Effective Metrics for Multi-Robot Motion-Planning*

Aviel Atias, Kiril Solovey and Dan Halperin

Blavatnik School of Computer Science
Tel Aviv University, Israel

***Abstract*—We study the effectiveness of metrics for Multi-Robot Motion-Planning (MRMP) when using RRT-style sampling-based planners. These metrics play the crucial role of determining the nearest neighbors of configurations and in that they regulate the connectivity of the underlying roadmaps produced by the planners and other properties like the quality of solution paths. After screening over a dozen different metrics we focus on the five most promising ones—two more traditional metrics, and three novel ones which we propose here, adapted from the domain of shape-matching. In addition to the novel multi-robot metrics, a central contribution of this work are tools to analyze and predict the effectiveness of metrics in the MRMP context. We identify a suite of possible substructures in the configuration space, for which it is fairly easy (i) to define a so-called *natural distance*, which allows us to predict the performance of a metric. This is done by comparing the distribution of its values for sampled pairs of configurations to the distribution induced by the natural distance; (ii) to define equivalence classes of configurations and test how well a metric covers the different classes. We provide experiments that attest to the ability of our tools to predict the effectiveness of metrics: those metrics that qualify in the analysis yield higher success rate of the planner with fewer vertices in the roadmap. We also show how combining several metrics together leads to better results (success rate and size of roadmap) than using a single metric.**

I. Introduction

Multi-robot motion-planning (MRMP) is the problem of planning the motion of a fleet of robots from given start to goal configurations, while avoiding collisions with obstacles and with each other. It is a natural extension of the standard single-robot motion-planning problem. MRMP is notoriously challenging, both from the theoretical and practical standpoint, as it entails a prohibitively-large search space, which accounts for a multitude of robot-obstacle and robot-robot interactions.

Sampling-based planners have proven to be effective in challenging settings of the single-robot case, and a number of such planners have been proposed for MRMP [44, 50, 51]. Sampling-based planners attempt to capture the connectivity of the free space by sampling random configurations and connecting *nearby* configurations by simple collision-free paths. In order to measure similarity, or "closeness", between a given pair of configurations a *metric* is employed by the algorithm. The choice of metric has a tremendous effect on the performance of planners and the quality of the returned solutions (see Section II for further discussion about metrics that are tailored for various robotic systems). Nevertheless, no specialized metrics for multi-robot systems have been proposed, to the best of our knowledge.

* This work has been supported in part by the Israel Science Foundation (grant no. 825/15), by the Blavatnik Computer Science Research Fund, and by the Hermann Minkowski–Minerva Center for Geometry at Tel Aviv University. Kiril Solovey is also supported by the Clore Israel Foundation.

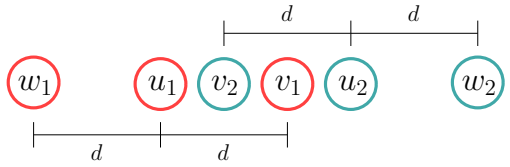


Fig. 1. Example of ΣL_2 for the setting of $m = 2$ disc robots in the plane. The red discs, centered in u_1, v_1, w_1 represent possible positions for the first robot, whereas the blue discs, centered in u_2, v_2, w_2, represent possible positions for the second robot. We set the positions in the following manner: $\|u_1 - v_1\|_2 = \|u_1 - w_1\|_2, \|u_2 - v_2\|_2 = \|u_2 - w_2\|_2$. $U = (u_1, u_2), V = (v_1, v_2), W = (w_1, w_2)$ represent three simultaneous placements of the two robots. While $\Sigma L_2(U, V) = \Sigma L_2(U, W)$, it is intuitive that it is easier to connect U to W rather than to V. This example hints that ΣL_2 may not be suitable for all cases as it fails to capture robot-robot interaction.

