Motion Planning by Sampling in Subspaces of Progressively Increasing Dimension

Marios Xanthidis · **Joel M. Esposito** ·
Ioannis Rekleitis · **Jason M. O'Kane**

Abstract This paper introduces an enhancement to traditional sampling-based planners, resulting in efficiency increases for high-dimensional holonomic systems such as hyper-redundant manipulators, snake-like robots, and humanoids. Despite the performance advantages of modern sampling-based motion planners, solving high dimensional planning problems in near real-time remains a considerable challenge. The proposed enhancement to popular sampling-based planning algorithms is aimed at circumventing the exponential dependence on dimensionality, by progressively exploring lower dimensional volumes of the configuration space. Extensive experiments comparing the enhanced and traditional version of RRT, RRT-Connect, and Bidirectional T-RRT on both a planar hyper-redundant manipulator and the Baxter humanoid robot show significant acceleration, up to two orders of magnitude, on computing a solution. We also explore important implementation issues in the sampling process and discuss the limitations of this method.

Keywords Motion and Path Planning · Redundant Robots

1 Introduction

It is well known that the general motion planning problem is PSPACE-complete [30] and that the runtime of even the best known exact algorithm is exponential in the dimension of the configuration space [4]. While sampling-based motion planning algorithms such as rapidly exploring random trees (RRTs) [22] and probabilistic roadmaps (PRMs) [16] are

Marios Xanthidis
Computer Science and Engineering Department, University of South Carolina
E-mail: mariosx@email.sc.edu

Joel M. Esposito
Robotics and Control Engineering Department, United States Naval Academy
E-mail: esposito@usna.edu

Ioannis Rekleitis
Computer Science and Engineering Department, University of South Carolina
E-mail: yiannisr@cse.sc.edu

Jason M. O'Kane
Computer Science and Engineering Department, University of South Carolina
E-mail: jokane@cse.sc.edu

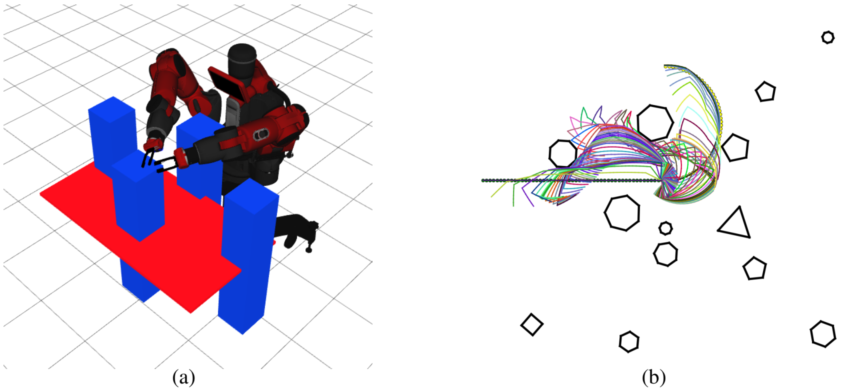

(a) (b)

Fig. 1: (a) A Baxter robot in a heavily cluttered environment. (b) A 50-DoF kinematic chain moving amidst obstacles.

able to avoid explicit reconstruction of the free configuration space, which Canny's algorithm [4] relies upon, they cannot avoid the underlying curse of dimensionality. Indeed, Esposito's Conditional Density Growth Model (CDGM) for RRTs [8] predicts that the expected number of samples required for an RRT to explore a certain volume fraction with a given probability grows exponentially with the Configuration Space dimensionality. For this reason, even though modern sampling-based planners have exhibited dramatic improvements over the original RRT and PRM techniques, computing real time solutions for systems with 10 or more degrees of freedom such as hyper-redundant manipulators, snake-like robots, or humanoids, remains a significant challenge. Moreover, this challenge prohibits real time applications of such high dimensional systems even when only simple kinematic constraints need to be satisfied, such as producing trajectories that avoid obstacles. More complex planning problems that require producing paths that satisfy dynamic constraints, or guarantees optimality for some criteria, such as energy efficiency, inertia reduction, etc, are open problems that have yet to be addressed using fast, real-time efficient, methods. Rapidly generating, even highly sub-optimal, solutions would provide great utility for planners such as InformedRRT* [9] and Batch Informed Trees [10] that rely on initial paths to reduce the search space, and to fast path optimization-based planners such as TrajOpt [31]. In general, techniques such as the above require initial valid paths, in order to avoid excessive computational costs for large search spaces, and/or addressing local-minimum issues that are difficult to overcome.

This study addresses the path planning problem in high-dimensional configuration spaces for holonomic systems. The main contribution of this paper is introducing an algorithmic enhancement that:

- Substantially accelerates the discovery of solutions for systems with many DoFs, up to two orders of magnitude compared to the original planners.
- Inherits the fundamental properties of the original planners, such as completeness or the local connection strategy.
- Is general enough to be applied to a large variety of holonomic robotic systems.
- Is scalable and linear with no explicit limitations on the number of dimensions.
- Requires no user-defined mapping functions, although simple generic policies could be developed to provide solutions even faster.

The proposed enhancement, rather than directly enabling planning for systems subject to desired constraints, aims to produce solutions that could be utilized quickly by path optimization methods that require an initial path to produce a desired solution such as CHOMP [29], STOMP [15], or Trajopt [31] variants. More specifically, assuming a feasible initial solution is provided fast enough —which is typically the most time consuming part of the process— path optimization techniques could be deployed to optimize quickly the given path into a desired path that satisfies dynamic or energy efficiency constrains in real-time.

Our approach is based on the observation that, for many redundant systems, often only a subset of the kinematic abilities are needed to complete a task [43]. Therefore, we propose beginning the search in a *lower dimensional subspace* of the configuration space $\mathscr{C}$ in the hopes that a simple solution will be found quickly. An important property of these subspaces is that a solution lying entirely inside these subspaces should be feasible, in the absence of obstacles, so the initial and goal configurations must lie inside every subspace. The proposed method, by construction, generates subspaces that satisfy this constraint. After a certain number of samples are generated, if no solution is found, we increase the dimension of the search subspace and continue sampling in the larger subspace. We repeat this process until a solution is found. In the worst case, the search expands to include the full dimensional configuration space — making the completeness properties identical to the original version of the planer.

