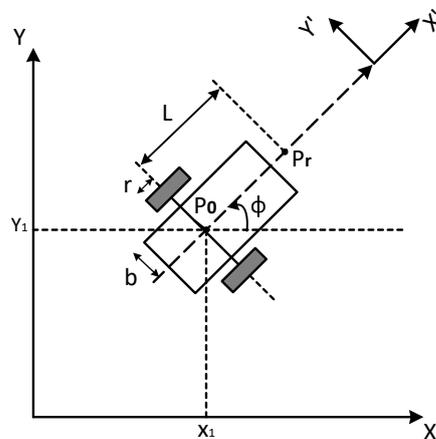


Figure 1. Differential drive mobile robot.

The authors in [37], showed the linearization of the input–output map using feedback linearization, as well as the nonlinear internal dynamics under look-ahead control for the point P_r . The linear behavior of the system can be seen in Equation (9), and the internal dynamics of the wheels are in Equation (10).

$$\dot{z}' = Az' + Bu \tag{9}$$

$$\dot{z}'' = w(z', z'') \tag{10}$$

$$y = Cz' \tag{11}$$

where

$$z = \begin{bmatrix} z' \\ z'' \end{bmatrix}, \quad z' = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix}, \quad z'' = \begin{bmatrix} z_5 \\ z_6 \end{bmatrix},$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$w(z', z'') = \Phi^{-1}(z) \begin{bmatrix} z_2 \\ z_4 \end{bmatrix} = -\frac{1}{2c^2bL} \begin{bmatrix} cb \sin \phi - cL \cos \phi & -cb \cos \phi - cL \sin \phi \\ -cb \sin \phi - cL \cos \phi & cb \cos \phi - cL \sin \phi \end{bmatrix} \begin{bmatrix} z_2 \\ z_4 \end{bmatrix},$$

z' is the vector of states that contains the linear positions and velocities of point P_r of the robot in the plane, while z'' is the vector that contains the angular positions of the wheels. c and b are parameters of the robot's geometry, while $L \neq 0$ is the distance between the center of mass and point P_r . ϕ represents the orientation of the mobile robot with respect to the x axis of the reference frame.

3.1. Problem Description

We address the problem of optimal trajectory planning for a differential drive mobile robot in a plane, building on the analysis performed for the double integrator (DI) in [38], and the dynamic model presented in Equation (9) and developed in [37]. The look-ahead motion control of the robot at point (P_r), is analogous to that of the DI in the plane. We will then present different solutions to the trajectory planning problem, including minimum distance, minimum control effort, minimum navigation time, and multi-objective optimization.

To obtain the reference trajectories, the motion of the robot is initiated from the initial condition of $(x_0, y_0) = (-1, -1)$ and terminates at the final condition of $(x_1, y_1) = (1, 0.5)$. The initial and final velocities are both zero, i.e., $(\dot{x}_0, \dot{y}_0) = (\dot{x}_1, \dot{y}_1) = (0, 0)$, and with a set of control signals $\{-0.1, 0, 0.1\}$ applied in the cell mapping algorithm, for a time of $t_s = 1$ s.

3.2. Combinatorial Optimization Using CMA for Minimum Distance (F_d)

We use Dijkstra's algorithm to find the shortest path between the initial and final cells of the graph, constructed by means of Equation (4). The initial conditions were described in the previous section. This generates the reference trajectory that minimizes the distance traveled by the differential drive mobile robot. The optimization problem has multiple solutions in the state space, corresponding to symmetries in the workspace or speed variations at the intermediate nodes. In the path found by the algorithm, the robot passes through different cells in the state space, increasing the navigation time due to decreases in velocity in the x direction. In this case, the time it takes the robot to complete the path is 12 seconds, and the distance is 2.6847 m.

Figure 2a displays the G_d data structure obtained using Algorithm 1, while Figure 2b,c shows the state space for each of the (x, y) plane dimensions found using Algorithm 2. Nodes 1, 7, 9, and 13 of the obtained trajectory are indicated with green, blue, cyan, and yellow colors, respectively. Node 1 corresponds to cell 223,881, node 7 to 276,425, node 9 to 351,341, and node 13 to 225,751. It should be noted that in Figure 2c, node 1 has three cells, which can be explained by the fact that the initial velocity on the y -axis is zero. The robot's motion in the plane can be observed in Section 3.5.

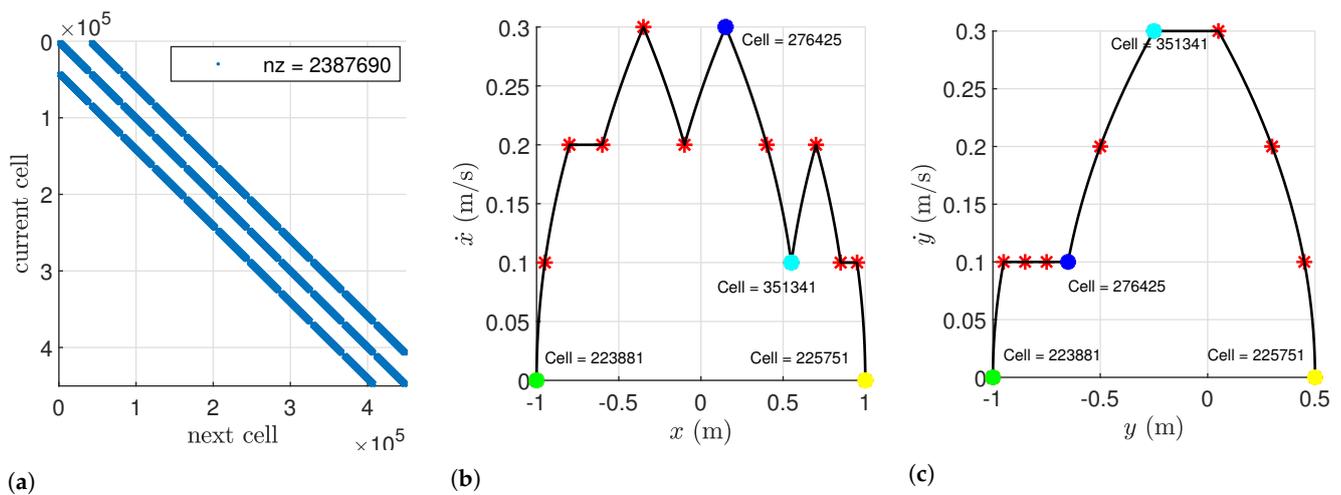


Figure 2. Data structure obtained by CMA with number of connections (nz) and state space of combinatorial optimization problem for minimum distance.

