applied sciences

MDPI

Article

Hybrid Sampling-Based Path Planning for Mobile Manipulators Performing Pick and Place Tasks in Narrow Spaces

Hanlin Chen *, Xizhe Zang, Yanhe Zhu and Jie Zhao

State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, Harbin 150001, China; zangxizhe@hit.edu.cn (X.Z.); yhzhu@hit.edu.cn (Y.Z.); jzhao@hit.edu.cn (J.Z.)
* Correspondence: 19b908064@stu.hit.edu.cn

Abstract: A mobile manipulator is capable of traversing a vast area while performing manipulation tasks in confined spaces. However, the high degree of freedom presents a challenge for path planning. In this paper, a hybrid sampling-based path planning method is proposed for mobile manipulators performing pick and place tasks in confined spaces. This method employs a random sampling approach, yet differs from the traditional RRT method. Firstly, a sampling-based configuration generation method for mobile manipulators is proposed, with the objective of generating a valid, collision-free configuration with the end-effector at the desired pose. A path for the end-effector corresponding to the goal configuration is then planned using the RRT method. Secondly, an area-restricted approach that samples in the vicinity of the previous configuration is introduced to generate the next valid configuration. Subsequently, a cost computation rule is devised to identify the optimal subsequent configuration utilizing the trajectory of the end-effector as a guiding principle. Finally, the obtained path is smoothed. Simulations demonstrate that the proposed hybrid sample-based method is an effective solution to the path planning problem for mobile manipulators performing pick and place tasks in narrow spaces.

Keywords: mobile manipulator; motion planning; collision avoidance; sampling-based method

check for updates

Citation: Chen, H.; Zang, X.; Zhu, Y.; Zhao, J. Hybrid Sampling-Based Path Planning for Mobile Manipulators Performing Pick and Place Tasks in Narrow Spaces. *Appl. Sci.* **2024**, *14*, 10313. https://doi.org/10.3390/app142210313

Academic Editors: Slawomir Jan Stepien, Marcin Chodnicki, Wojciech Giernacki and Dariusz Horla

Received: 11 October 2024
Revised: 4 November 2024
Accepted: 7 November 2024
Published: 9 November 2024

Copyright: © 2024 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

A mobile manipulator (MM) is defined as a mobile base with one (or more) traditional robotic arm, thereby combining the advantages of both mobile robots and fixed-base manipulators [1]. A mobile manipulator's ability to traverse a vast area while retaining dexterous manipulation capabilities renders it a versatile tool for a multitude of applications, including rescue operations [2], object transportation [3], fruit harvesting [4], large-scale complex component machining [5], flexible manufacturing [6], and healthcare [7] in different scenarios. A mobile manipulator is frequently a kinematically redundant system, with more than nine degrees of freedom (DOFs). The redundancy of mobile manipulators increases their flexibility when operating in complex environments [8]. However, this also makes the path planning process complex and challenging due to the infinite solutions to the inverse kinematic problem. Consequently, the efficient planning of a safe and reachable path for a mobile manipulator is a topic worthy of further research.

Two principal categories of algorithms for robot path planning are search-based and sample-based algorithms. Search-based algorithms, also known as graph search algorithms, such as A* [9], require the creation of a graphical representation of the environment. This is followed by the search for the shortest-distance path or other lowest-cost path from the start to the goal. Some researchers have proposed methods to improve the efficiency of these algorithms [10]. However, the difficulty and complexity of creating graphical representations preclude the applicability of these algorithms to mobile manipulators with high DOFs.

Sampling-based algorithms, such as rapidly exploring random trees (RRT) [11], employ a random sampling process to select points and then connect these samples to generate

a tree. These methods can be employed in high-dimensional spaces and complex environments. One limitation of these methods is that they cannot guarantee optimal paths and are sensitive to sampling strategies [12]. Furthermore, the efficiency of the algorithm can be significantly impaired in confined or constrained environments. As a result, a number of RRT-based variants have been developed with the aim of enhancing efficiency, including RRT-connect [13], RRT* [14], informed RRT* [15], and EB-RRT [16]. Currently, Artificial Intelligence (AI), a tool for obstacle detection and optimization, has increased the performance of traditional algorithms [17].

When sampling-based algorithms are employed for mobile manipulators, such as those described in [18], the algorithms' dimensionality will be similarly high due to the high DOFs, and the search space will be exceedingly large, resulting in low sampling efficiency. Furthermore, the overall algorithm will be highly complex and time-consuming due to the necessity of complex collision detection for each sample. A novel sampling-based planning algorithm designed for mobile manipulators named XXL was proposed where workspace information is used as a guide through the high-dimensional planning space [19]. However, XXL also has a high computational overhead. One potential solution to this issue is to enhance the sampling strategy [20]. An alternative approach is to initially plan the trajectory in the workspace and subsequently utilize inverse kinematics and dynamic constraints to generate the configuration of the mobile manipulator at each waypoint along the path. The path of the end-effector is initially planned using a sampling-based algorithm. Subsequently, existing trajectory tracking methods can be employed for the mobile manipulator to follow the planned path [21,22]. These methods are more suitable for open environments, but it is challenging to obtain a valid solution that meets all constraints and does not collide with the environment in a confined space. Furthermore, the differing kinematic characteristics of the mobile base and the manipulator present an additional challenge in determining the optimal base position and the pose of the manipulator [23,24].

To avoid the above problems, the mobile base and the manipulator are planned separately using suitable methods. For example, the RRT-connect method can be used for both the mobile base and the manipulator [25], but planning is performed separately. It is also possible to use different methods, e.g., the mobile base is planned using a search-based method and the manipulator is planned using a sampling-based method [26,27]. These methods improve the efficiency of path planning, but they do not consider the motion coupling between the mobile base and the manipulator, and so they cannot adapt to confined and complex environments nor can they fully exploit the flexibility provided by the redundant degrees of freedom of the mobile manipulator. Therefore, the idea of adaptive dimensionality has been proposed, i.e., planning in high dimensions only when a feasible path cannot be searched in a low dimension [28]; Hierarchical and Adaptive Mobile Manipulator Planner (HAMP) [29] and Optimized Hierarchical Mobile Manipulator Planner (OHMP) [30] have been proposed based on this idea. These methods reduce the computational burden in a high-dimensional space, but still need to plan in high dimensions and suffer from a long, time-consuming search for a feasible path. In addition, similar to the idea of adaptive dimensionality, changing the robot pose only when necessary during the robot's movement is another feasible idea [31,32]; however, such methods focus on improving the ability of the mobile manipulator to move over a wide range and pass through some obstacles, but still do not solve the planning challenge of performing manipulation tasks in confined spaces.

In this paper, we propose a hybrid sampling-based method for the motion planning of pick and place tasks for mobile manipulators. We note that some studies [33,34] have proposed planning methods for mobile manipulator manipulation tasks, but these methods are applicable to open spaces where collision avoidance is not a very difficult problem. Our approach is dedicated to the problem of collision-free path planning in confined spaces. The proposed approach refers to tunnel-like path planning and mixes different random sampling-based methods. Currently, operations in confined spaces such as pipelines are carried out using complex devices [35]. Mobile manipulators have the potential to be

applied to operations in relatively short pipelines using the proposed planning method. The main contributions of our work are as follows: (1) A sampling-based configuration generation method is proposed to rapidly generate a collision-free configuration for a mobile manipulator, including the mobile base pose and manipulator joint states, that satisfies the target configuration of the end-effector. (2) A sampling strategy that samples in the neighborhood of the last configuration of the mobile manipulator is introduced, which avoids the inefficiency of sampling in a large space and increases the probability of obtaining a collision-free sample. (3) An informed guiding approach is introduced to determine the best pose among the samples.