To evaluate this approach, we modified three well established planners — RRT [22], RRT-Connect [20], and Bidirectional T-RRT [13] — to produce RRT$^+$, RRT$^+$-Connect, and Bidirectional T-RRT$^+$, with the $^+$ symbol indicating that the planners are enhanced using the idea described above. All three planners were compared to the original planners and to KPIECE [34] and STRIDE [11]. These planners were tested on a planar hyper-redundant arm, varying from 12 to 30 DoFs, and on a simulated Baxter humanoid robot, both shown in Figure 1, utilizing OMPL [36] and the MoveIt! framework [7]. In many cases, our experiments indicate that a solution is found much faster using the proposed approach and the run time appears to be less sensitive to the full dimension of the configuration space. For example, our enhanced version of Bidirectional T-RRT [13] found solutions for the Baxter robot 200 times faster than the original planner, outperformed the other planners we tested by a large margin. Perhaps surprisingly, the proposed method does not seem to trade-off path length for speed; in most cases, path quality was slightly improved.

The remainder of this paper is structured as follows. Section 2 reviews other approaches for motion planning for high-dimensional systems. Section 3 provides technical details of the enhancement, considering important issues such as how the search subspaces are selected, how to generate the samples in those subspaces, and when to expand the search dimension. Section 4 presents our experiments applying the enhancement to three well established planners: RRT [22], RRT-Connect [20], and Bidirectional T-RRT [13]. Finally, Section 5 concludes with lessons learned and a discussion of future work.

2 Related work

The problem of motion planning has been proven to be PSPACE-hard [30]. During the late 1990's, sampling-based methods were introduced and shown to be capable of solving challenging motion planning problems, but without guarantees of finding the solution in finite time [23]. The two most prominent representatives of those algorithms are probabilistic roadmaps (PRMs) by Kavraki *et al.* [16], which are useful for multiple queries in stable environments, and RRTs by LaValle [22], that are more suitable for single query applications. Two other variations of RRTs were used in this study: RRT-Connect by Kuffner and LaValle [20], which extends the tree more aggressively in each iteration, and T-RRT by Jaillet *et al.* [13], which plans efficiently in costmaps.

Although the performance of these techniques can be affected substantially by the number of degrees of freedom of the system, some studies have used them successfully for a variety of high dimensional robotic systems including hyper-redundant arms, mobile manipulators, multi-robot systems, and humanoid robots.

A method that uses PRMs and finds collision-free paths for hyper-redundant arms was presented by Park *et al.* [27]. Other studies use RRTs for motion planning of redundant manipulators, such as the work of Bertram *et al.* [3], which solves the inverse kinematics in a novel way. Weghe *et al.* [41] apply RRTs to redundant manipulators without the need to solve the inverse kinematics of the system. A study by Qian and Rahmani [28] combines RRTs and inverse kinematics in a hybrid algorithm that drives the expansion of RRTs by the Jacobian pseudo-inverse.

Other works have applied RRTs to mobile manipulators. Vannoy *et al.* [38] propose an efficient and flexible algorithm for operating in dynamic environments. The work of Berenson *et al.* [2] provides an application of their technique to a 10-DoF mobile manipulator.

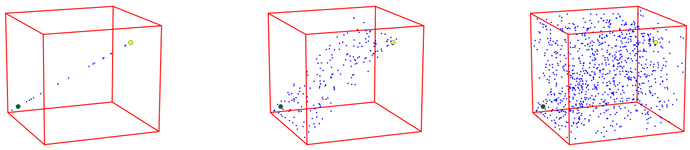
For multi-robot systems, many sampling-based algorithms have been proposed. The study of van den Berg and Overmars [37] uses a PRM and presents a prioritized technique for motion planning of multiple robots. Other studies use RRT-based algorithms such as the study by Carpin and Pagello [5] which introduced the idea of having multiple parallel RRTs for multi-robot systems. The work of Wagner [40] plans for every robot individually and, if needed, coordinates the motion in higher dimensional spaces. Other studies propose efficient solutions using a single RRT [26,33].

Sampling-based planning algorithms have been applied to humanoid robots in Kuffner *et al.* [21,19]. Other studies, such as the work of Liu *et al.* [25], use RRTs for solving the step selection problem for humanoid robots.

Regardless of the application, several studies explicitly attempt to reduce the dependence on dimensionality in sampling-based motion planning. Vernaza and Lee [39] extract structural symmetries in order to reduce the apparent dimension, providing near-optimal solutions but only for known environments where the cost function is stable. Yoshida [45] tries to sample in ways that exploit the redundancy of a humanoid. Wells and Plaku [42] reduce the dimensionality for 2-D hyper-redundant manipulators by modeling the end-effector as a single mobile robot, and the other links as trailers being pulled. While there are many specialized approaches to reducing dimension, few of these apply to general robot systems.

Planning in high dimensional spaces has also been done with path optimization techniques such as CHOMP [29], STOMP [15], and Trajopt [31]. These techniques can produce high quality paths and deal with narrow passages by optimizing an initial trajectory that could be highly infeasible. But often the optimization depends on the initial path and can produce infeasible solutions due to local minima issues. Thus these techniques very often are used as a post-processing step on the result from a time consuming sampling-based motion planner, whose overhead is the focus of our study.

Very recent works propose the application of machine learning techniques to drive the tree growth or produce heuristics so a solution will be found faster. For example, the work of Zha *et al.* [46] proposed a framework based on a Gaussian Mixture Model with reported $30\% - 200\%$ acceleration on the computation speed, in comparison to the original planners. On the other hand, Klamt and Behnke [18] proposed an A* approach with a learned heuristic produced from a Convolutional Neural Network to plan paths efficiently for a high dimensional robot.



(a) (b) (c)

Fig. 2: Sampling in one, two and three dimensions in $\mathscr{C}$. Red lines indicate the boundaries of $\mathscr{C}$, the yellow dots indicates q_{init}, and the green dots indicate q_{goal}.

More relevant to our work are planners that attempt to focus sampling in the relevant regions of the configuration space. Gipson *et al.* developed STRIDE [11], which samples non-uniformly with a bias toward unexplored areas of the configuration space. Yershova *et al.* [44] proposed an approach to focus sampling in the most relevant regions. KPIECE [34] by Şucan and Kavraki uses random 2D and 3D projections to estimate the coverage of Configuration Space $\mathscr{C}$ where the density of samples is lower, provided a fast planner for high-dimensional configuration spaces. Gochev *et al.* [12] proposed a motion planner that decreases the effective dimensionality by recreating a configuration space with locally adaptive dimensionality. Kim *et al.* [17] present an RRT-based algorithm for articulated robots that reduces the dimensionality of the problem by projecting each sample into subspaces that are defined by a metric. Shkolnik and Tedrake [32] plan for highly redundant manipulators in the low dimensional task space with the use of Jacobian transpose and Voronoi bias. As shown by Şucan and Kavraki [35], even random projections of the configuration space can provide good estimates for its coverage. Recent work of Chamzas *et al.* [6] proposes a novel framework for experience-based sampling, where it decomposes the workspace to local primitives which are stored in a database and on the planning phase corresponding local-planners are synthesized to bias sampling. Lastly, the work of Bayazit *et al.* [1] where a PRM was used to plan in subspaces of the configuration space, creates paths that solve an easier problem than the original, by shrinking the obstacles and then iteratively optimize the solution until the solution becomes valid. Other examples include [37,40,39,11,12,45,17, 42].