3.3. Combinatorial Optimization Using CMA for Minimum Control Effort (F_u)

Now, the idea is to obtain a reference trajectory that requires the lowest control effort to move the robot, from the same initial and final states. To achieve this goal, we used the cost function presented in Equation (5). As shown in the results, the mobile robot takes 26 s to complete the path, which is more than twice the time taken in the previous problem, and the control effort is $0.04 u^2$.

3.4. Combinatorial Optimization Using CMA for Minimum Navigation Time (F_t)

Once again, we implement the CMA to obtain a reference trajectory that minimizes the navigation time of a mobile robot. Since the total navigation time is minimized within the adopted discretization, we used the cost function shown in Equation (6) to construct the network. In this case, we observed that the mobile robot would take 10 s to complete the route, which is the fastest solution within the adopted discretization.

3.5. Multi-Objective Combinatorial Optimization Using CMA (F_{Mop})

We are now searching for the trajectory that minimizes the cost function presented in Equation (12), for demonstration purposes we use an arbitrary weighted sum, using the graphs computed for distance, control effort, and navigation time. If a different performance is desired, it is necessary to change the weights assigned to each optimization parameter in the cost function. Finding the values of the best-performing weights requires the computation of a Pareto set, which is a problem outside the scope of this research at the moment, but certainly will be considered as future work. By taking the same initial conditions as in the previous examples, we obtain a trajectory in which the robot would take 11 s to complete the path.

$$F_{MOP} = 0.78 * F_d + 0.2 * F_u + 0.02 * F_t \tag{12}$$

Figures 3 and 4 show the control signals obtained for the x and y dimensions, respectively, in each of the studied optimization cases. As can be seen, the maximum and minimum values are limited by the discretization of the control signals used in the cell mapping algorithm for the given example. Figures 3b,c and 4b,c exhibit the least amount of switching, as they correspond to the time and control effort minimization problems, where the former aims to maximize speed and the latter seeks to minimize control energy. In contrast, Figures 3a and 4a, which depict the control signals' behavior for the minimum distance case, show a high number of switching events. This is because the robot accelerates and decelerates at the intermediate nodes along this path. To address this issue, a cost function with an additional to distance optimization parameter can be employed.

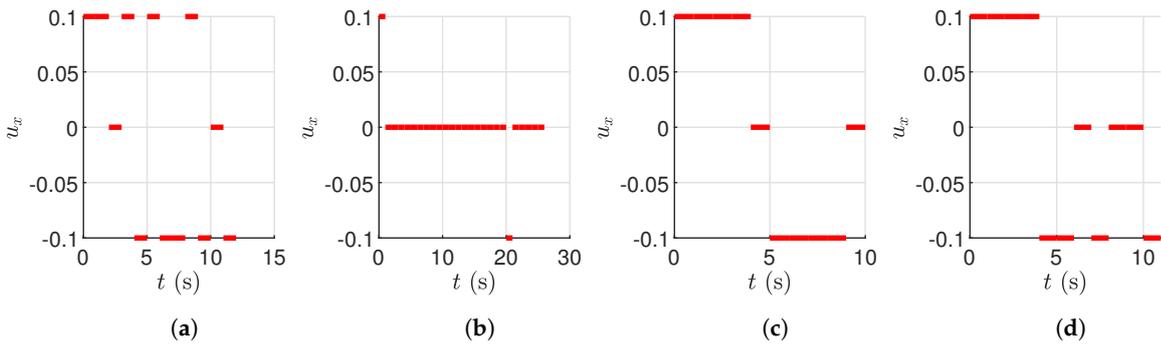


Figure 3. Control signals in x-dimension for each optimization case . (a) F_d . (b) F_u . (c) F_t . (d) F_{MOP} .

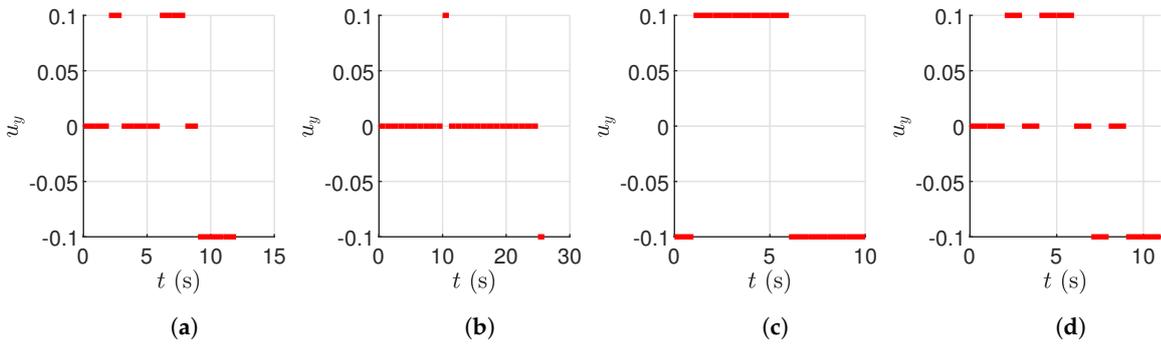


Figure 4. Control signals in y-dimension for each optimization case. (a) F_d . (b) F_u . (c) F_t . (d) F_{MOP} .