Nowadays, a common metric for multi-robot systems is defined as a sum of metric values for single robots ([38, 44], and in fact this is the default in OMPL [46]), i.e., the sum of distances induced by each of the robots separately. We denote this metric by ΣL_2 (to be formally defined in Section IV). This metric does not always adequately express distance in the *configuration space* ($\mathcal{C}$-space) because it does not account for interactions between different robots. A simple example is shown in Figure 1.

A. Contribution

In this work we consider the problem of devising good distance metrics for MRMP. We proceed to introduce several new metrics for MRMP and show that they improve upon the standard ΣL_2 metric in various settings. Our new metrics combine ideas from various fields of study such as computational geometry, shape matching and image processing. The main benefit of the new metrics, is that they do not only take the relative positions of the same robot into consideration, but also the interactions between the different robots. We consider several properties that such metrics should maintain and describe how to analyze those properties for a given metric.

We present experimental results, which show that our metrics improve the effectiveness of motion planning, even in complex environments, and suggest to use this type of metrics side-by-side with traditional multi-robot metrics in order to be able to effectively solve various problem instances.

B. Organization

The organization of this paper is as follows. In Section II we review related work. In Section III we describe the early phase of our investigation, where we tested a large number of metrics with different planners, and explain why we chose the metrics and planner on which we focus in the sequel. In Section IV we formally define five metrics which will be discussed later. In Sections V and VI we present methods for analyzing the proposed metrics using identification of substructures arising in MRMP. In Section VII we provide experimental results allowing us to compare the utility of the metrics. Finally, in Section VIII we outline the possible future work. An extended version of this paper, to which we refer throughout the text, is available at https://arxiv.org/abs/1705.10300.

II. Related Work

We start this section with work related to multi-robot motion planning (MRMP). Then, we proceed to discuss metrics in the context of robots and beyond. We assume some familiarity with basic concepts of sampling-based motion planning (see, e.g., [12, 19, 31]).

A. Multi-robot motion-planning

Approaches to solving MRMP can be roughly subdivided into two types: *coupled* and *decoupled*. In the latter approach (see, e.g., [5, 33, 48]), a path or an initial plan are found for each robot separately, and then the paths are coordinated with each other. Although this approach is less sensitive to the number of robots, when compared with the coupled approach, it gives no completeness guarantees.

The coupled approach usually treats the entire system as a single robot, for which the number of *degrees of freedom* (DOFs) is equal to the sum of the number of DOFs of the individual robots in the system. This approach usually comes with stronger theoretical guarantees such as completeness [29, 39, 40, 41, 44] or even optimality [51] of the returned solutions. However, due to the computational hardness of MRMP [21, 22, 26, 42, 45], coupled techniques do not scale well with the increase in the number of robots. We do mention that, when simplifying assumptions are made concerning the separation of initial and goal positions, MRMP can be solved in polynomial time, as function of the number of robots and the complexity of the workspace environment (see, [1, 43, 47]).

B. Metrics

The choice of a metric for nearest-neighbors queries in a sampling-based planner can be crucial. Amato et al. [3] were the first to study the effect of a metric on sampling-based planners. They consider PRM as the planner and define effectiveness as the number of discovered edges in the roadmap. They compare effectiveness of some variants of the Euclidean metric in settings that involve translation and rotation of a single robot. Kuffner [30] considers metrics for rigid-body motion and proposes an interpolation between the rotation component and the translation component.

Extensive research was done in order to find suitable metrics for other settings of motion planning, such as robots with differential constraints [7, 8, 32, 36].

Pamecha et al. [37] analyze metrics for systems with a single robot consisting of multiple modules that must stay in touch with each other (*multi-module systems*). Though any module can be thought of as a robot, the system restrictions are that modules are only allowed to move on a grid, and must stay in contact in order to form a metamorphic robot. Hence, their results are not straightforward to extend to arbitrary multi-robot systems. Further analysis for multi-module systems can be found in Winkler et al. [52] and Zykov et al. [53].