The observation that high-dimensional systems are often overactuated has been very recently highlighted. Lee *et al.* [24] were able to find efficient solutions by simplifying the kinematic abilities of humanoids, while the method proposed by Jia *et al.* [14] provided a method for solving fast motion planning problems with dynamic constraints by remapping the problem into a grid based search.

This paper introduces the idea of iteratively searching in lower-dimensional subspaces, and emphasizes the potential of using such an approach for efficient motion planning on arbitrary hyper-redundant systems. Unlike previous works, this approach tries to find paths that are not only confined entirely to a subspace but also in a subspace in which a solution can exist, since the initial and goal configurations are part of the subspace. Contrary to the work of Bayazit *et al.* [1], the approach searches in subspaces that are strictly lower-dimensional, leading to faster computation of the solution.

The main advantage of our approach, excluding the major acceleration on the computation of a solution in comparison to the state-of-the-art, is that it can be adapted easily to enhance many of the existing algorithms. The proposed enhancement does not use approaches that do not scale well with the dimensionality, such as grids, and more importantly, no user

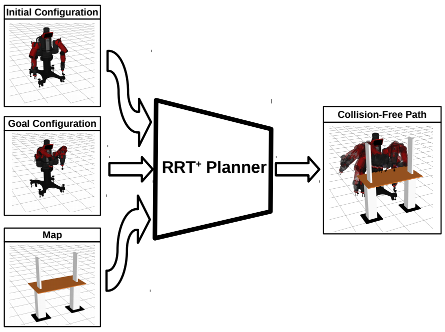


Fig. 3: An overview of the usage of the proposed planners. The initial and goal configurations along with a map, shown on the left, are the inputs that the RRT$^+$ planners utilize to produce a collision free path, shown on the right.

defined mapping functions are needed, although if applied they could provide even better performance.

3 Methodology

3.1 Problem Statement

The problem the proposed algorithm is solving is motion planning, and a simple outline is shown in Figure 3. Formally, let $\mathscr{C}$ denote a configuration space with n degrees of freedom, partitioned into free space $\mathscr{C}_{\text{free}}$ and obstacle space $\mathscr{C}_{\text{obs}}$ with $\mathscr{C} = \mathscr{C}_{\text{free}} \cup \mathscr{C}_{\text{obs}}$. The obstacle space $\mathscr{C}_{\text{obs}}$ is not explicitly represented, but instead can be queried using collision checks on single configurations or short path segments. Given initial and goal configurations $q_{\text{init}}, q_{\text{goal}} \in \mathscr{C}_{\text{free}}$, we would like to find a continuous path within $\mathscr{C}_{\text{free}}$ from q_{init} to q_{goal}.

For purposes of sampling, we assume that each degree of freedom in $\mathscr{C}$ is parameterized as an interval subset of $\mathbb{R}$, so that

$$\mathscr{C} = \left[c_1^{(\min)}, c_1^{(\max)}\right] \times \cdots \times \left[c_n^{(\min)}, c_n^{(\max)}\right] \subseteq \mathbb{R}^n. \tag{1}$$

Note that we treat $\mathscr{C}$ as Euclidean only in the context of sampling; other operations such as distance calculations and the generation of local path segments utilize identifications on the boundary of $\mathscr{C}$ as appropriate for the topology. The focus of this work is on holonomic systems. Transitions are allowed from a state A$\in \mathscr{C}$ to another state B$\in \mathscr{C}$ as long as no collisions occur.

3.2 Planning in Subspaces

The proposed method is based on searching for a solution in lower dimensional subspaces of $\mathscr{C}$, in expectation that such a path might be found faster than searching in the entirety of $\mathscr{C}$. The underlying idea is to exploit the redundancy of each system for each problem. To achieve this, the algorithm starts searching in the unique linear 1-dimensional subspace of $\mathscr{C}$ that contains q_{init} and q_{goal}. If this search fails, the planner expands its search subspace by one dimension. This process continues iteratively until the planner finds a path, or until

Algorithm 1: RRT$^+$

Input : A configuration space $\mathscr{C}$, an initial configuration q_{init}, and a goal configuration q_{goal}.
Output: RRT graph G

1 $\mathscr{C}_{\text{sub}} \leftarrow$ 1-d subspace of $\mathscr{C}$, through q_{init} and q_{goal}
2 G.init(q_{init})
3 **while** *True* **do**
4     $q_{\text{rand}} \leftarrow$ sample drawn from $\mathscr{C}_{\text{sub}}$
5     $q_{\text{near}} \leftarrow$NearestVertex$(q_{\text{rand}}, G)$
6     $q_{\text{new}} \leftarrow$NewConf$(q_{\text{near}}, q_{\text{rand}})$
7   G.AddVertex(q_{new})
8   G.AddEdge(q_{near}, q_{new})
9   **if** *done searching* $\mathscr{C}_{sub}$ **then**
10   **if** dim$(\mathscr{C}_{sub}) <$ dim$(\mathscr{C})$ **then**
11   Expand $\mathscr{C}_{\text{sub}}$ by one dimension.
12   **else**
13   **return** G
14   **end**
15   **end**
16 **end**

it searches in all of $\mathscr{C}$. In each subsearch, the tree structure created in lower dimensions is kept and expanded in subsequent stages.

Algorithm 1 summarizes the general approach as applied to RRT. Lines 2 through 8 are encapsulating the typical RRT algorithm [22]. The rest are showing the proposed modifications, with emphasis on line 4, which calls the novel sampler. These enhancements are implementing the following behavior: The planner starts optimistically by searching in one dimension, along the line passing through q_{init} and q_{goal}. If this search fails to find a path—a certainty, unless there are no obstacles between q_{init} and q_{goal}—the search expands to a planar subspace that includes q_{init} and q_{goal}, then to a 3D flat,[1] and so on until, in the worst case, the algorithm eventually searches all of $\mathscr{C}$; see Figure 2.

The description in Algorithm 1 leaves three important elements unspecified. First, the algorithm needs a method for selecting and representing the subspace $\mathscr{C}_{\text{sub}}$ (Lines 1 and 11). Second, a method is required for sampling from this subspace (Line 4). Third, the conditions that must be met before moving to the next subsearch must be defined (Line 9). The choices explored in this study are described in the next sections.