Figure 5 shows the robot’s motion results in the workspace, while Figures 6–9 show the state variables of point P_r of the robot. Figures 10–13 depict the angular positions (θ_1, θ_2) and angular velocities ($\dot{\theta}_1, \dot{\theta}_2$) of the robot’s wheels, the mobile platform rotation angle (ϕ), and the angular velocity of rotation ($\dot{\phi}$) for each of the optimization cases discussed above.

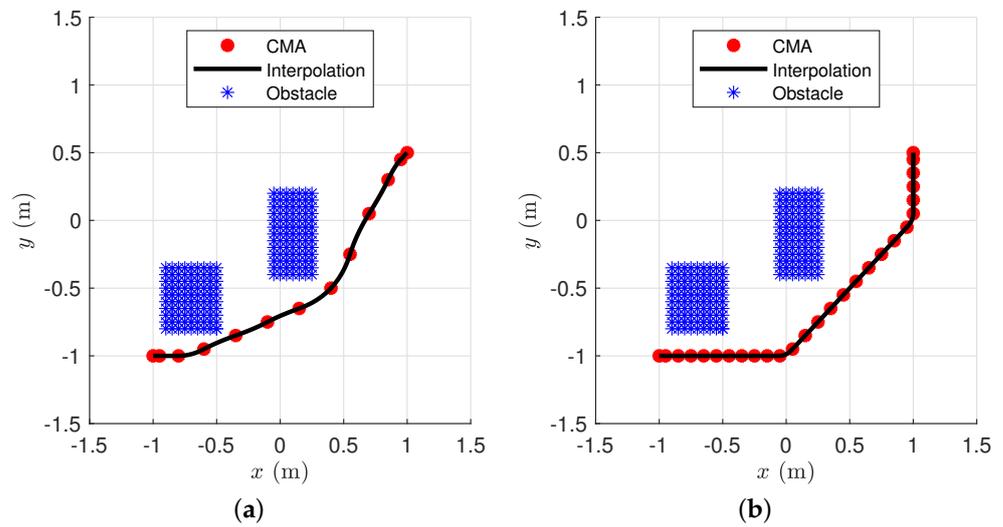


Figure 5. Cont.

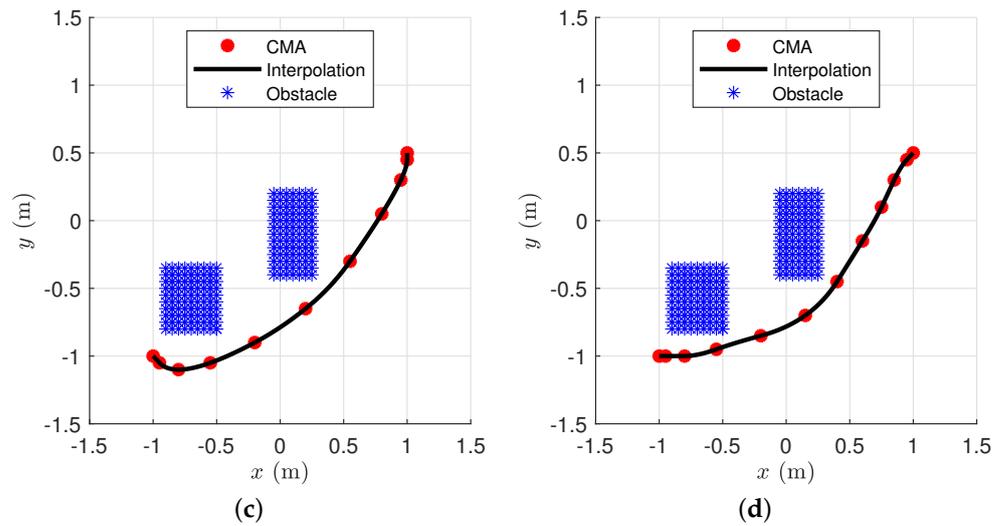


Figure 5. Position of the robot in the workspace: (a) shortest distance path (F_d), (b) least control effort path (F_u), (c) least navigation time path (F_t), and (d) minimum MOP path (F_{MOP}).

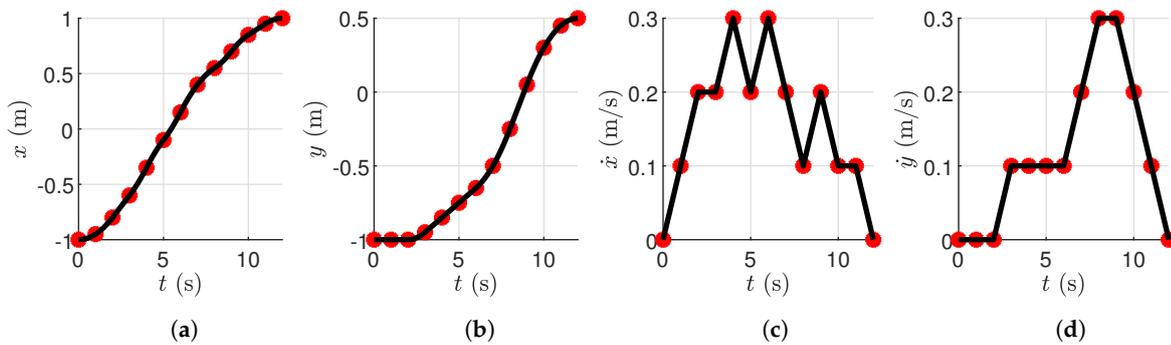


Figure 6. Robot position and velocity for minimum distance: (a) x , (b) y , (c) \dot{x} , and (d) \dot{y} . The red dots correspond to nodes found by CMA and black line to the interpolation taking into account position and velocity constraints given by CMA.