The remainder of this paper is organized as follows: Section 2 presents the kinematic model of a mobile manipulator and defines the general problem. Section 3 introduces normal sampling-based algorithms for mobile manipulators, followed by an analysis of their shortcomings when applied in confined spaces. Section 4 introduces our hybrid sampling-based method, which includes sampling-based configuration generation, the sampling strategy, and the informed guiding approach. Section 5 presents the simulation results and conducts a comparative analysis of the different algorithms. Finally, Section 6 concludes the paper by summarizing the findings and providing future perspectives.

2. Modeling and Problem Formulation

2.1. Modeling of Mobile Manipulator

A typical single-arm mobile manipulator consists of a mobile base and a multi-joint manipulator with an end-effector. The mobile manipulator in this paper is shown in Figure 1. The mobile base is an ordinary mobile robot, which is differentially driven with two drive wheels. It can rotate or move forward in a plane. The manipulator has seven DOFs and has a gripper as an end-effector.

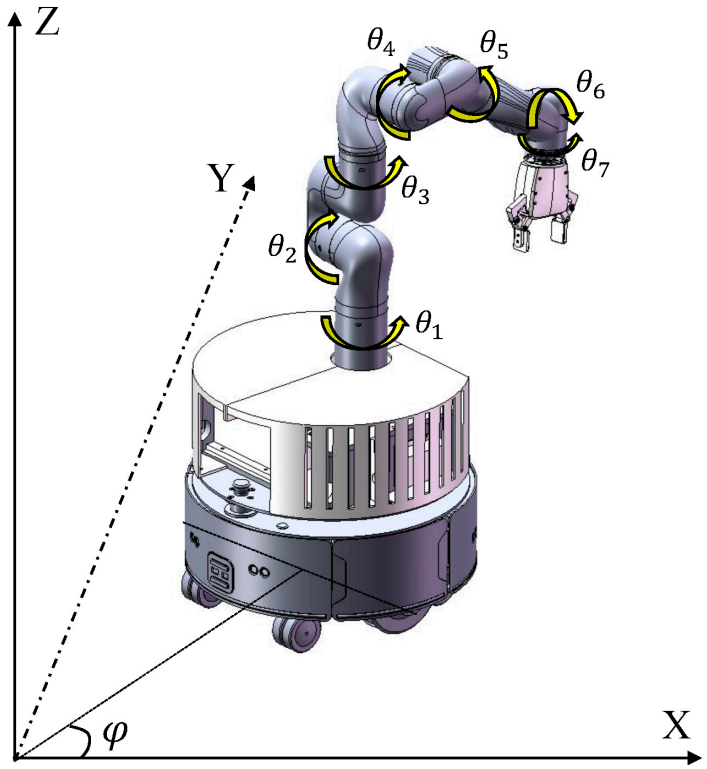


Figure 1. Model of a mobile manipulator.

The kinematic model of the mobile base can be represented by three parameters: the position of the center and the heading angle of the mobile base. The kinematic model of the manipulator can be developed using the classical modified Denavit–Hartenberg (D-H)

coordinate system. The overall kinematics of the mobile manipulator can be derived by transforming the manipulator coordinate to the mobile base coordinate. The transformation matrix from the end-effector to the world frame is

$$^{W}T_{E}\left(q^{mm}\right) = {}^{W}T_{MB}\,{}^{MB}T_{M}\,{}^{M}T_{E}, \tag{1}$$

where q^{mm} is the configuration of the mobile manipulator, $^{W}T_{MB}$ is the transformation matrix from the mobile base to the world frame, $^{MB}T_{M}$ is the transformation matrix from the manipulator to the mobile base, and $^{M}T_{E}$ is the transformation matrix from the end-effector to the manipulator.

2.2. Problem Statements

The path planning problem for a mobile manipulator in this paper can be defined as finding a collision-free path that starts from the initial configuration and reaches the goal configuration where the end-effector reaches the target position and satisfies the requirements of the pose of the end-effector, as shown in Figure 2.

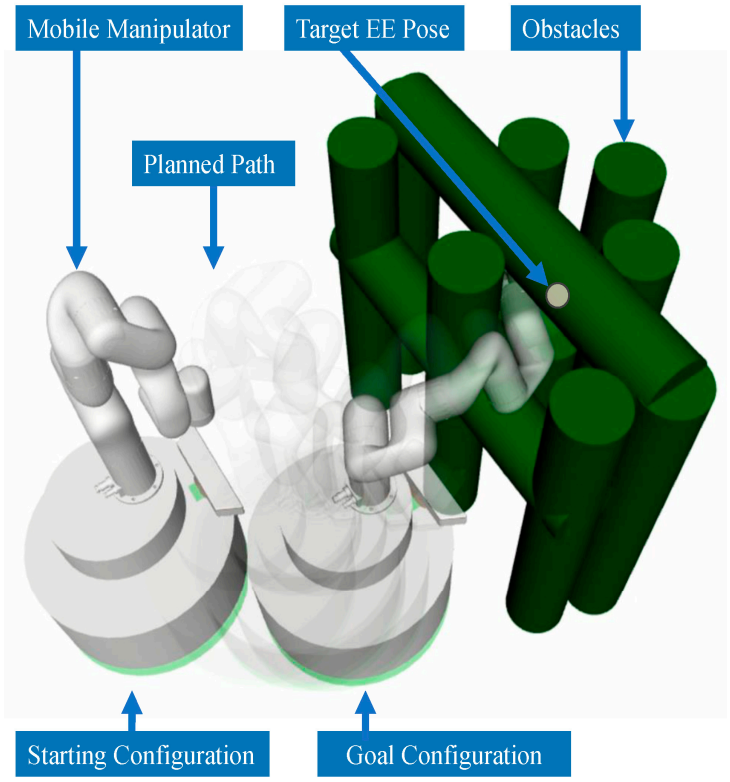


Figure 2. Mobile manipulator path planning description.

The configuration of the mobile base can be represented by a vector $q^{mb} = [x, y, \varphi]$, where x and y are the position of the mobile base and φ is the heading angle of the mobile base. The configuration of the manipulator can be represented by another vector $q^{m} = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7]$, where θ_i is the ith joint angle of the manipulator. The configuration of the mobile manipulator is $q^{mm} = [q^{mb}, q^{m}]$. The path for the mobile manipulator can be expressed as follows:

$$\Pi^{mm} = \left[q_s^{mm}, q_1^{mm}, \cdots, q_n^{mm}, q_g^{mm}\right]. \tag{2}$$

Π^{mm} is an ordered set of mobile manipulator configurations. q_s^{mm} is the starting configuration of the mobile manipulator for path planning. $q_i^{mm}(1 \leq \mathrm{i} \leq \mathrm{n})$ is a waypoint in the path. q_g^{mm} is the goal configuration of the mobile manipulator for path planning. Unlike

the start configuration, the goal configuration of the mobile manipulator is not directly given. In pick and place tasks, the position and rotation constraints of the end-effector are often given as known conditions. The constraints of the end-effector can be represented as $X_{goal} = [R, T]$, where R is the rotation constraint and T is the position constraint. So, q_g^{mm} that can satisfy the constraint X_{goal} and not collide with the environment should be calculated first in path planning for a mobile manipulator.

When q_s^{mm} and q_g^{mm} are known, a path Π^{mm} between them can be searched and smoothed for the mobile manipulator. Some sampling-based methods could solve this problem, but they are not suitable for pick and place tasks in a confined space.

3. Related Work and Shortcomings

3.1. Sampling-Based Path Planning for Mobile Manipulators

When sampling-based path planning methods are used for mobile manipulators, there are three different ways and they are mainly based on the RRT algorithm.

3.1.1. Using Traditional RRT Algorithms

Since RRT-based algorithms can be used in high-dimensional spaces, this is the simplest and most direct idea. The mobile manipulator is considered a whole system with high DOFs and the difference between the mobile base and the manipulator is neglected.