It is worth noticing that the enhanced planners inherit the transition method of the original planners (Lines 6 and 8), since only the sampling stage is altered. Thus the collision checking and the transition functions are also inherited directly from the original planners, and the proposed enhancement does not effect those elements in any way.

3.3 Representing and Sampling from Subspaces

The central idea is to search for solutions in subspaces of progressively higher dimensions. The primary constraint on these subspaces is that they must contain both q_{init} and q_{goal}. Subspaces that violate this constraint cannot, of course, contain a path connecting q_{init} to q_{goal}. In general, the algorithm's selections for $\mathscr{C}_{\text{sub}}$ should ideally be directed by the likelihood that a solution will exist fully within $\mathscr{C}_{\text{sub}}$. However, it is not clear how this likelihood should be computed. Instead, we consider a simple random technique that is quite effective, especially for highly-redundant systems.

[1] We use the term *flat* to refer to a subset of $\mathbb{R}^n$ congruent to some lower-dimensional Euclidean space.

Algorithm 2: Prioritized Sampler

Input : Initial configuration q_{init}, goal configuration q_{goal}, set of constrained DoFs P_{con}, boundaries of the random number r r_{min}, r_{max}.
Output: Sample configuration q

$q \leftarrow$ random point in $\mathscr{C}$ along line L from q_{init} to q_{goal} with r_{min} and r_{max}
for $i \in 1,\ldots,n$ **do**
  if $i \notin P_{con}$ **then**
    $q[i] \leftarrow \text{Random}(0, 1) * (c_i^{(max)} - c_i^{(min)}) + c_i^{(min)}$
  end
end
return q

The choice that we investigate in this paper—one that trades generality for simplicity—is the prioritized release of the degrees of freedom. The idea is that initially all the DoFs will be constrained to vary linearly together, so that the available subspace $\mathscr{C}_{\text{sub}}$ is the single line connecting q_{init} to q_{goal}. Each time the search is ready to expand to a higher dimensional subspace, one DoF is chosen to be released. For the released DoFs, instead of enforcing the above-mentioned linear constraints, they take values from their range, independently from the other DoFs. More formally, given a set $P_{\text{con}} \subseteq \{1,\ldots,n\}$ of DoFs to be constrained, we can form $\mathscr{C}_{\text{sub}}$ by constraining the DoFs in P_{con} to form a line passing from q_{init} and q_{goal} and allowing the remaining DoF to vary freely. In each step, one randomly-selected DoF from P_{con} is removed, thus increasing the dimensionality of $\mathscr{C}_{\text{sub}}$.

Next, the algorithm requires a technique for drawing samples from $\mathscr{C}_{\text{sub}}$. The sampling uses a very efficient linear time method that initially generates a sample within $\mathscr{C}$ along the line between q_{init} and q_{goal} by selecting a random scalar r and applying:

$$q_{\text{rand}}^{(i)} = (q_{\text{goal}}^{(i)} - q_{\text{init}}^{(i)})r + q_{\text{init}}^{(i)}. \tag{2}$$

The algorithm then modifies q_{rand} by inserting, for each DoF not in P_{con}, a different random value within the range for that dimension; see Algorithm 2. More specifically, the sampler starts by having all the DoFs constrained, thus a random sample is generated along line L, and no value is modified in the loop that follows. When a DoF has been removed from P_{con}, then a random sample is selected on the 1-D line L, but the corresponding value of the released DoF is altered with a random value from its entire range. Thus, the sample will be drawn from a 2-D plane extended by the corresponding basis vector of that dimension. The process continues by releasing an additional DoF at every iteration, until a solution is found or all DoFs are released and planning proceeds on the complete configuration space, as with the original planner.

To ensure that the samples along the line L between q_{init} and q_{goal} remain within $\mathscr{C}$, we compute r_{min} and r_{max} using Algorithm 3 and select a scalar r randomly from the interval $[r_{min}, r_{max}]$. The *ComputeBoundaryValues* function calculates the line passing from q_{init} and q_{goal} (lines 1 and 2), finds the intersections of the line with all the different $c_i^{(min)}$ and $c_i^{(max)}$ flats (lines 3 through 5), and returns the limits r_{min} and r_{max} (line 16). When r takes the value r_{min} or r_{max} then the resulting sample is one of the two intersection points of the line and the boundaries of $\mathscr{C}$ (lines 6 through 12). The *ComputeBoundaryValues* function is called only once before the planning loop.

The prioritized method provides an efficient and easy way to sample from projections of arbitrary dimensionality, while, as stated before, it provides valuable understanding that may be utilized while developing new prioritization policies for each robotic system.

Algorithm 3: ComputeBoundaryValues

Input : Initial configuration q_{init}, goal configuration q_{goal}, dimensionality of configuration space n, limits of configuration space $c^{(min)}$, $c^{(max)}$.
Output: Minimum value of scalar r_{min}, maximum value of scalar r_{max}

$D \leftarrow q_{\text{goal}} - q_{\text{init}}$
$L = Dt + q_{init}$
for $i \leftarrow 1$ **to** n **do**
    for c **in** $\{c_i^{(min)}, c_i^{(max)}\}$ **do**
        Find the intersection $p = Dt_p + q_{init}$ of L and c
        if $p \in \mathscr{C}$ **then**
            Find t_p so $p = Dt_p + q_{init}$
            if $(t_p \leq 0)$ **then**
                $r_{min} \leftarrow t_p$
            else
                $r_{max} \leftarrow t_p$
            end
        end
    end
end
return r_{min}, r_{max}.

3.4 Terminating the subsearches

The only remaining detail to be discussed is how long the search in each subspace should continue. A set of timeouts $\{t_1, t_2, \ldots, t_n\}$ is generated in which t_i corresponds to the amount of time spent in the i^{th} iteration. Ideally, the planner should stop searching in subspaces that seem unlikely to provide a solution. For simplicity, in this paper we precompute the timeouts by assuming that the t_i follows a geometric progression. The idea is to exponentially increase the number of samples in each successive subsearch, acknowledging the need for more samples in higher dimensions.