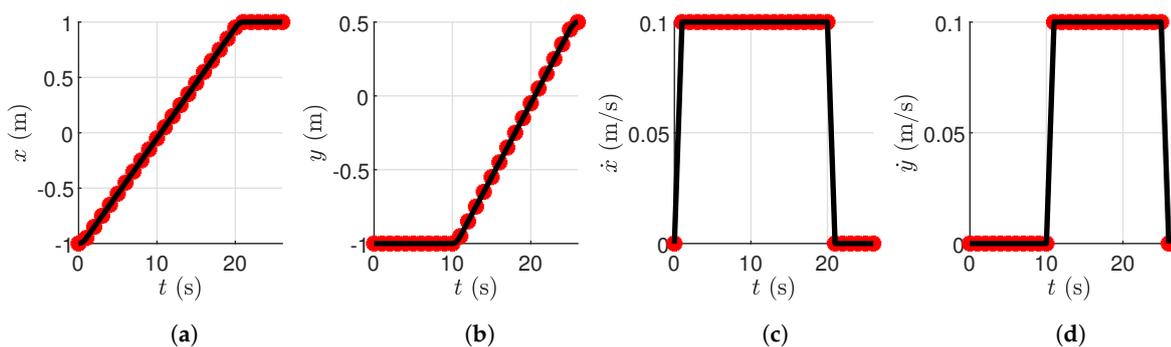


Figure 7. Robot position and velocity for minimum control effort: (a) x , (b) y , (c) \dot{x} , and (d) \dot{y} . Red dots (CMA) and black line (Interpolation).

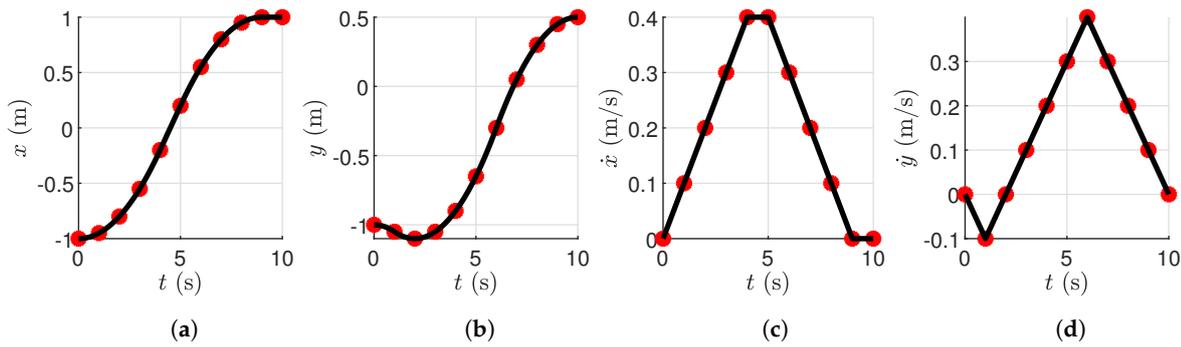


Figure 8. Robot position and velocity for minimum navigation time: (a) x , (b) y , (c) \dot{x} , and (d) \dot{y} . Red dots (CMA) and black line (Interpolation).

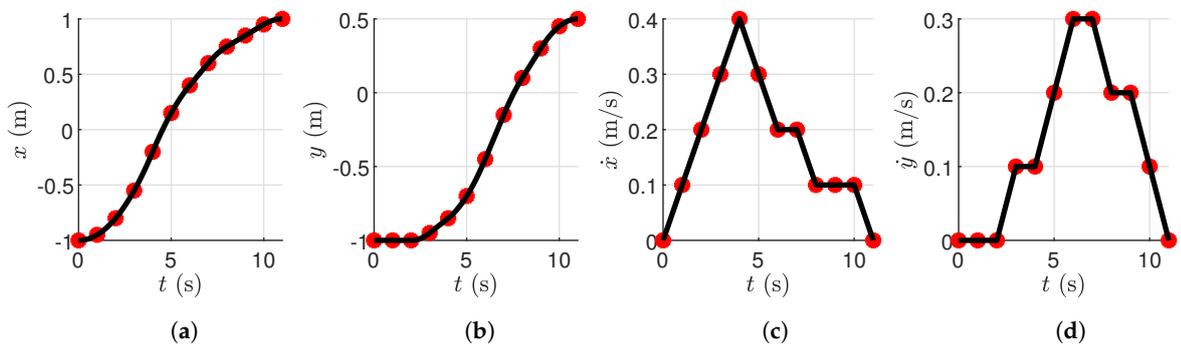


Figure 9. Robot position and velocity for minimum MOP: (a) x , (b) y , (c) \dot{x} , and (d) \dot{y} . Red dots (CMA) and black line (Interpolation).

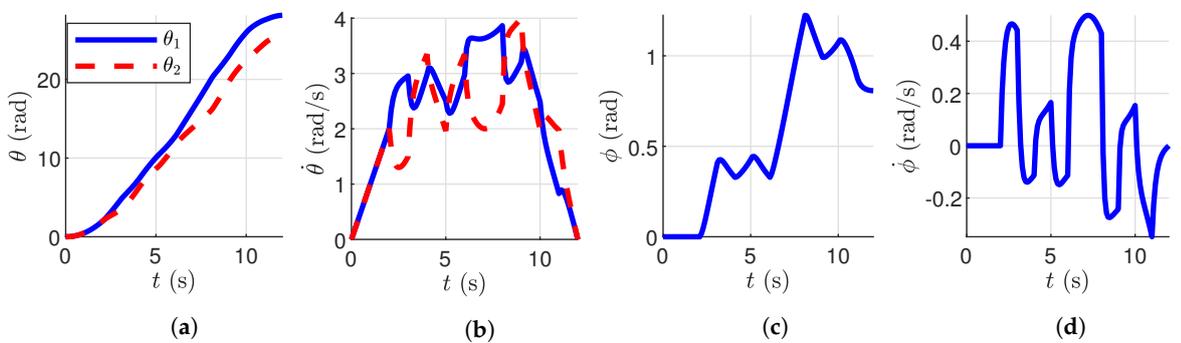


Figure 10. Position and angular velocity for minimum distance: (a) wheel position 1 = θ_1 (blue) and wheel position 2 = θ_2 (red), (b) wheel velocity 1 = $\dot{\theta}_1$ (blue) and wheel velocity 2 = $\dot{\theta}_2$ (red), (c) robot guidance ϕ , and (d) robot angular velocity $\dot{\phi}$.