3.1.2. Using End-Effector Path and Inverse Kinematics

An available path for the end-effector is first planned using the RRT-based algorithm. Then, the inverse kinematics of the mobile manipulator are used to find the position and heading angle of the mobile base and each joint position of the manipulator at each waypoint of the end-effector path. This idea is often used in conventional robot arms.

3.1.3. Separate Planning

Another common idea is to plan the mobile base and the manipulator separately, and there are actually two different approaches. One approach is to plan the paths for the mobile base and the manipulator from the start configuration to the goal configuration using the RRT algorithm and then combine them together. The other approach is to plan the path of the mobile base first, adjust the configuration of the manipulator to avoid collision during the movement of the mobile base, and after the mobile base reaches the target position, plan the path for the manipulator that reaches the goal configuration.

3.2. Shortcomings

However, these methods have different shortcomings.

3.2.1. Time Consuming

The efficiency of sampling is directly related to the size and dimension of the sampling space. For the RRT algorithm, the processing time complexity of processing is $O(n\ logn)$, the time complexity of the query is $O(n)$, and the space complexity is $O(n)$ [36]. In a large sampling space with high dimensions, the algorithm would sample many more times and spend much more time searching for the available path. For mobile manipulators performing pick and place tasks in confined spaces, the main problem is that the probability of obtaining collision-free sampled configurations by random sampling is extremely low. Consequently, the number of samples required increases significantly, resulting in a significant reduction in efficiency (as shown in Figure 3a).

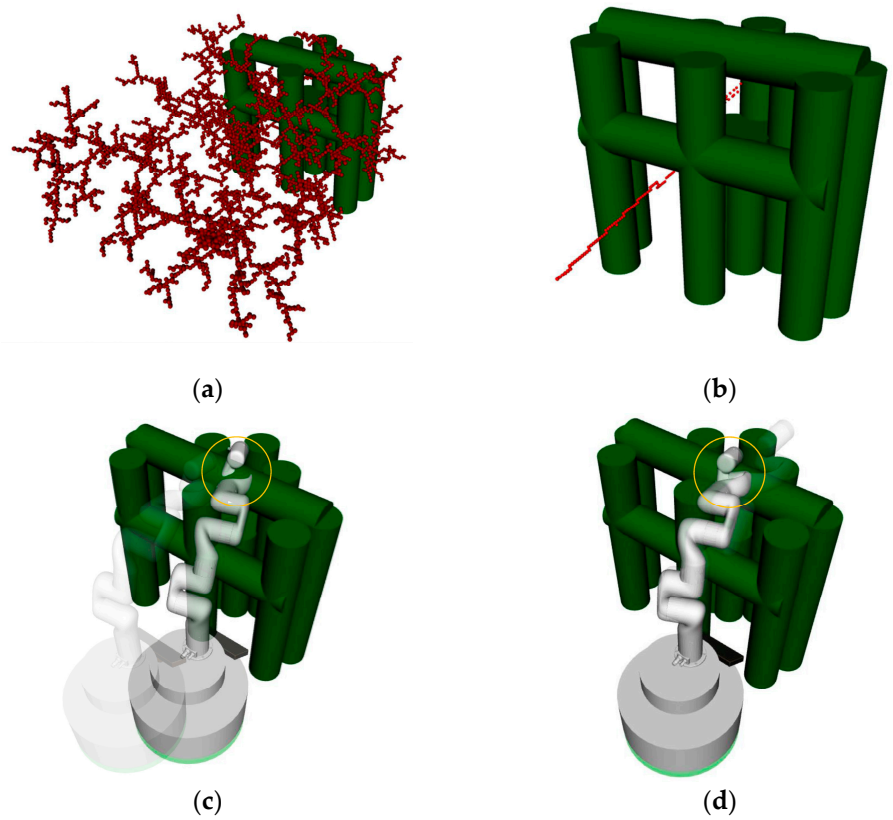


Figure 3. Issues. In (**a**), a large sampling space leads to large time consumption. In (**b**), there is no collision-free configuration for the mobile manipulator when the end-effector is at these waypoints. In (**c**), the manipulator collides with the environment while the mobile base reaches the target configuration. In (**d**), after the mobile base arrives at the target configuration, the collision-free path for the manipulator cannot be found due to the confined space.

3.2.2. Complex Inverse Kinematics

Planning a path for the end-effector first and using inverse kinematics to compute the configuration of the robot corresponding to each waypoint in the path is a common method for the path planning problem of traditional robotic arms. Since the mobile manipulator is a hyper-redundant system with two parts that have different kinematic characteristics, using inverse kinematics and dynamic constraints to compute a configuration is much more difficult than that of robotic arms. Confined spaces make it even worse. Considering that neither the mobile base nor the manipulator should collide with the environment, the process of obtaining a suitable mobile manipulator can become very complex. Furthermore, there may be some waypoints in the end-effector path for which there is no feasible collision-free configuration, which may cause the entire end-effector path to fail and lead to a failure of the mobile manipulator path planning (as shown in Figure 3b).

3.2.3. Neglecting the Manipulating Flexibility

Separate planning for the mobile base and the manipulator from the starting configuration to the goal configuration has a fatal flaw. The combined path can cause collisions between the robot and the environment since the other part is not considered in planning for the mobile base or the manipulator (refer to Figure 3c). This method is only suitable for open spaces without any obstacles. In contrast, a better approach is to allow the mobile base to reach the target configuration and then adjust the manipulator configuration so that the end-effector reaches the target pose. However, this approach does not take into account the coupled motion of the mobile base and the manipulator, and it ignores the manipulation flexibility provided by the redundant DOFs of the mobile manipulator. This method may

work well in some scenarios, but in some confined spaces, the manipulator may not be able to find an available collision-free path to make the end-effector reach the target pose, while the mobile base may have already reached the target configuration (refer to Figure 3d). In other words, in some scenarios, simultaneous movement of the mobile base and the manipulator is required to bring the end-effector to the goal pose while avoiding collision with the environment.

4. Approach

The overall process of the proposed approach is shown in Algorithm 1 Hybrid Sampling-based Path Planning (HSPP). First, a valid collision-free configuration consisting of both the mobile base and the manipulator that makes the end-effector at the target position satisfy the rotation constraints. A Sampling-based Configuration Generation (SCG) method is proposed to quickly generate the configuration and will be described in detail later. A smoothed path for the end-effector is then generated as a reference path for the mobile manipulator. Next, starting from the target configuration for the mobile manipulator, iterative sampling is performed in the neighborhood of the last waypoint and the configuration for the mobile manipulator that is closest to the reference path is obtained. The path for the mobile manipulator is generated iteratively. Finally, post-processing is performed to cut and smooth the whole generated path. The sampling strategy and the configuration selection principle will be described in detail later.

Algorithm 1: Hybrid Sampling-based Path Planning

1 **Input:** $q_s^{mm} := (q_s^{mb}, q_s^{m})$, $X_{goal} := [R, T]$
2 **Output:** Π^{mm}
3 $q_g^{mm} \leftarrow$ SCG (X_{goal})
4 $\Pi^{ee} \leftarrow$ GENERATEEEPATH(q_s^{mm}, q_g^{mm})
5 $\Pi^{mm} \leftarrow \{q_g^{mm}\}$
6 $q_{now}^{mm} \leftarrow q_g^{mm}$
7 **while** !ISREACHED(q_{now}^{mm}, q_s^{mm}) **do**
8 $q_{rand}^{mm} \leftarrow$SAMPLEINNEIGHBOURHOOD(q_{now}^{mm}, Π^{ee})
9 $\Pi^{mm} \leftarrow \Pi^{mm} \cup \{ q_{rand}^{mm} \}$
10 $q_{now}^{mm} \leftarrow q_{rand}^{mm}$
11 **end while**
12 PERFORMPOSTPROCESSING(Π^{mm})

4.1. Sampling-Based Configuration Generation Method

In pick and place tasks, only the position and rotation constraints of the end-effector are given, the corresponding configuration for the mobile manipulator should be calculated first before path planning. In the traditional RRT-based algorithm, the configuration of the mobile base and the grasping pose of the end-effector are first selected according to some principles, and then the configuration of the mobile manipulator is calculated using inverse kinematics. This method is useful in open spaces without massive obstacles [33], but in confined spaces, it is not a simple task. The main problem is that it is difficult to determine a suitable configuration of the mobile base and a suitable configuration of the end-effector. The concept of inverse reachability map seems to be useful here, but an inverse reachability map is not suitable for narrow spaces [37,38]. In order to ensure that the corresponding configuration is available and collision-free in the scenario, it is necessary to establish the inverse reachability map for the specific scenario. Therefore, we propose a method similar to the creation of an inverse reachability map to directly find a feasible mobile manipulator configuration corresponding to an end-effector pose.