Recent methods employ machine learning to develop metrics that are tailored to the specific motion-planning problem at hand. Ekenna et al. [15] introduce a framework in which there is a candidate set of metrics, and the planner adaptively selects a metric on-the-fly. The selection may vary over time or between different regions of the workspace. This implies that a set of metrics, each suitable for a different setting, can be combined in order to solve more diverse settings that consist of smaller, specific, (sub)settings. Morales et al. [35] have the same observation that different portions of the $\mathcal{C}$-space may behave differently. In our work we will also refer to the case where different metrics are more effective than others in different portions of the $\mathcal{C}$-space.

Estimating distances between sets of points is in broad use in shape matching (see the survey [49]). Such techniques (see, e.g., Belongie et al. [6]) are concerned with estimating the distance between shapes and with finding a matching between shapes. Kendall [28] provides a rigorous mathematical study of the subject, where point sets are mapped to high-dimensional points, on which distance measures can be more easily defined (see more details in Section IV).

Another area where distance between sets of points is of interest is graph drawing. Bridgeman and Tamassia [9] list a large number of distance metrics between planar graphs. Some of the metrics give a significant weight to the relative order between the nodes, which is also the guideline for the metrics we propose in this paper. Lyons et al. [34] address the same problem, and measure similarity based on both Euclidean distance and relative order between the nodes.

III. Initial Screening

We began our study by experimenting with four different planners, fifteen different metrics and variations of them. For planners we tried RRT-style and EST-style [23] planners that are adapted to the multi-robot setting. We tested both single-tree and bi-directional variants of each algorithm. PRM-style planners cannot cope with the induced high-dimensional space. RRT-style planners showed much better success rate in solving MRMP problems when compared to EST-style planners. This is why the study continues henceforth with dRRT [44]—an adaptation of RRT to the multi-robot setting, which can cope with a larger number of robots and more complicated tasks than RRT as-is. We mention that M* [51], which is another sampling-based planner tailored for MRMP, is less

relevant to our current discussion since it only employs metrics concerning individual robots.

For metrics, we began by following the common approach of choosing metrics that have high correlation with the failure rate of the local planner [12, pp. 210]. Note that this is also the guideline behind using the swept volume and its approximations as a metric for rotating robots [3, 15, 30]. It turns out that when using such metrics with RRT-style planners, the exploration of the $\mathcal{C}$-space is unbalanced— the explored configurations tend to have the robots separated from each other. The analogue for single-robot planning is exploration of configurations that tend to be far from obstacles, avoiding paths that go near the obstacles. This phenomenon is reported in the extended version of this paper.

We continued with metrics that adapt geometric methods from the domain of shape-matching [6, 16, 17, 28], including existing methods that are used for mismatch measure [2]. We also used measures of similarities that are employed in the domain of graph-drawing [9, 34].

Out of the fifteen tested metrics and their variations, we remained with the most successful five metrics that are described below in Section IV.

Finally, we mention that we experimented with several types of robots including planar ones that are allowed to translate and rotate. However, we chose to conduct our final experiments with robots bound to translate in the plane, as it makes the presentation clearer. Moreover, we believe that the study of complex rigid-body motion [30] in the context of metrics is mostly orthogonal to our current efforts.

IV. Metrics for Multi-Robot Motion-Planning

In this section we discuss the role of metrics in sampling-based MRMP. Then, we formally define the standard ΣL_2, $\max L_2$ metrics and introduce the metrics ε_2, ε_∞, Ctd, which will be evaluated in Section VII.

We consider m robots $r_1, \ldots, r_m$ operating in a shared workspace. For simplicity we assume that the robots are identical in shape and function, i.e., the $\mathcal{C}$-space of each individual robot can be denoted by some $\mathcal{X}$. Note that we still distinguish between the different robots. We assume that each r_i represents a translating disc in the plane, and so $\mathcal{X} = \mathbb{R}^2$. Denote the *joint $\mathcal{C}$-space* for the m individual robots by $\mathcal{X}^m = \mathcal{X} \times \ldots \times \mathcal{X}$, i.e., a *joint configuration* $U = (u_1, \ldots, u_m)$ represents a set of configurations for the m robots. We note that the metrics described below can be extended to more general settings of the problem, such as non-disc robots and 3D environments.