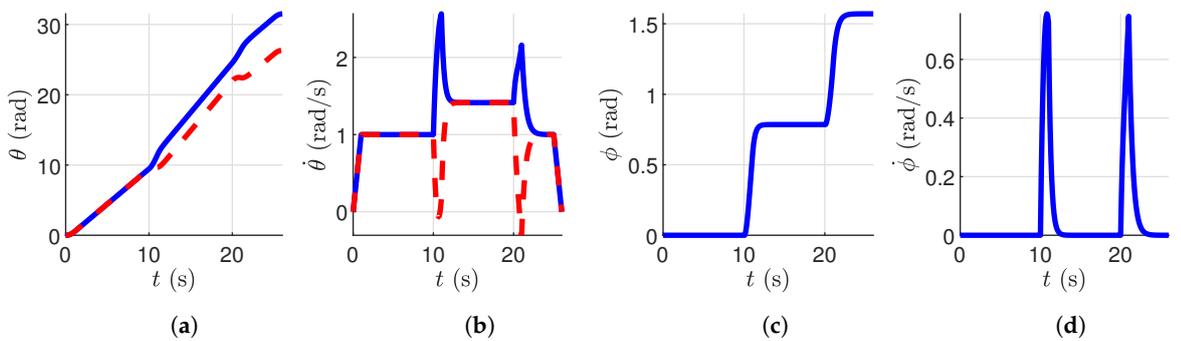


Figure 11. Position and angular velocity for minimum control effort: (a) wheels position = θ , (b) wheels velocity = $\dot{\theta}$, (c) robot guidance ϕ , and (d) robot angular velocity $\dot{\phi}$.

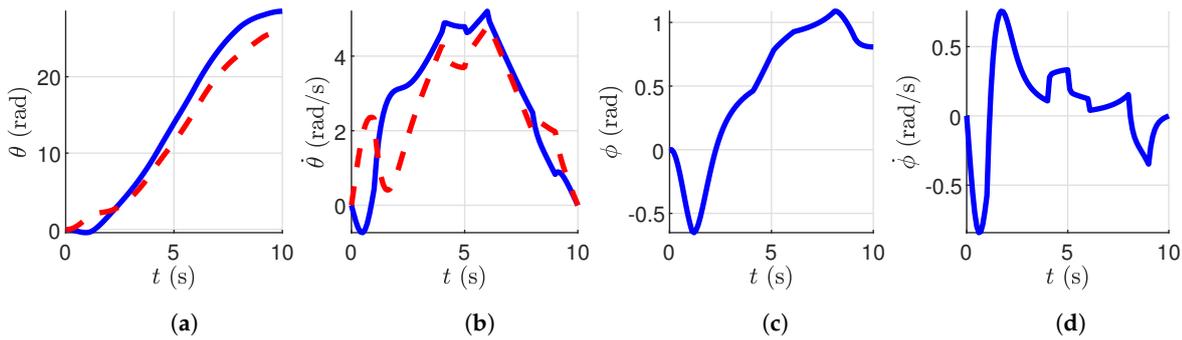


Figure 12. Position and angular velocity for minimum navigation time: (a) wheels position = θ , (b) wheels velocity = $\dot{\theta}$, (c) robot guidance ϕ , and (d) robot angular velocity $\dot{\phi}$.

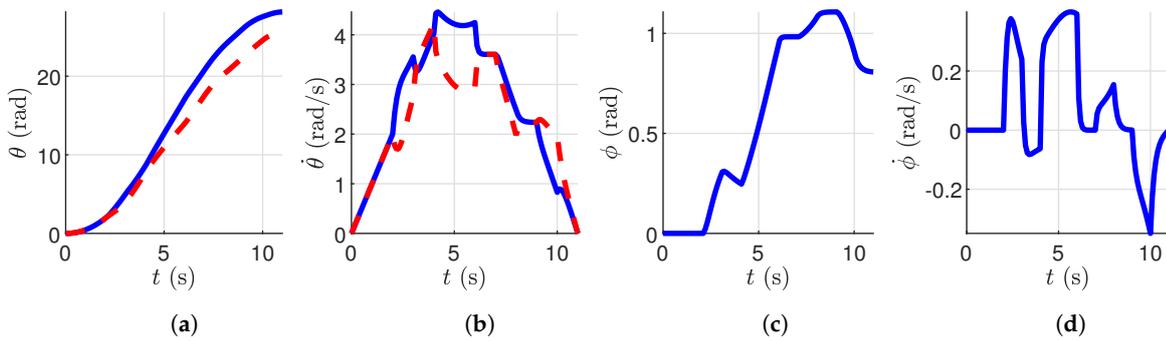


Figure 13. Position and angular velocity for minimum MOP: (a) wheels position = θ , (b) wheels velocity = $\dot{\theta}$, (c) robot guidance ϕ , and (d) robot angular velocity $\dot{\phi}$.

4. Discussion

4.1. Comparison of Trajectories Generated by the CMA

To analyze the performance of the proposed method using CMA, we present a comparison of the four optimization cases mentioned above (see Figure 14). We can observe that all of the trajectories avoid obstacles. In the solution of the minimum navigation time problem, we see the undershoot phenomenon, associated with non-minimum phase systems for differential drive mobile robots [39]. The nature of the minimum distance and time problems is opposite to the minimum control effort problem, and as a result, the solutions can be significantly different. The MOP solution represents a compromise between all of the above cases.

Table 1 shows the results of the distance, control effort, navigation time, and the multi-objective optimization for each of the trajectories generated by the CMA in the four combinatorial optimization problems presented above. As expected, the minimum values are on the main diagonal. When considering the combination of the three performance criteria (distance, control effort, and time), it is found that the MOP trajectory efficiently combines all of them, as it has a small control effort, navigation time, and distance.

Table 1. Comparison of trajectories using CMA with different cost functions.