In confined spaces, it is difficult to establish a rule for determining the configuration of the mobile base, since it is almost impossible to know whether a corresponding collision-free configuration of the manipulator exists. Therefore, we adopt a sampling-based approach to find a feasible collision-free configuration of the manipulator by trying different mobile base configurations and end-effector configurations. The procedure is shown in Algorithm 2.

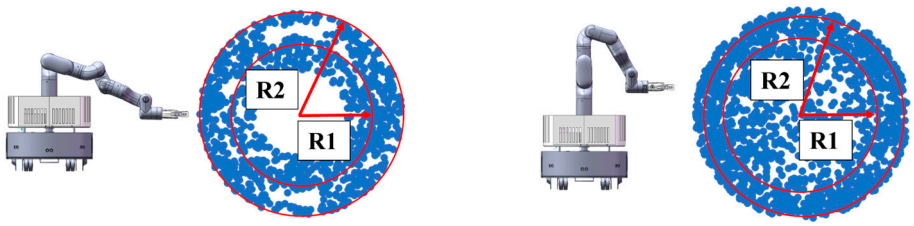
Algorithm 2: SCG(X_{goal})

Input: X_{goal}: = [R, T]
Output: q_g^{mm} : = [q_g^{mb} , q_g^{m}]
X_{ee} is the pose of the end-effector
while !ISMMCONFIGVALID(q_g^{mm}) **do**
    while !ISEECONFIGVALID(X_{ee}, X_{goal}) **do**
        X_{ee} ← SAMPLEEECONFIG(X_{goal})
    end while
    while !ISMBCONFIGVALID(q_g^{mb}) **do**
        q_g^{mb} ← SAMPLEMBCONFIG()
    end while
    q_g^{m} ← CALMANIPCONFIG(X_{ee}, q_g^{mb})
    q_g^{mm} ← [q_g^{mb} , q_g^{m}]
end while

First, a valid end-effector configuration X_{ee} is sampled based on the given end-effector position and rotation constraints. The configuration X_{ee} can be represented by a position matrix T and a rotation matrix R. Usually the position matrix is fully given, but the rotation matrix is partially given with some constraints. The orientation of the end-effector affects the mobile manipulator configuration and is related to the ability to find collision-free mobile manipulator configurations. For example, the end-effector is constrained to be parallel to the ground, but the direction is not required. In this case, the direction of the end-effector, or the rotation matrix R, could be sampled in order to generate a valid collision-free configuration for the manipulator. Algorithm 2 SAMPLEEECONFIG() is used to generate the end-effector configuration, and ISEECONFIGVALID() is used to check whether the configuration satisfies the constraints of X_{goal}.

The configuration of the mobile base q_g^{mb} should then be sampled, as shown in Algorithm 2 SAMPLEMBCONFIG(). A simple way to perform this is to sample the mobile base configuration within a circle centered on the end-effector position and radiused from the workspace radius of the manipulator. In practice, however, the mobile base positions that allow the acquisition of viable mobile manipulator configurations are not uniformly distributed within the circle. As shown in Figure 4, the images on the left illustrate the distribution of mobile base positions that can be obtained for a valid mobile manipulator configuration when the height of the end-effector is 0.3 m. Correspondingly, the images on the right illustrate the distribution of mobile base positions that can be obtained for a valid mobile manipulator configuration when the height of the end-effector is 0.9 m. The results were obtained from one thousand random sampling experiments. It can be seen that there are regions where feasible mobile base locations are more densely populated, and sampling in these regions has a higher probability of obtaining feasible mobile base locations. These areas are circular in shape and can be represented by their linear distance from the end-effector. Since moving the mobile base too far or too close to the end-effector reduces the manipulating flexibility of the manipulator, we can define a small area where 50% of the feasible mobile base positions are located, which is called the High Manipulating

Flexibility Area (HMFA) in this paper. As shown in Figure 4, the areas are the rings between R1 and R2. The R1 and R2 of the HMFA corresponding to each height of the end-effector can be derived from the results of the sampling experiment, which indicates the distributions of the valid mobile base positions corresponding to different end-effector heights (as shown in Figure 5).



(**a**) (**b**)

Figure 4. Valid mobile base positions with different end-effector heights. (**a**) The High Manipulating Flexibility Area (the ring between R1 and R2) of the mobile manipulator when the height of the end-effector is low. (**b**) The High Manipulating Flexibility Area (the ring between R1 and R2) of the mobile manipulator when the height of the end-effector is high.

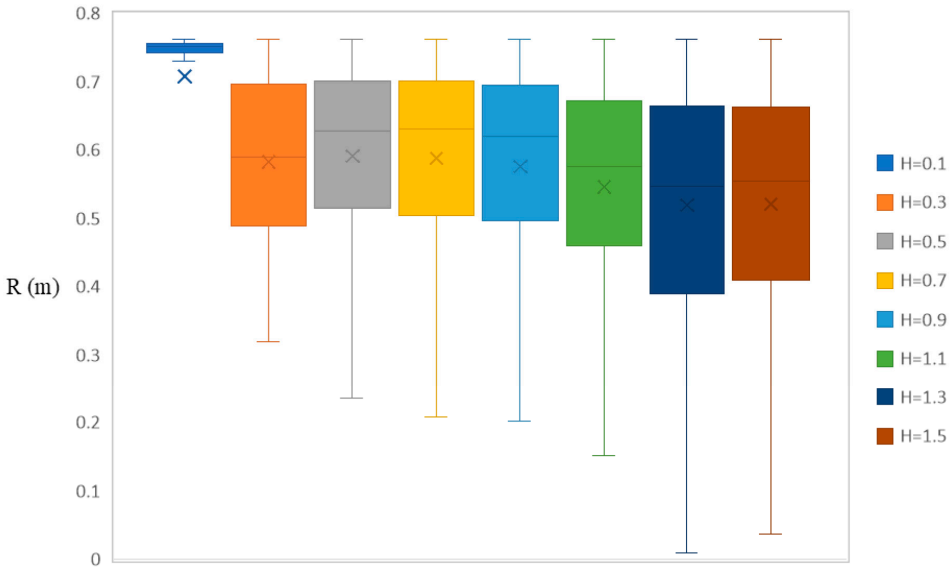


Figure 5. Valid mobile base position distribution.

When sampling the mobile base positions, sampling in the HMFA corresponding to the end-effector height should have a higher priority than sampling in the other areas. This strategy increases the probability of sampling a feasible mobile base position and reduces the number of sampling repetitions, thus increasing the efficiency of sampling.

The sampled configuration of the mobile base should be checked for collision with the environment using ISMBCONFIGVALID() in Algorithm 2. After sampling the configurations of the end-effector and the mobile base, the configuration of the manipulator could be calculated with CALMANIPCONFIG(). In this paper, we use the Kinematics and Dynamics Library (KDL) [39] to solve the inverse kinematics problem.

The SCG process is repeated until a valid configuration, checked by ISMMCONFIGVALID(), is generated for the mobile manipulator.