The proposed approach uses two different parameters specified by the user. The first parameter T is the timeout for the entire algorithm. The second parameter $\alpha > 1$ is a factor describing a constant ratio of the runtime between successive subsearches. The total time T available to the algorithm can be expressed in terms of α and a base time t_0:

$$T = \sum_{i=1}^{n} t_0 \alpha^i. \tag{3}$$

Solving for t_0 we obtain:

$$t_0 = \frac{\alpha - 1}{\alpha(\alpha^n - 1)} T. \tag{4}$$

Using this t_0 every t_i can be computed as:

$$t_i = \alpha t_{i-1}. \tag{5}$$

In this study, good performance was achieved when α took a value between 1 and 2 and T was scaled linearly with the dimensions of the configuration space. Moreover, in the experiments it is shown that the T parameter does not negatively affect the performance beyond a certain value. If the T value is too small, the search in the lower subspaces terminates early and the value of the proposed enhancement is reduced, since the original algorithm's sampling in the entire $\mathscr{C}$ is applied.Not sure why

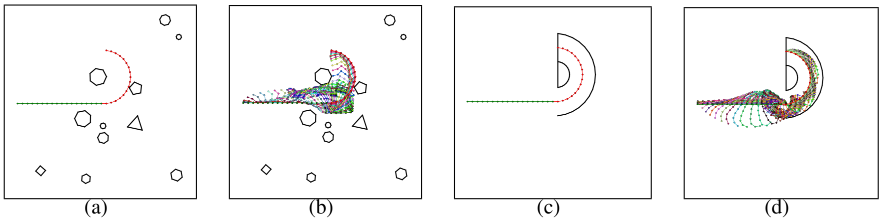


Fig. 4: The two different environments, called Random Cluttered and Horn [11]. In (a) and (c) the two planning problems are presented. The red chain indicates the initial configuration (q_{init}) and the green chain represents the goal configuration (q_{goal}). In (b) and (d), solutions for the two environments, produced by RRT^{+}-Connect, are shown using a random color-palette.

4 Experiments

For the experiments three new planners were developed using the OMPL framework [36]. These planners work by applying the proposed technique to three RRT variants: (1) RRT [22] with default goal bias of 0.05, (2) RRT-Connect [20], and (3) the BiT-RRT [13] by assuming a uniform costmap. The BiT-RRT is intended to test the effect of our method on a powerful costmap planner.

4.1 Experiments with a 2D hyper-redundant manipulator

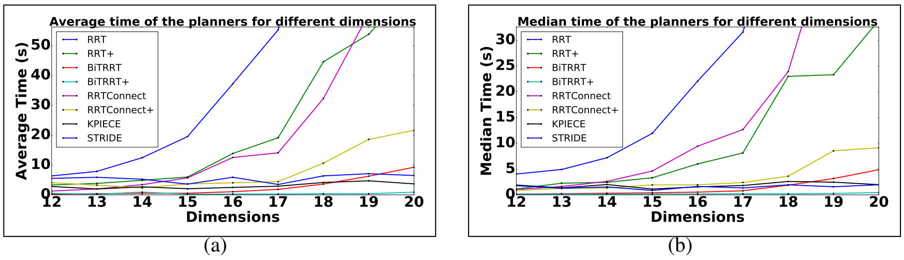
In order to test the ability of the new planners to adapt to different problems, the prioritization of the degrees of freedom was chosen randomly for each run. We demonstrate that, given enough redundancy, even a random prioritization provides results much faster (up to two orders of magnitude). A computer with 6^{th} Generation Intel Core i7-6500U Processor (4MB Cache, up to 3.10 GHz) and 16GB of DDR3L (1600MHz) RAM was used.

OMPL's standard example of a 2D hyper-redundant manipulator, created for STRIDE [11], was used in order to test and compare the enhanced planners in a standard way. Two different environments were tested 100 times each for a kinematic chain with varying degrees of freedom (between 12 to 20): A Cluttered Random environment and a Horn environment; see Figure 4. The initial and goal configurations were the same as shown in Figure 4, so a qualitative comparison in problems with different redundancies could be done.

In all cases, the enhanced versions of each planner were faster and in most cases significantly faster than the original ones. The average and the median times for the Random Cluttered environment are presented in Figure 5, and in Figure 6 the same for the Horn environment. As can be seen, as the dimensions of the configuration space increase the proposed enhancement outperforms the original algorithm (lower is better).

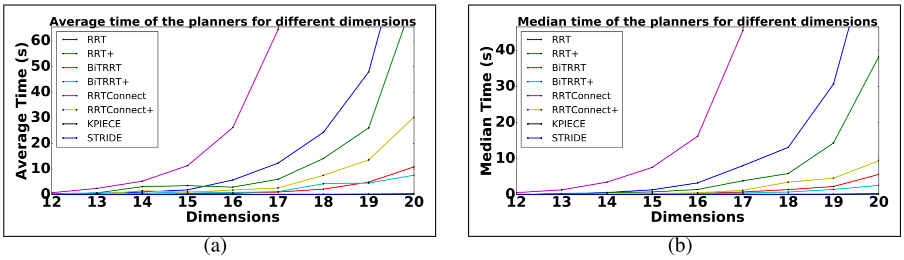
The BiT-RRT^{+} not only outperformed the BiT-RRT by a wide margin but also outperformed all the other planners using uniform sampling. Additionally, it provided competitive results for robust planners with biased sampling such as KPIECE [34] and STRIDE [11], which as expected perform much better in less redundant environments. Interestingly, for each problem, the single fastest solution across all trials was generated by BiT-RRT^{+}. This suggests that a good choice of prioritization may give results faster in a consistent way.

Additional experiments with the fastest planners were performed by varying the number of degrees of freedom between 12 and 30 for the Cluttered environment. The performance of



(a) (b)

Fig. 5: The average (a) and the median (b) time for the Random Cluttered environment.



(a) (b)

Fig. 6: The average (a) and the median (b) time for the Horn environment.

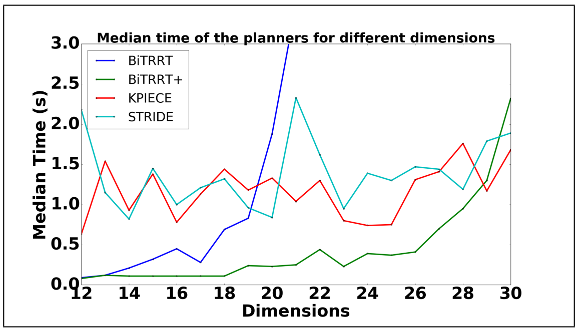


Fig. 7: Comparison of the median time for the Cluttered Random environment from 12 to 30 dimensions of BiT-RRT and BiT-RRT$^+$, KPIECE and STRIDE. Interestingly, BiT-RRT$^+$ is more than 200 times faster than BiT-RRT for 30 DoF.

the BiT-RRT$^+$ in Figure 7 demonstrated superior performance in increased dimensionality. Also the BiT-RRT$^+$ provided the fastest results among all the planners. Even with random prioritization only after the 29 dimensions due to insufficient prioritizations the median runtime of BiT-RRT$^+$ becomes slitghly larger than KPIECE and STRIDE.