Path	Optimization Criterion	Distance (m)	Effort (u^2)	t (s)	MOP Cost
1	Distance	2.6847	0.16	12	2.3661
2	Control Effort	2.9023	0.04	26	2.7918
3	Time	2.7702	0.18	10	2.3709
4	MOP	2.6862	0.14	11	2.3432

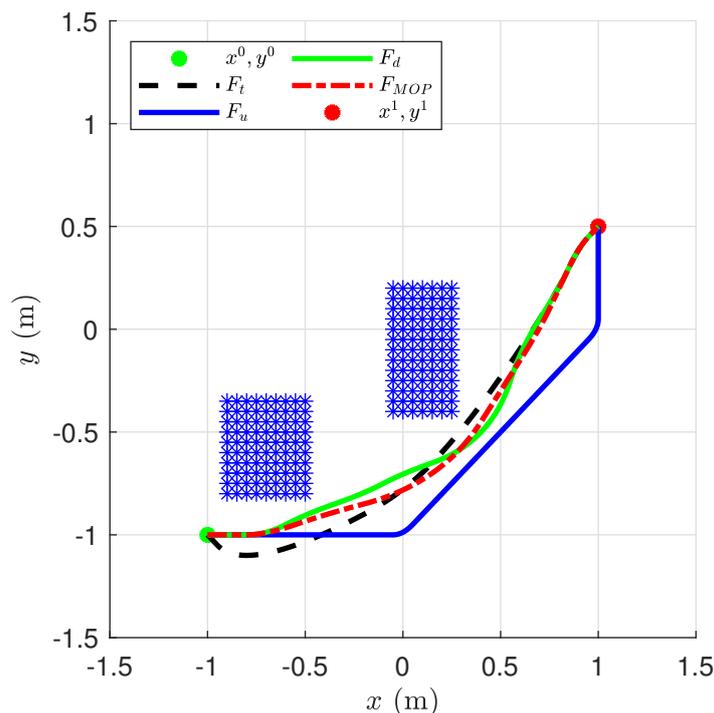


Figure 14. Planned trajectories using CMA with F_d , F_u , F_t , and F_{MOP} .

4.2. Comparison of Different Trajectory Planners

We now present a comparison of the distance of the paths generated by four different methods: (a) combinatorial optimization using CMA for minimum distance, (b) multi-objective combinatorial optimization problem using CMA, (c) the shortest path search algorithm A^* , and (d) the optimized version of the rapidly exploring random tree (RRT^*) sampling method. This comparison is only based on distance, as the classical algorithms implemented generate paths only based on position. Figure 15 shows the four solutions in the case of a mobile robot moving as per the described problem.

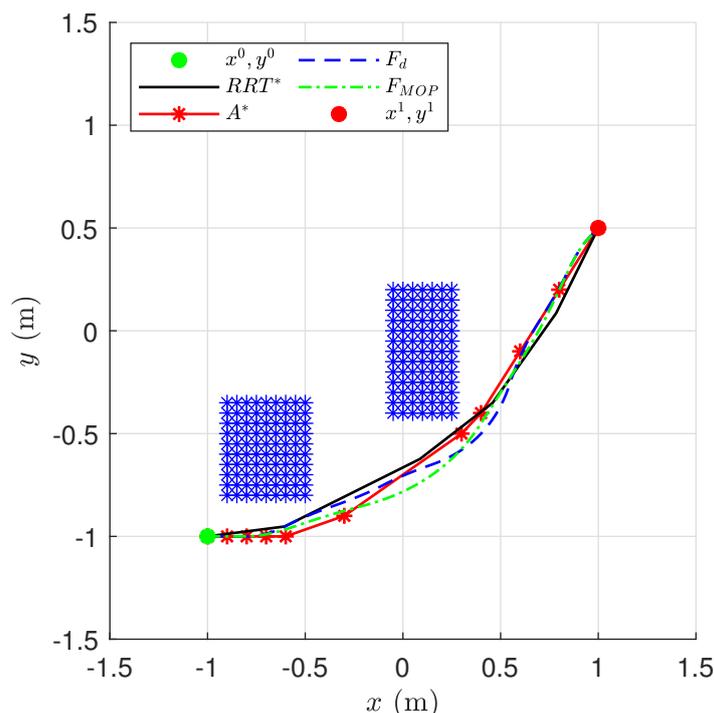


Figure 15. Comparison of trajectory planners.

Table 2 shows the distance generated by each of the paths. The planner that generates the path with the shortest distance is RRT^* , which does not take into account the dynamics, while the planner based on the CMA does. The CMA with F_d , presents oscillations in the intermediate speeds, increasing the navigation time compared to the CMA with F_{MOP} . The latter delivers a route with good performance in terms of speed profiles, navigation time, distance, and control effort.

The A^* planner is implemented because it is one of the most widely used classical methods in the literature, and is known for generating short trajectories that avoid obstacles. However, for this example, the algorithm finds a solution to the distance optimization problem that is not the minimum of the problem, according to the selected heuristic function. Moreover, this method does not take into account the others parameters of the path: time, control effort, and energy consumption.

When comparing the planners in terms of percentage, with the RRT^* algorithm taken as a reference, it is found that the CMA with F_d generates a path that is 1.69% longer, while the CMA with F_{MOP} generates a path that is 1.75% longer. Additionally, the CMA takes into account velocity profiles, the time vector for the mobile robot, and incorporates dynamic parameters of the robot, to obtain better performance in path tracking.

Table 2. Comparison of trajectories of a robot's motion in the plane.

Path	Planner	Distance (m)
1	RRT^* with maximum distance of connection 0.8 m	2.6400
2	A^*	2.6604
3	CMA with F_d	2.6847
4	CMA with F_{MOP}	2.6862

5. Conclusions

The use of optimal control theory for motion planning is a difficult task. Cell mapping offers a methodology to formulate combinatorial optimization problems, allowing the inclusion of complex cost functions, as well as multi-objective optimization problems. The proposed method allows the transformation of continuous optimization problems into discrete optimization problems. With this, suboptimal solutions are obtained, which depend on the discretization of the adopted cell.