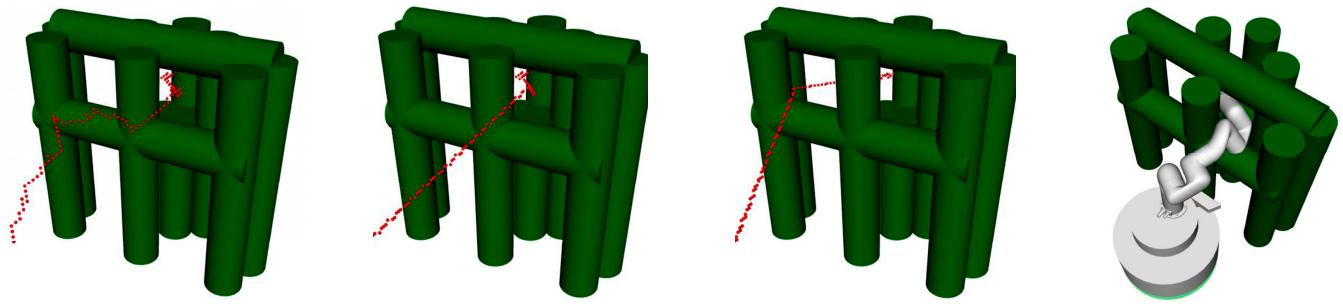
4.2. Iterative Sampling in the Neighborhood

It is extremely difficult, if not impossible, to obtain a feasible path in a complex high-dimensional space relying only on random sampling. Therefore, we proposed a method that iteratively generates a feasible path by continuously sampling in the neighborhood, starting from the searched collision-free goal configuration of the mobile manipulator. In order to improve efficiency, we use a collision-free path of the end-effector as a reference to guide the iterative generation.

As shown in Algorithm 3, CALEESTARTANDGOAL() first calculates the start and goal positions of the end-effector, and then a traditional RRT algorithm [14] is used to generate a path, as shown in Figure 6a. A cutting method that is the same as in [33] is used here, which is PRUNE() in Algorithm 3, to optimize the path generated by the RRT, and the result is shown in Figure 6b. Sometimes, there are different paths, such as shown in Figure 6c, and so it is necessary to check whether the path is suitable with the goal configuration of the mobile manipulator. If the configuration of the mobile manipulator is as shown in Figure 6d, it is better to use the path in Figure 6b as the reference path rather than the path in Figure 6c. This is checked with CALCOSTPATH() in Algorithm 3. Here, we propose a keypoint-based approach to represent the proximity of the mobile manipulator configuration to the end-effector path.

Algorithm 3: GENERATEEEPATH()

1 **Input:** q_s^{mm} , q_g^{mm}
2 **Output:** Π^{ee}
3 **while** CALCOSTPATH(Π^{ee}, q_g^{mm}) > Threshold **do**
4 q_s^{ee} , q_g^{ee} ← CALEESTARTANDGOAL(q_s^{mm} , q_g^{mm})
5 Π_{origin}^{ee} ← RRT(q_s^{ee} , q_g^{ee})
6 Π^{ee} ← PRUNE(Π_{origin}^{ee})
7 **end while**



(**a**) (**b**) (**c**) (**d**)

Figure 6. (**a**) Path generated by the traditional RRT. (**b**) Path optimized with the cutting method. (**c**) Another path for the same goal. (**d**) Goal configuration for the mobile manipulator.

As shown in Figure 7a, we select four keypoints for the mobile manipulator. It can be seen that the distance between any two adjacent keypoints is constant, regardless of the rotation of any joint. Thus, the distance between these keypoints and the path can indicate the degree of fit of the mobile manipulator configuration to the end-effector path. Here, we use the sum of the shortest distances between each keypoint K_i and the path Π_{ee} as the cost,

as shown in Figure 7b. If the cost is less than the set threshold, the mobile manipulator configuration is considered to match the end-effector path. The cost can be expressed as

$$cost_{path}(q^{mm}) = \sum \text{distance}(K_i, \Pi^{ee}). \tag{3}$$

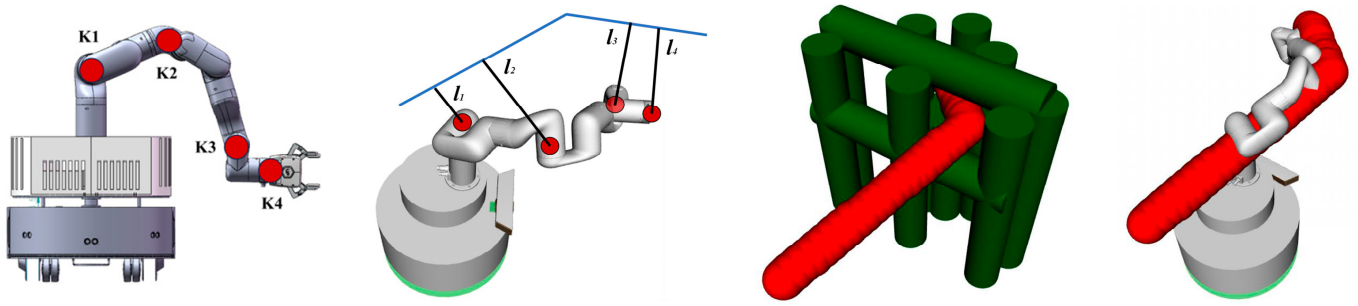


Figure 7. (**a**) Selected keypoints for the mobile manipulator. (**b**) Calculated cost between the path and the mobile manipulator. (**c**) A collision-free tunnel generated with the path. (**d**) The configuration of the manipulator along with the tunnel.

After determining the reference path, we propose an iterative sampling-based method to generate the mobile manipulator path, which samples in the neighborhood of the last configuration, causing the mobile manipulator to move along the reference path (as shown in Algorithm 4). Given q^{mm}_{now} as an initial value, SAMPLEQ() in Algorithm 4 is used to sample a configuration q^{mm}_{sample} so that each element q^{sample}_i of q^{mm}_{sample} satisfies

$$q^{sample}_i \in \dot{U}(q^{now}_i, \delta_i), \tag{4}$$

$$\dot{U}(q^{now}_i, \delta_i) = \{x | \, 0 \langle | x - q^{now}_i | < \delta_i\}, \tag{5}$$

where δ_i is the stepsize of ith element in the configuration, and q^{now}_i is the element of q^{mm}_{now}. Choosing the appropriate stepsize ensures that if both the initial value configuration and the sampled configuration are collision-free, then the motion between the two configurations can also be considered collision-free.

Since both the mobile base and the manipulator are sampled during the iterative sampling process, the generated motions are simultaneous, which facilitates the manipulation flexibility of the mobile manipulator. Here, we first sample a certain number of collision-free configurations as alternatives, and then select an optimal configuration to add to the path of the mobile manipulator. When selecting the optimal configuration, there are two main issues to consider: how well the configuration fits the reference path of the end-effector, and whether it is moving towards the target.

As shown in Figure 7c, the reference path actually represents a collision-free tunnel, and trying to move the manipulator along the tunnel (as shown in Figure 7d) is more likely to find a feasible collision-free path, thus avoiding the time wasted caused by random sampling. Thus, the cost used to evaluate the configuration consists of two parts, $cost_{path}$, which represents the degree of fit to the reference path, and $cost_{target}$, which represents the movement towards the target. The first part, $cost_{path}$, was defined in Equation (3). The second part can be expressed as

$$cost_{target}(q^{mm}) = \sum |q_i - q^t_i|, \tag{6}$$

where q_i is the ith element of the configuration q^{mm} and q_i^t is the ith element of the target configuration q_t^{mm}. Since the environment in which q_g^{mm} is located is more complex, the iterative sampling is started at q_g^{mm}, and so q_s^{mm} is the target q_t^{mm}.