As shown in Figure 8, our method does not significantly affect the path quality as measured by path length.

We also demonstrate that the performance of the planners is relatively insensitive to T across a wide range, as shown in Figure 9. In order to study the sensitivity by including also the outliers, T was used for Equation 4, and in case a solution was not found in T time, the planners continued sampling uniformly in $\mathscr{C}$ until a solution was found. Although a good tuning may positively affect the efficiency, no clear trend is observed on the efficiency of the enhanced planner as a function of T.

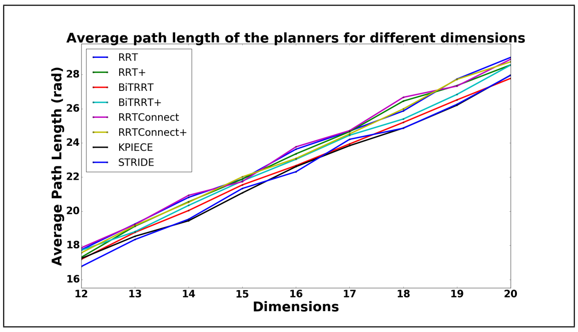


Fig. 8: The average path length for the Horn environment after the standard path-simplification method of OMPL.

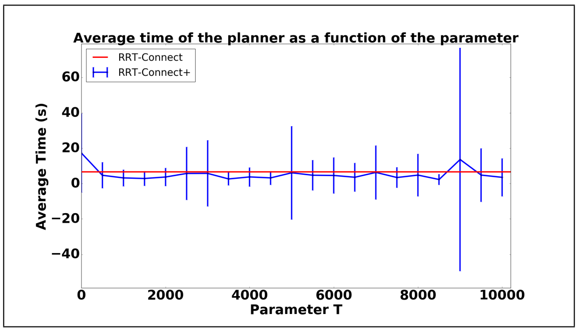


Fig. 9: Sensitivity to T from 0 to 10000 with step 500, given $\alpha = 1.6$ for 200 runs per value of RRT$^+$-Connect in the Random Cluttered environment for 15 degrees of freedom.

	Success rate (%)	Average (s)	Median (s)
BiT-RRT$^+$	**66**	0.29±0.25	**0.44**
KPIECE	34	0.19±0.26	0.65
STRIDE	39	0.24±0.27	0.58

Table 1: Success rate, average and median of 100 trials for the 50-DoF kinematic chain, with a time-out of 1 second. BiT-RRT$^+$ is close to twice more successful than KPIECE [34] and 1.7 times more successful than STRIDE [11], with shorter median computation time for the solution.

Lastly, we demonstrated further the capability of the enhancement to solve very challenging problems with many dimensions in real-time, by simply using random prioritizations. BiT-RRT$^+$ was benchmarked along with KPIECE [34] and STRIDE [11] on a 50-DoF manipulator in the Cluttered environment (Figure 1-b) with a small time-out of only 1 second. As shown in the results (Table 1) BiT-RRT$^+$ is twice as effective for solving fast problems of such high-dimensionality, with a shorter median time than the other two state-of-art techniques, illustrating the potential of the method on generating paths for real-time applications. Note that the original algorithm, BiT-RRT, is not reported, since it was not able to successfully produce any solution within the given time-out.

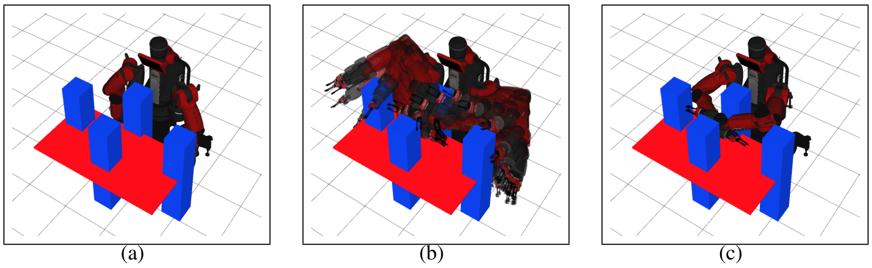


Fig. 10: The initial (a) and goal (c) configuration, along with the path (b) used in the experiments with the Baxter. The table is indicated with red, and the four pillars are indicated with blue.

4.2 Experiments with Baxter

Experiments are presented for the Baxter humanoid robot (Figure 1) with 14 degrees of freedom using the OMPL [36] and MoveIt! framework [7] with an Intel i7-7700 8-core processor (3.6GHz), and 32 GiB RAM.

Given knowledge of the system, instead of choosing the prioritizations randomly, a generic task independent policy was used to show that even naive policies can eliminate the outliers observed in the earlier experiments and lead to superior performance. The policy was giving priority to the joints closer to the base.

We tested the enhanced planners in a cluttered workspace, which consisted by a table and four parallel pillars. As shown in Figure 10, Baxter starts with the manipulators in the relaxed position below the table and the goal is to reach the configuration that both arms fit between the pillars. Moreover, in order to make the problem more challenging for the enhanced planners, one of the manipulators should end up above the other, reducing even more the redundancy of the problem and forcing our planners to explore subspaces with high dimensionality. The RRT$^+$, RRT$^+$-Connect, BiT-RRT$^+$ were compared with their original versions, and also with KPIECE and a bidirectional version of KPIECE provided in OMPL, called BiKPIECE, for 100 trials. A timeout of 60 seconds was used for each run.

As shown in Figure 11, each enhanced planner outperformed the corresponding non-enhanced planner. In particular, RRT$^+$-Connect and BiTRRT$^+$ provided solutions much faster than BiKPIECE and KPIECE. Non-bidirectional planners had difficulty finding a solution, showing the difficulty of the problem close to the goal region. The enhanced planners, even when they were failing in the case of RRT$^+$, produced solutions very quickly (in less than 5 seconds) showing the advantage of sampling in lower-dimensional subspaces. Similarly to the experiments presented in the previous section, the fastest planner among all was the BiTRRT$^+$. Most of the solutions from RRT$^+$- were found in an 8-dimensional subspace, and from RRT$^+$-Connect and BiTRRT$^+$ in a 6-dimensional subspace.

5 Conclusion

We proposed a novel method for accelerating motion planning in high dimensional configuration spaces by sampling in subspaces of progressively increasing dimension. The method provides, on average, solutions up to two orders of magnitude than the original RRT-based methods without a negative effect on the path quality.