The cell mapping algorithm (CMA) can be systematically used to plan trajectories for mobile robots navigating complex spaces with obstacles, provided that these robots are represented by a linear mathematical model. One of the advantages of using the CMA, is that it optimizes not only the distance criterion but also additional parameters, such as navigation time and control effort. By generating graphs that combine different optimization criteria in the objective function, the CMA algorithm enables us to find the shortest path in the resulting data structure. In this paper, a comparison was made of the distance optimization criterion, which resulted in a path that was only 2% longer than the path generated by the state-of-the-art rapidly exploring random tree star (RRT^*) algorithm. For future work, we intend to use other path planning algorithms, to compare the energy consumption and navigation time obtained by the CMA.

The complexity involved in constructing the CMA lies in the discretization of the state space, as well as in selecting the set of control signals to be used. For future work, we intend to explore dynamic discretization methods, similar to QuadTree, to optimize the number of cells. We also intend to change the way of computing each of the simulations, taking advantage of the geometry of the problem and also employing parallelization techniques.

The proposed methodology, utilizing CMA, can be applied to formalize optimization problems in mobile robot trajectory planning with various cost functions, such as minimum energy, maximum safety, minimum total variation, jerk minimization, and smoothness, as well as multi-objective problems that combine these functions. This is in contrast to

the classical methods studied in this paper, which only optimize distance, and provide position vectors for constructing the route. The use of CMA enables the inclusion of more complex cost functions, making it a more versatile and powerful tool for mobile robot trajectory planning.

Author Contributions: Conceptualization, E.G.-R.; methodology, E.G.-R. and G.O.; data analysis and simulations, E.G.-R.; validation, E.G.-R.; writing—original draft preparation, E.G.-R. and G.O.; writing—review and editing, E.G.-R. and G.O.; funding acquisition, E.G.-R. and G.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding. 50 percent of the APC was financed by the Universidad Autónoma de Manizales.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the program of scholarships by degree of honor of undergraduate degree scholarship program of the Universidad Nacional de Colombia and the Universidad Autónoma de Manizales, for their support for publishing the article. Additionally, the authors would like to thank the Department of Electrical, Electronic and Computer Engineering of the Universidad Nacional de Colombia in Manizales for their partnership.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CMA	Cell mapping algorithm
SCM	Simple cell mapping
DI	Double integrator
1D	1-Dimensional
2D	2-Dimensional
MOP	Multi-objective optimization problem
RRT*	Rapidly exploring random tree star

References

1. LaValle, S.M. Motion Planning Part 2: Wild Frontiers. *IEEE Robot. Autom. Soc. Mag.* **2011**, *18*, 108–118. [[CrossRef](#)]
2. Miao, C.; Chen, G.; Yan, C.; Wu, Y. Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm. *Comput. Ind. Eng.* **2021**, *156*, 107230. [[CrossRef](#)]
3. Lee, G.D.S.; Lee, K.S.; Park, H.G.; Lee, M.H. Optimal path planning with holonomic mobile robot using localization vision sensors. In Proceedings of the Control Automation and Systems (ICCAS), Gyeonggi, Republic of Korea, 27–30 October 2010; pp. 1883–1886.
4. Masoud, A. Kinodynamic motion planning. *IEEE Robot. Autom. Mag.* **2010**, *17*, 1048–1066. [[CrossRef](#)]
5. Yang, D.; Xie, F.; Liu, X. Velocity Constraints Based Approach for Online Trajectory Planning of High-Speed Parallel Robots. *Chin. J. Mech. Eng. (Engl. Ed.)* **2022**, *35*, 127. [[CrossRef](#)]
6. Hoang, H.; Khoa Tran, A.; Nhat Thai Tran, L.; Le, M.; Tran, D. A Shortest Smooth-path Motion Planning for a Mobile Robot with Nonholonomic Constraints. In Proceedings of the IEEE International Conference on System Science and Engineering, Ho Chi Minh City, Vietnam, 26–28 August 2021; pp. 145–150. [[CrossRef](#)]
7. Galgamuwa, G.I.R.K.; Liyanage, L.K.G.; Ekanayake, M.P.B.; Samaranyake, B.G.L.T. Simplified controller for three wheeled omni directional mobile robot. In Proceedings of the IEEE 10th International Conference on Industrial and Information Systems (ICIIS), Peradeniya, Sri Lanka, 18–20 December 2015; pp. 314–319. [[CrossRef](#)]
8. Azizi, M.R.; Rastegarpanah, A.; Stolkin, R. Motion Planning and Control of an Omnidirectional Mobile Robot in Dynamic Environments. *Robotics* **2021**, *10*, 48. [[CrossRef](#)]
9. Cong, D.; Liang, C.; Gong, Q.; Yang, X.; Liu, J. Path Planning and Following of Omnidirectional Mobile Robot Based on B-spline. In Proceedings of the Chinese Control and Decision Conference (CCDC), Shenyang, China, 9–11 June 2018; pp. 4931–4936. [[CrossRef](#)]