The importance of these two components varies at different stages; when the mobile manipulator is close to q_g^{mm}, $cost_{path}$ is more important because the environment is complex. When the mobile manipulator is close to q_t^{mm}, there is no need to stay close to the path, and $cost_{target}$ is more important. Therefore, a linear combination of $cost_{path}$ and $cost_{target}$ is used to represent the $cost$

$$cost(q^{mm}) = k_1 * cost_{path}(q^{mm}) + k_2 * cost_{target}(q^{mm}), \tag{7}$$

where k_1 and k_2 are coefficients associated with the value of the remaining length of Π^{ee}. After each successful sample, the last element of Π^{ee} is deleted to make the mobile manipulator to move to the target configuration.

Finally, simple post-processing of the path is performed to obtain a smoothed path for the mobile manipulator.

Algorithm 4: SAMPLEINNEIGHBOURHOOD()

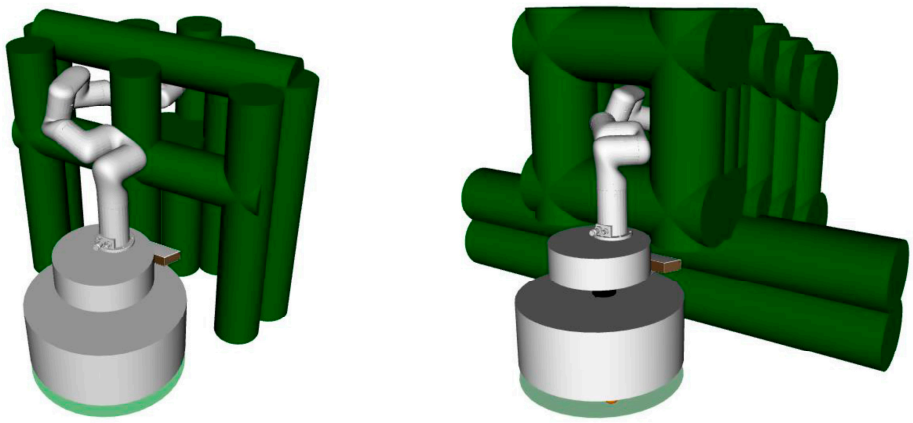
```
Input: q_now^mm, Π^ee
Output: q_rand^mm
Q is a set of q
i ← 0
while i < NumOfAlter do
    q_sample^mm ← SAMPLEQ(q_now^mm)
    if ISMMCONFIGVALID(q_sample^mm) then
        Q ← Q ∪{ q_sample^mm }
        i ← i + 1
    end if
end while
for all q ∈ Q do
    cost ← CALCOST(q, Π^ee)
    if cost < mincost then
        q_rand^mm ← q
        mincost ←cost
    end if
end for
delete the last element of Π^ee
```

5. Simulations and Discussion

5.1. Case Description

In this section, we conduct simulations on the proposed methods for different scenarios and cases. All methods are implemented in C++ and all experiments are run on an Intel i5 9400F at 2.9 GHz with 16 GB of RAM. All simulation experiments are conducted with a Robot Operating System (ROS) [40]. We consider the complex 3D scenario (scenario $C1$) shown in Figure 8a and the tunnel-like 3D scenario (scenario $C2$) shown in Figure 8b. In scenario 1, we consider four different cases with two different starting configurations, (0, 0, 0, 0, 0, 0, 0, 0) and (0.6, 0, 0, 0, 0, 3.14, 0, 0, 0), and two different goal points: (1.28, 0, 0.65) and (1.28, 0, 0.75) (denoted as $S1$, $S2$, $G1$, and $G2$, respectively). In scenario 2, we consider

two different cases, they have two starting configurations that are the same as scenario *C*1, and they have the same goal point (1.31, −0.1, 0.65) (denoted as *G*3).



(**a**) (**b**)

Figure 8. Simulation scenarios. (**a**) A complex 3D scenario. (**b**) A tunnel-like 3D scenario.

For the above cases, we apply different algorithms for the simulation, including the traditional RRT using end-effector path and inverse kinematics, separate planning for the mobile base and the manipulator and combining the paths, separate planning but planning for the manipulator when the mobile base is at the target position, the proposed hybrid sampling-based path planning, and its variant that only use $cost_{target}$ as cost when sampling in the neighborhood (denoted as RRT, IK, SP-1, SP-2, HSPP-pt, and HSPP-t). Since the RRT, IK, SP-1, and SP-2 methods also require the goal configuration of the mobile manipulator as a known condition to generate a path, we first test the performance of the proposed Sampling-based Configuration Generation (SCG) method and use it to generate the goal configuration for all methods.

5.2. Results and Discussion

First, we execute 1000 runs for each goal point to get the performance of the proposed Sampling-based Configuration Generation (SCG) method. We regard a single generation with a total number of attempts greater than 1000 and time consumed more than 1 s as a failure. The result is shown in Table 1, the success rate and time consumed in all cases are acceptable, which indicates that the method is effective.

Table 1. Results of the SCG method.

Case	Success Rate	Mean # of Samples	Mean Time Consumed [s]	Median # of Samples
C1G1	98.6%	159.64	0.48	115
C1G2	99.8%	128.99	0.14	93
C2G3	100%	92.25	0.31	63

Except for cases and success rates, failed generations are not included.

We then perform 100 runs for each algorithm in each case. *NumOfAlter* in Algorithm 4 is 100, and the threshold for determining the appropriate end-effector path is 0.1. The stepsize of the mobile base is 0.01, and the stepsize of the manipulator is 0.1. For the whole path planning process, a single path planning taking more than 120 s is regarded as a failure. Table 2 shows the success rates of the different algorithms in different cases. In all cases, the proposed HSPP-pt method exhibits the best success rates. In contrast, RRT, IK, SP-1, and SP-2 fail to find a path in the given time in any case. For the RRT and IK methods, the main cause of failure is that they require a large amount of time, while for the SP-1 and SP-2

methods, the main cause of failure is that they cannot find a collision-free path near the target configuration. For the HSPP method, different case difficulties do not significantly affect the success rate, whereas for the HSPP-t method, which only uses $cost_{target}$, different settings affect its success rate due to the fact that it may fall into a local optimum solution at a particular configuration.

Table 2. Comparison of success rates.

Case	RRT	IK	SP-1	SP-2	HSPP-t	HSPP-pt
C1S1G1	0%	0%	0%	0%	80%	97%
C1S1G2	0%	0%	0%	0%	73%	98%
C1S2G1	0%	0%	0%	0%	79%	96%
C1S2G2	0%	0%	0%	0%	72%	94%
C2S1G3	0%	0%	0%	0%	81%	99%
C2S2G3	0%	0%	0%	0%	80%	98%

Table 3 compares the average completion times of different algorithms. HSPP-t achieves faster search speeds in scenario *C*1, while HSPP-pt performs better in scenario *C*2. It can also be seen that, similar to the case of success rates, the time consumption of the HSPP-pt method is less affected by the different case settings and consumes a more even amount of time across cases, whereas the time consumption of the HSPP-t method is more affected. The time consumed by each part of the algorithm is illustrated in Figure 9. The SCG part consumes very little time and is negligible in relation to the total time consumed. The time consumed by the sampling in neighborhood for HSPP-t is much more than that of HSPP-pt, which means that $cost_{target}$ improves the performance of the process. Although the HSPP-pt method increases the time consumed by adding a reference path planning part, it is comparable to the total time consumed by the HSPP-t method, and the HSPP-pt method has better stability and higher success rate. In addition, the HSPP-pt method can further reduce the time consumed by optimizing the reference path planning.

Table 3. Comparison of average time consumption.

Case	RRT	IK	SP-1	SP-2	HSPP-t	HSPP-pt
C1S1G1	N/A	N/A	N/A	N/A	7.66 ± 4.61	11.93 ± 5.46
C1S1G2	N/A	N/A	N/A	N/A	8.25 ± 5.20	12.54 ± 6.89
C1S2G1	N/A	N/A	N/A	N/A	7.48 ± 4.47	13.42 ± 7.51
C1S2G2	N/A	N/A	N/A	N/A	8.88 ± 6.19	11.58 ± 6.23
C2S1G3	N/A	N/A	N/A	N/A	20.70 ± 12.43	13.56 ± 5.98
C2S2G3	N/A	N/A	N/A	N/A	21.11 ± 13.08	14.71 ± 9.04

Except for cases and success rates, failed generations are not included.