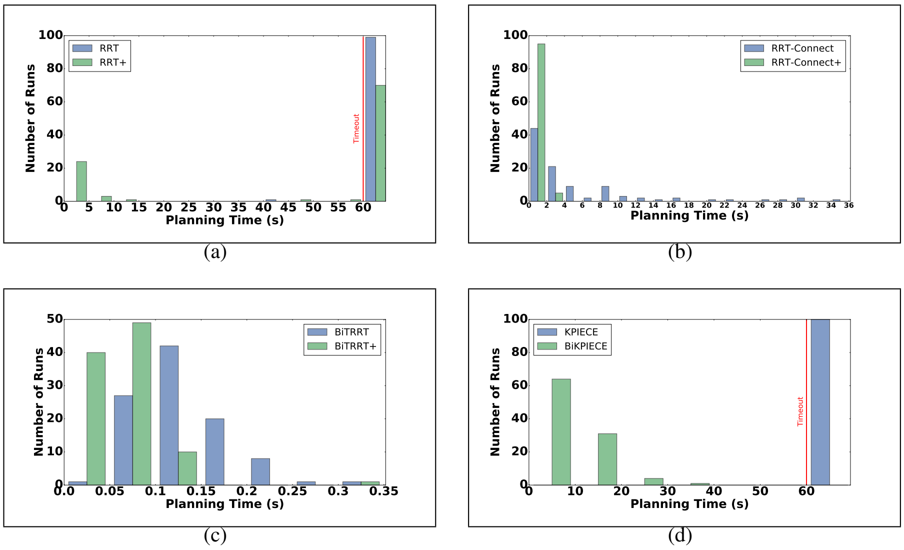


Fig. 11: The histograms comparing the enhanced planners, with the original ones (a-b-c) along with the histograms of KPIECE and BiKPIECE (d). The red line at 60 seconds indicates the Timeout, and the results after that line should be considered failures. The timeout was chosen to facilitate the large number of tests.

The approach is general enough to be applied to a broad variety of motion planning problems. For example, our experiments show potential for planning with costmaps via an adaptation of the bidirectional T-RRT. The enhanced planners are also readily adapted, by customizing the subspace selection technique, to individual problems. The study clearly shows that such methods have potential for rapidly solving seemingly difficult problems. One possible avenue for future research would be to utilize these kinds of planners for the initial step of InformedRRT* [9], Batch Informed Trees [10] or Trajopt [31], in order to reduce the search space faster, extending the applicability of such optimal planning techniques to systems of higher dimensionality.

Future work will target a number of important questions. First, it would be very interesting to find an efficient way to choose subspaces that are more likely to contain solutions. Second, it is important to provide a general way to identify when a new iteration should begin, using a metric of the expansion of the tree, and eliminating the α and T parameters. Moreover, by using the previous metric, it is possible to identify when the tree overcame a difficult area and then reduce the dimensionality of the search, in order to accelerate the results further. For both of the above problems discrete methods such as the one of Şucan and Kavraki [35] should also be considered.

Currently, there are two planners in OMPL that in some cases outperform the RRT$^+$ planners in our experiments: KPIECE which uses random 2D or 3D projections to estimate the coverage efficiently and STRIDE which samples non-uniformly with a bias to narrow spaces. Although, we show that this is not the case for very high dimensional problems that are demanding a solution in a timely manner, there are two different ways of enhancing those planners with the proposed method. Our study can extend the first planner to sample strictly in subspaces of $\mathscr{C}$ and use 2D, 3D or 4D projections to estimate the coverage, and the second planner can be used with the enhancement to accelerate the solutions in the lower dimensional subspaces where narrow passages are expected to be common. The integration of the ideas underlying the proposed method with those planners to accelerate the results is left for future work.

Lastly, we plan to explore the ability of the method to efficiently produce paths that satisfy some natural constraints of each system. Since the subspaces are defined only by simple constraints between the DoFs, subspaces that satisfy some dynamic constraints can be explored by defining dynamic constraints between the different DoFs. Similarly, the ability of the method to find alternative solutions by avoiding exploring non-desired areas shows some potential.

ACKNOWLEDGMENT

The authors would like to thank the generous support of the Google Faculty Research Award and the National Science Foundation grants (NSF 0953503, 1513203, 1526862, 1637876, 1659514, 1849291).

References

1. Bayazit, O.B., Xie, D., Amato, N.M.: Iterative relaxation of constraints: A framework for improving automated motion planning. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3433–3440 (2005)
2. Berenson, D., Kuffner, J., Choset, H.: An optimization approach to planning for mobile manipulation. In: IEEE Int. Conf. on Robotics and Automation, pp. 1187–1192 (2008)
3. Bertram, D., Kuffner, J., Dillmann, R., Asfour, T.: An integrated approach to inverse kinematics and path planning for redundant manipulators. In: Proc. IEEE Int. Conf. on Robotics and Automation, pp. 1874–1879 (2006)
4. Canny, J.: The complexity of robot motion planning. MIT press (1988)
5. Carpin, S., Pagello, E.: On parallel RRTs for multi-robot systems. In: Proc. 8th Conf. Italian Association for Artificial Intelligence, pp. 834–841 (2002)
6. Chamzas, C., Shrivastava, A., Kavraki, L.E.: Using Local Experiences for Global Motion Planning. arXiv e-prints arXiv:1903.08693 (2019)
7. Chitta, S., Sucan, I., Cousins, S.: Moveit![ros topics]. IEEE Robotics & Automation Magazine **19**(1), 18–19 (2012)
8. Esposito, J.M.: Conditional density growth (cdg) model: a simplified model of rrt coverage for kinematic systems. Robotica **31**(05), 733–746 (2013)
9. Gammell, J.D., Srinivasa, S.S., Barfoot, T.D.: Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. arXiv preprint arXiv:1404.2334 (2014)
10. Gammell, J.D., Srinivasa, S.S., Barfoot, T.D.: Batch informed trees (bit*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs. In: Robotics and Automation (ICRA), 2015 IEEE International Conference on, pp. 3067–3074. IEEE (2015)
11. Gipson, B., Moll, M., Kavraki, L.E.: Resolution independent density estimation for motion planning in high-dimensional spaces. In: IEEE Int. Conf. on Robotics and Automation, pp. 2437–2443 (2013)
12. Gochev, K., Cohen, B., Butzke, J., Safonova, A., Likhachev, M.: Path planning with adaptive dimensionality. In: Fourth annual symposium on combinatorial search (2011)
13. Jaillet, L., Cortés, J., Siméon, T.: Transition-based RRT for path planning in continuous cost spaces. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 2145–2150 (2008)
14. Jia, B., Pan, Z., Manocha, D.: Fast motion planning for high-dof robot systems using hierarchical system identification. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 5140–5147. IEEE (2019)
15. Kalakrishnan, M., Chitta, S., Theodorou, E., Pastor, P., Schaal, S.: Stomp: Stochastic trajectory optimization for motion planning. In: 2011 IEEE international conference on robotics and automation, pp. 4569–4574. IEEE (2011)
16. Kavraki, L.E., Švestka, P., Latombe, J.C., Overmars, M.H.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Trans. on Robotics and Automation **12**(4), 566–580 (1996)