10. Wang, J.; Chepinskiy, S.A.; Krasnov, A.J.; Zhang, B.; Liu, H.; Chen, Y.; Khvostov, D.A. Geometric path following control for an omnidirectional mobile robot. In Proceedings of the 21st International Conference on Methods & Models in Automation & Robotics (MMAR), Międzyzdroje, Poland, 29 August–1 September 2016; pp. 1063–1068. [CrossRef]
11. Kim, H.; Kim, B.K. Online Minimum-Energy Trajectory Planning and Control on a Straight-Line Path for Three-Wheeled Omnidirectional Mobile Robots. *IEEE Trans. Ind. Electron.* **2014**, *61*, 4771–4779. [CrossRef]
12. Masmoudi, M.S.; Krichen, N.; Masmoudi, M.; Derbel, N. Fuzzy logic controllers design for omnidirectional mobile robot navigation. *Appl. Soft Comput.* **2016**, *49*, 901–919. [CrossRef]
13. Al-Taharwa, I.; Sheta, A.; Al-Weshah, M. A mobile robot path planning using genetic algorithm in static environment. *J. Comput. Sci.* **2008**, *4*, 341–344. [CrossRef]
14. Thomaz, C.E.; Vellasco, M. Mobile robot path planning using genetic algorithms. In *World Wide Web Internet And Web Information Systems*; Springer: Berlin/Heidelberg, Germany, 1999.
15. Lu, J.; Yang, D. Path planning based on double-layer genetic algorithm. In Proceedings of the 3rd International Conference on Natural Computation (ICNC), Hainan, China, 24–27 August 2007.
16. Buniyamin, N.; Ngah, W.W. A simple local path planning algorithm for autonomous mobile robots. *Int. J. Syst. Appl. Eng. Dev.* **2011**, *5*, 151–159.
17. Goyal, J.K.; Nagla, K.S. A new approach of path planning for mobile robots. In Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI), Delhi, India, 24–27 September 2014; pp. 863–867.
18. Stentz, A. Optimal and efficient path planning for partially-known environments. In Proceedings of the IEEE International Conference on Robotics and Automation, San Diego, CA, USA, 8–13 May 1994; pp. 3310–3317. [CrossRef]
19. Lingelbach, F. Path planning using probabilistic cell decomposition. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), New Orleans, LA, USA, 26 April–1 May 2004; pp. 467–472. [CrossRef]
20. Piaggio, M.; Zaccaria, R. Using roadmaps to classify regions of space for autonomous robot navigation. *Robot. Auton. Syst.* **1998**, *25*, 209–217. [CrossRef]
21. Yang, S.X.; Luo, C. A neural network approach to complete coverage path planning. *IEEE Trans. Syst.* **2004**, *34*, 718–725. [CrossRef] [PubMed]
22. Shi, B.; Cheng, P.; Cheng, N. 3D Flight Path Planning Based on RRTs for RNP Requirements. In Proceedings of the IEEE International Conference on Information and Automation, Chengdu, China, 5–8 August 2012; pp. 51–56.
23. Luger, G.F. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*; Addison Wesley: Boston, MA, USA, 2005.
24. LaValle, S.M. *Planning Algorithms*; Cambridge University Press: Cambridge, UK, 2006; p. 842. [CrossRef]
25. Gasparetto, A.; Boscaroli, P.; Lanzutti, A.; Vidoni, R. Path Planning and Trajectory Planning Algorithms: A General Overview. In *Motion and Operation Planning of Robotic Systems: Background and Practical Approaches*; Carbone, G., Gomez-Bravo, F., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 3–27. [CrossRef]
26. Karaman, S.; Frazzoli, E. Sampling-based Algorithms for Optimal Motion Planning. *J. Robot. Res.* **2011**, *30*, 846–894. [CrossRef]
27. Dong, Z.; Chen, Z.; Zhou, R.; Zhang, R. A hybrid approach of virtual force and A* search algorithm for UAV path re-planning. In Proceedings of the 6th IEEE Conference on Industrial Electronics and Applications, Beijing, China, 21–23 June 2011; pp. 1140–1145. [CrossRef]
28. Hsu, C.S. *Book: Cell-to-Cell Mapping: A Method of Global Analysis for Nonlinear Systems*; Springer: Berlin/Heidelberg, Germany, 1987. [CrossRef]
29. Zufiria, P.J.; Guttalu, R.S. The adjoining cell mapping and its recursive unraveling, part I: Description of adaptive and recursive algorithms. *Nonlinear Dyn.* **1993**, *4*, 207–226. [CrossRef]
30. Hsu, C.S. A probabilistic theory of nonlinear dynamical systems based on the cell state space concept. *J. Appl. Mech. Trans. ASME* **1982**, *49*, 895–902. [CrossRef]
31. Hsu, C.S. Global analysis of dynamical systems using posets and digraphs. *Int. J. Bifurc. Chaos* **1995**, *5*, 1085–1118. [CrossRef]
32. Sun, J.Q.; Xiong, F.R.; Schütze, O.; Hernández, C. *Book: Cell Mapping Methods: Algorithmic Approaches and Applications*; Springer: Singapore, 2018; pp. 1–226. [CrossRef]
33. Zhang, W.; Han, B.; Li, X.; Sun, J.; Ding, Q. Multiple-objective design optimization of squirrel cage for squeeze film damper by using cell mapping method and experimental validation. *Mech. Mach. Theory* **2019**, *132*, 66–79. [CrossRef]
34. Zhang, W.; Han, B.; Li, X.; Sun, J.; Ding, Q. Multi-objective system optimization method and experimental validation of a centralized squeeze film damper using a cell mapping method considering dynamic constraints. *Eng. Optim.* **2021**, *53*, 941–961. [CrossRef]
35. Schütze, C.H.O.; Sun, J.Q. *Global Multi-Objective Optimization by Means of Cell Mapping Techniques*; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; pp. 25–56. [CrossRef]
36. Qin, Z.C.; Xiong, F.R.; Ding, Q.; Hernández, C.; Fernandez, J.; Schütze, O.; Sun, J.Q. Multi-objective optimal design of sliding mode control with parallel simple cell mapping method. *J. Vib. Control.* **2017**, *23*, 46–54. <https://doi/10.1177/1077546315574948>. [CrossRef]
37. Yun, X.; Yamamoto, Y. Internal dynamics of a wheeled mobile robot. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 26–30 July 1993; pp. 1288–1294. [CrossRef]

38. Grisales-Ramirez, E.; Jaramillo-Morales, M.F.; Osorio, G.; Gomez-Mendoza, J.B. Multi-objective optimal path planning for mobile robots using state space cell mapping. In Proceedings of the 4th IEEE Colombian Conference on Automatic Control (CCAC), Medellin, Colombia, 15–18 October 2019. [[CrossRef](#)]
39. Nakata, R.; Tanemura, M.; Chida, Y. Undershoot Responses of Circular Path-Following Control for a Vehicle Based on Time-State Control Form. *IFAC-PapersOnLine* **2021**, *54*, 66–71. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.