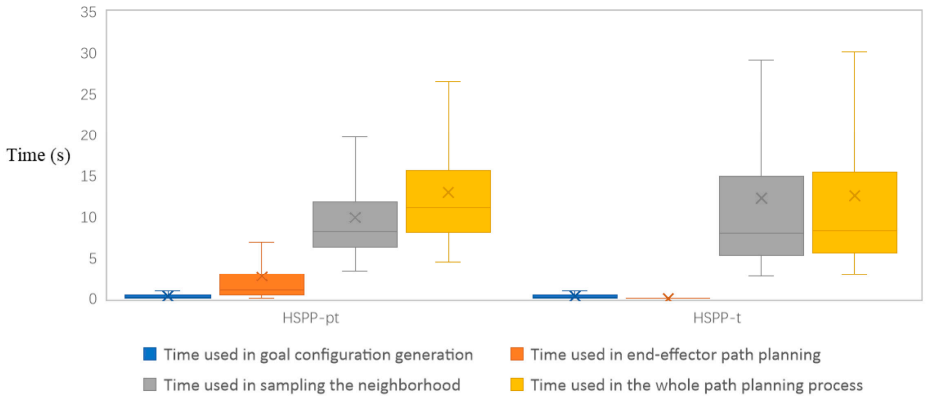


Figure 9. Comparison of time consumption in all cases.

In Figure 10, the comparison of the generated path length is shown. For HSPP-pt, the generated path length is basically positively correlated with the distance from the start configuration to the goal configuration, while for HSPP-t, in complex cases (C1S2G1, C1S2G2, and C2S2G3), the path length is increased due to the fact that the more complex the action, the greater the likelihood of it being caught in the local oscillations, which could increase the path length or even lead to a planning failure.

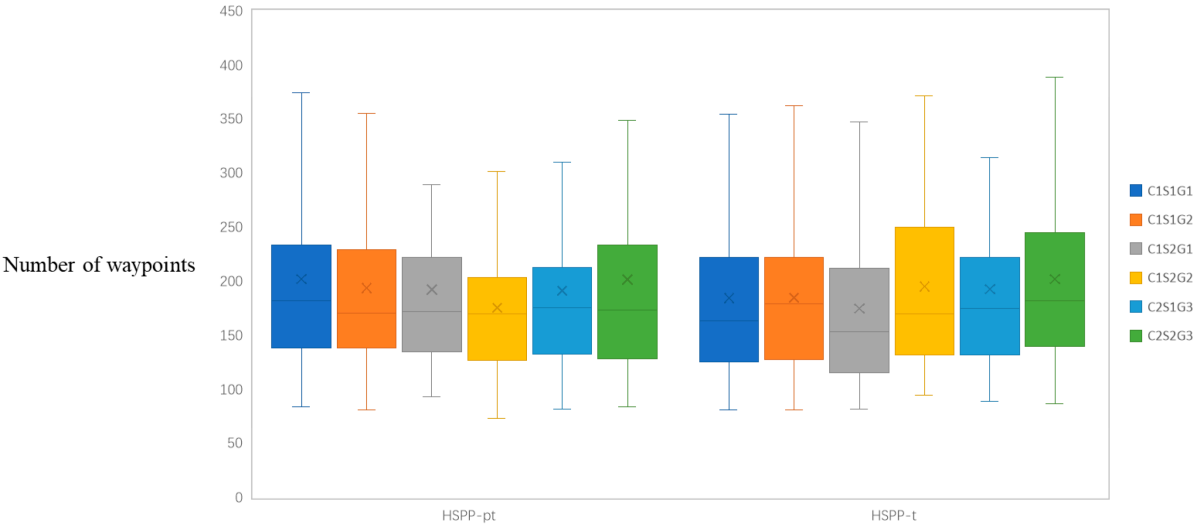


Figure 10. Comparison of the lengths of the generated paths in all cases.

6. Conclusions and Future Work

This paper proposes HSPP-pt, a hybrid sampling-based algorithm for efficiently finding a collision-free path for mobile manipulators performing pick and place tasks in confined spaces. Traditional methods are not applicable to such scenarios for various reasons. HSPP-pt incorporates an SCG method for computing a collision-free mobile manipulator target configuration based on the end-effector position and rotation constraints. Also, HSPP introduces end-effector paths as reference paths and proposes a path generation method based on sampling in the neighborhood. Although there might be a possibility that the method will fall into a local minima when sampling in the neighborhood, it avoids the complexity of random sampling in high-dimensional spaces and strikes a balance between planning efficiency and the success rate.

For future work, we plan to improve the planning methodology of the reference path to further reduce the time consumption, to establish a coordination mechanism to fully exploit the advantages of HSPP-pt and HSPP-t in different scenarios, and, finally, to further optimize the generated paths to improve the path quality. Furthermore, this method is currently only applicable to known static environments. However, there is potential for this method to be extended to dynamic obstacle avoidance with the utilization of rapid environmental perception and modelling.

Author Contributions: Conceptualization, H.C. and X.Z.; methodology, H.C. and X.Z.; software, H.C.; validation, H.C. and Y.Z.; resources, X.Z.; data curation, H.C.; writing—original draft preparation, H.C.; writing—review and editing, X.Z.; visualization, H.C.; supervision, X.Z., Y.Z. and J.Z.; project administration, J.Z.; funding acquisition, Y.Z. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by in part by the National Outstanding Youth Science Fund Project of National Natural Science Foundation of China, grant number 52025054, and in part by the Major Research Plan of the National Natural Science Foundation of China, grant number 92048301.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