17. Kim, D.H., Choi, Y.S., Park, T., Lee, J.Y., Han, C.S.: Efficient path planning for high-DOF articulated robots with adaptive dimensionality. In: IEEE Int. Conf. on Robotics and Automation, pp. 2355–2360 (2015)
18. Klamt, T., Behnke, S.: Towards learning abstract representations for locomotion planning in high-dimensional state spaces. arXiv preprint arXiv:1903.02308 (2019)
19. Kuffner, J., Nishiwaki, K., Kagami, S., Inaba, M., Inoue, H.: Motion planning for humanoid robots. In: The Eleventh Int. Symposium Robotics Research, pp. 365–374 (2005)
20. Kuffner, J.J., LaValle, S.M.: RRT-connect: An efficient approach to single-query path planning. In: IEEE Int. Conf. on Robotics and Automation, vol. 2, pp. 995–1001 (2000)
21. Kuffner Jr, J.J., Kagami, S., Nishiwaki, K., Inaba, M., Inoue, H.: Dynamically-stable motion planning for humanoid robots. Autonomous Robots **12**(1), 105–118 (2002)
22. LaValle, S.M.: Rapidly-exploring random trees: A new tool for path planning. Tech. Rep. TR 98-11, Computer Science Dept., Iowa State University (1998)
23. LaValle, S.M.: Planning Algorithms. Cambridge Univ. press (2006)
24. Lee, I., Oh, J., Bae, H.: Constrained whole body motion planning in task configuration and time. International Journal of Precision Engineering and Manufacturing **19**(11), 1651–1658 (2018)
25. Liu, H., Sun, Q., Zhang, T.: Hierarchical RRT for humanoid robot footstep planning with multiple constraints in complex environments. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 3187–3194 (2012)
26. Otani, T., Koshino, M.: Applying a path planner based on RRT to cooperative multirobot box-pushing. Artificial Life and Robotics **13**(2), 418–422 (2009)
27. Park, J.J., Kim, H.S., Song, J.B.: Collision-free path planning for a redundant manipulator based on PRM and potential field methods. Journal of Institute of Control, Robotics and Systems **17**(4), 362–367 (2011)
28. Qian, Y., Rahmani, A.: Path Planning Approach for Redundant Manipulator Based on Jacobian Pseudoinverse-RRT Algorithm. In: 6th Int. Conf. on Intelligent Robotics and Applications, pp. 706–717. Busan, South Korea (2013)
29. Ratliff, N., Zucker, M., Bagnell, J.A., Srinivasa, S.: Chomp: Gradient optimization techniques for efficient motion planning. In: International Conference on Robotics and Automation (ICRA), pp. 489–494. IEEE (2009)
30. Reif, J.H.: Complexity of the generalized mover's problem. Tech. rep., Harvard University, Cambridge, MA Aiken Computation Lab (1985)
31. Schulman, J., Duan, Y., Ho, J., Lee, A., Awwal, I., Bradlow, H., Pan, J., Patil, S., Goldberg, K., Abbeel, P.: Motion planning with sequential convex optimization and convex collision checking. The International Journal of Robotics Research **33**(9), 1251–1270 (2014)
32. Shkolnik, A., Tedrake, R.: Path planning in 1000+ dimensions using a task-space voronoi bias. In: IEEE International Conference on Robotics and Automation, pp. 2061–2067. IEEE (2009)
33. Solovey, K., Salzman, O., Halperin, D.: Finding a needle in an exponential haystack: Discrete rrt for exploration of implicit roadmaps in multi-robot motion planning. In: Algorithmic Foundations of Robotics XI, pp. 591–607. Springer (2015)
34. Şucan, I.A., Kavraki, L.E.: Kinodynamic motion planning by interior-exterior cell exploration. In: Algorithmic Foundation of Robotics VIII, pp. 449–464. Springer (2009)
35. Şucan, I.A., Kavraki, L.E.: On the performance of random linear projections for sampling-based motion planning. In: Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference On, pp. 2434–2439. IEEE (2009)
36. Şucan, I.A., Moll, M., Kavraki, L.E.: The open motion planning library. IEEE Robotics & Automation Magazine **19**(4), 72–82 (2012)
37. Van Den Berg, J.P., Overmars, M.H.: Prioritized motion planning for multiple robots. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 430–435 (2005)
38. Vannoy, J., Xiao, J.: Real-time adaptive motion planning (ramp) of mobile manipulators in dynamic environments with unforeseen changes. IEEE Trans. on Robotics **24**(5), 1199–1212 (2008)
39. Vernaza, P., Lee, D.D.: Efficient dynamic programming for high-dimensional, optimal motion planning by spectral learning of approximate value function symmetries. In: IEEE Int. Conf. on Robotics and Automation, pp. 6121–6127 (2011)
40. Wagner, G.: Subdimensional expansion: A framework for computationally tractable multirobot path planning. Master thesis (2015)
41. Weghe, M.V., Ferguson, D., Srinivasa, S.S.: Randomized path planning for redundant manipulators without inverse kinematics. In: 7th IEEE-RAS Int. Conf. on Humanoid Robots, pp. 477–482 (2007)
42. Wells, A., Plaku, E.: Adaptive sampling-based motion planning for mobile robots with differential constraints. In: Conference Towards Autonomous Robotic Systems, pp. 283–295. Springer (2015)
43. Xanthidis, M., Kyriakopoulos, K.J., Rekleitis, I.: Dynamically Efficient Kinematics for Hyper-Redundant Manipulators. In: The 24th Mediterranean Conf. on Control and Automation, pp. 207–213. Athens, Greece (2016)

44. Yershova, A., Jaillet, L., Siméon, T., LaValle, S.M.: Dynamic-domain RRTs: Efficient exploration by controlling the sampling domain. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 3856–3861 (2005)
45. Yoshida, E.: Humanoid motion planning using multi-level dof exploitation based on randomized method. In: 2005 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 3378–3383 (2005)
46. Zha, F., Liu, Y., Wang, X., Chen, F., Li, J., Guo, W.: Robot motion planning method based on incremental high-dimensional mixture probabilistic model. Complexity **2018** (2018)