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

References

1. Feng, Y.; Tian, X.; Li, T.; Jiang, Y. Measurement of mobile manipulator chassis pose change caused by suspension deformation and end-effector accuracy improvement based on multi-sensor fusion. *Robot. Auton. Syst.* **2023**, *170*, 104553. [CrossRef]
2. Ding, L.; Xia, K.; Gao, H.; Liu, G.; Deng, Z. Robust adaptive control of door opening by a mobile rescue manipulator based on unknown-force-related constraints estimation. *Robotica* **2018**, *36*, 119–140. [CrossRef]
3. Thakar, S.; Rajendran, P.; Annem, V.; Kabir, A.; Gupta, S.K. Accounting for part pose estimation uncertainties during trajectory generation for part pick-up using mobile manipulators. In Proceedings of the 2019 IEEE International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019.
4. Xiong, Y.; Peng, C.; Grimstad, L.; From, P.J.; Isler, V. Development and field evaluation of a strawberry harvesting robot with a cable-driven gripper. *Comput. Electron. Agric.* **2019**, *157*, 392–402. [CrossRef]
5. Fan, Q.; Gong, Z.; Tao, B.; Gao, Y.; Yin, Z.; Ding, H. Base position optimization of mobile manipulators for machining large complex components. *Robot. Comput. Integr. Manuf.* **2021**, *70*, 102138. [CrossRef]
6. Outón, J.L.; Villaverde, I.; Herrero, H.; Esnaola, U.; Sierra, B. Innovative Mobile Manipulator Solution for Modern Flexible Manufacturing Processes. *Sensors* **2019**, *19*, 5414. [CrossRef]
7. Kapusta, A.; Kemp, C.C. Task-centric optimization of configurations for assistive robots. *Auton. Robot.* **2019**, *43*, 2033–2054. [CrossRef]
8. Sandakalum, T.; Ang, M.H., Jr. Motion Planning for Mobile Manipulators—A Systematic Review. *Machines* **2022**, *10*, 97. [CrossRef]
9. Song, R.; Liu, Y.; Bucknall, R. Smoothed A algorithm for practical unmanned surface vehicle path planning. *Appl. Ocean Res.* **2019**, *83*, 9–20. [CrossRef]
10. Aine, S.; Likhachev, M. Truncated Incremental Search: Faster Replanning by Exploiting Suboptimality. In Proceedings of the 2013 AAAI Conference on Artificial Intelligence (AAAI), Bellevue, WA, USA, 14–18 July 2013.
11. Kabutan, R.; Nishida, T. Motion planning by T-RRT with potential function for vertical articulated robots. *Elect. Eng. Jpn.* **2018**, *204*, 34–43. [CrossRef]
12. Zhang, W.; Shan, L.; Chang, L.; Dai, Y. SVF-RRT*: A stream-based VF-RRT* for USVs path planning considering ocean currents. *IEEE Robot. Autom. Lett.* **2023**, *8*, 2413–2420. [CrossRef]
13. Qureshi, A.H.; Ayaz, Y. Intelligent bidirectional rapidly-exploring random trees for optimal motion planning in complex cluttere environments. *Robot. Auton. Syst.* **2015**, *68*, 1–11. [CrossRef]
14. Jeong, I.-B.; Lee, S.-J.; Kim, J.-H. Quick-RRT*: Triangular inequality-based implementation of RRT* with improved initial solution and convergence rate. *Expert Syst. Appl.* **2019**, *123*, 82–90. [CrossRef]
15. Kim, M.-C.; Song, J.-B. Informed RRT* with improved converging rate by adopting wrapping procedure. *Intell. Serv. Robot.* **2018**, *11*, 53–60. [CrossRef]
16. Wang, J.; Meng, M.Q.H.; Khatib, O. EB-RRT: Optimal motion planning for mobile robots. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 2063–2073. [CrossRef]
17. Khalfallah, S.; Bouallegue, M.; Bouallegue, K. Object Detection for Autonomous Logistics: A YOLOv4 Tiny Approach with ROS Integration and LOCO Dataset Evaluation. *Eng. Proc.* **2024**, *67*, 65. [CrossRef]
18. Shao, J.; Xiong, H.; Liao, J.; Song, W.; Chen, Z.; Gu, J.; Zhu, S. RRT-GoalBias and Path Smoothing Based Motion Planning of Mobile Manipulators with Obstacle Avoidance. In Proceedings of the 2021 IEEE International Conference on Real-Time Computing and Robotics (RCAR), Xining, China, 15–19 July 2021.
19. Luna, R.; Moll, M.; Badger, J.; Kavraki, L.E. A scalable motion planner for high-dimensional kinematic systems. *Int. J. Robot. Res.* **2020**, *39*, 361–388. [CrossRef]
20. Thakar, S.; Rajendran, P.; Kim, H.; Kabir, A.M.; Gupta, S.K. Accelerating Bi-Directional Sampling-Based Search for Motion Planning of Non-Holonomic Mobile Manipulators. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021.
21. Xu, C.; Wang, M.; Chi, G.; Liu, Q. An inertial neural network approach for loco-manipulation trajectory tracking of mobile robot with redundant manipulator. *Neural Netw.* **2022**, *155*, 215–223. [CrossRef]
22. Nie, J.; Wang, Y.; Mo, Y.; Miao, Z.; Jiang, Y.; Zhong, H.; Lin, J. An HQP-Based Obstacle Avoidance Control Scheme for Redundant Mobile Manipulators Under Multiple Constraints. *IEEE Trans. Ind. Electron.* **2023**, *70*, 6004–6016. [CrossRef]
23. Yu, Q.; Wang, G.; Hua, X.; Zhang, S.; Song, L.; Zhang, J.; Chen, K. Base position optimization for mobile painting robot manipulators with multiple constraints. *Robot. Comput.-Integr. Manuf.* **2018**, *54*, 56–64. [CrossRef]
24. Vazquez-Santiago, K.C.; Goh, F.; Shimada, K. Motion Planning for Kinematically Redundant Mobile Manipulators with Genetic Algorithm, Pose Interpolation, and Inverse Kinematics. In Proceedings of the 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE), Lyon, France, 23–27 August 2021.

25. Saoji, S.; Rosell, J. Flexibly configuring task and motion planning problems for mobile manipulators. In Proceedings of the 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, Austria, 8–11 September 2020.
26. Castaman, N.; Pagello, E.; Menegatti, E.; Pretto, A. Receding Horizon Task and Motion Planning in Changing Environments. *Robot. Auton. Syst.* **2021**, *145*, 103863. [CrossRef]
27. Engemann, H.; Du, S.; Kallweit, S.; Cönen, P.; Dawar, H. OMNIVIL—An Autonomous Mobile Manipulator for Flexible Production. *Sensors* **2020**, *20*, 7249. [CrossRef] [PubMed]
28. Gochev, K.; Safonova, A.; Likhachev, M. Planning with adaptive dimensionality for mobile manipulation. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA), Saint Paul, MN, USA, 14–18 May 2012.
29. Pilania, V.; Gupta, K. Mobile manipulator planning under uncertainty in unknown environments. *Int. J. Robot. Res.* **2018**, *37*, 316–339. [CrossRef]
30. Li, Q.; Mu, Y.; You, Y.; Zhang, Z.; Feng, C. A Hierarchical Motion Planning for Mobile Manipulator. *IEEJ Trans. Elect. Electron. Eng.* **2020**, *15*, 1390–1399. [CrossRef]
31. Zhang, M.; Xu, C.; Gao, F.; Cao, Y. Trajectory Optimization for 3D Shape-Changing Robots with Differential Mobile Base. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023.
32. Chen, H.; Zang, X.; Liu, Y.; Zhang, X.; Zhao, J. A Hierarchical Motion Planning Method for Mobile Manipulator. *Sensors* **2023**, *23*, 6952. [CrossRef] [PubMed]
33. Thakar, S.; Tajendran, P.; Kabir, A.M.; Gupta, S.K. Manipulator Motion Planning for Part Pickup and Transport Operations from a Moving Base. *IEEE Trans. Autom. Sci. Eng.* **2022**, *19*, 191–206. [CrossRef]
34. Jang, K.; Kim, S.; Park, J. Motion Planning of Mobile Manipulator for Navigation Including Door Traversal. *IEEE Robot. Autom. Lett.* **2023**, *8*, 4147–4154. [CrossRef]
35. Panasiuk, J.; Siwek, M.; Kaczmarek, W.; Borys, S.; Prusaczyk, P. The concept of using the mobile robot for telemechanical wires installation in pipelines. *AIP Conf. Proc.* **2018**, *2029*, 020054. [CrossRef]
36. Karaman, S.; Frazzoli, E. Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.* **2011**, *30*, 846–894. [CrossRef]
37. Wang, F.; Guadarrama-Olvera, J.R.; Cheng, G. Optimal Order Pick-and-Place of Objects in Cluttered Scene by a Mobile Manipulator. *IEEE Robot. Autom. Lett.* **2021**, *6*, 6402. [CrossRef]
38. Xie, Y.; Liu, J.; Yang, Y. Pose optimization for mobile manipulator grasping based on hybrid manipulability. *Ind. Robot. Int. J. Robot. Res. Appl.* **2023**, *51*, 134. [CrossRef]
39. Khokar, K.; Beeson, P.; Burridge, R. Implementation of KDL Inverse Kinematics Routine on the Atlas Humanoid Robot. *Procedia Comput. Sci.* **2015**, *46*, 1441–1448. [CrossRef]
40. Quigley, M.; Gerkeyy, B.; Conleyy, K.; Fausty, J.; Footey, T.; Leibsz, J.; Bergery, E.; Wheelery, R.; Ng, A. ROS: An open-source Robot Operating System. In Proceedings of the ICRA Workshop on Open-Source Software, Kobe, Japan, 17 May 2009.

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.