Sampling-based tools for single and multi-robot systems rely on metrics to measure similarity between configurations. Let U, V, W be joint configurations of our multi-robot system. A metric in the context of MRMP is a distance function $\mathrm{d}\colon \mathcal{X}^m \times \mathcal{X}^m \to [0, \infty)$, which satisfies the five properties:

(a) *non-negativity*: $\mathrm{d}(U, V) \geq 0$;
(b) *identity*: $\mathrm{d}(U, U) = 0$;
(c) *identity of indiscernibles*: $\mathrm{d}(U, V) = 0 \Rightarrow U = V$;
(d) *symmetry*: $\mathrm{d}(U, V) = \mathrm{d}(V, U)$;
(e) *triangle inequality*: $\mathrm{d}(U, W) \leq \mathrm{d}(U, V) + \mathrm{d}(V, W)$.

Efficient nearest-neighbors data structures usually do not rely on property (c) (see, e.g., [10, 11, 13]), and so can be applied to *pseudometrics*, which satisfy properties (a), (b), (d) and (e). We extend the discussion also to *pseudosemimetrics* which are functions that satisfy only properties (a), (b), and (d). In that case, we cannot use sophisticated data structures that rely on the triangle inequality. For simplicity, from now on we will refer to any pseudosemimetric as a *metric*.

Standard metrics. The following two metrics are simple extensions of single-robot metrics to the multi-robot setting. Let L be a single-robot metric $L : \mathcal{X} \times \mathcal{X} \to [0, \infty)$. For any two joint configurations $U = (u_1, \ldots, u_m), V = (v_1, \ldots, v_m) \in \mathcal{X}^m$ we define ΣL and $\max L$ as:

$$\Sigma L\,(U, V) = \sum_{i=1}^{m} L\,(u_i, v_i)\,,$$
$$\max L\,(U, V) = \max_{i=1,\ldots,m} L\,(u_i, v_i)\,.$$

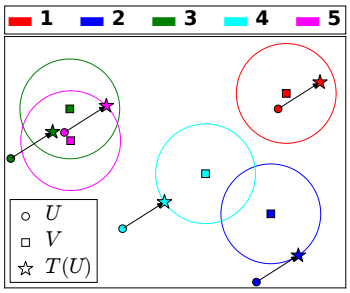
We consider the two metrics obtained by setting $L = L_2$, which is the standard Euclidean distance, and denote them by ΣL_2 and $\max L_2$. Those metrics satisfy properties (a)-(e). We note that the former is used by default in many settings, whereas the latter has earned much less attention.

ε-congruence metrics. Here we introduce new metrics, which are based on the notion of *approximate congruence* or *ε-congruence*, described by Alt et al. [2].

Definition IV.1 (*ε-congruence*). *Let $L : \mathcal{X} \times \mathcal{X} \to [0, \infty)$ be a single-robot metric, and let $\mathcal{T}$ be the set of all translations $T : \mathcal{X} \to \mathcal{X}$. For every two joint configurations $U = (u_1, \ldots, u_m), V = (v_1, \ldots, v_m) \in \mathcal{X}^m$ the ε-congruence with respect to L is defined as*

$$\varepsilon_L\,(U, V) = \min_{T \in \mathcal{T}} \max_{i=1,\ldots,m} L\,(T\,(u_i), v_i)\,.$$

This metric expresses the required tolerance (with respect to L) for the two sets of points to be equivalent to each other under translation. We denote ε-congruence with respect to L_2 and L_∞ by ε_2 and ε_∞, respectively. See illustration on the right: U is marked with circles, V with squares, and the translated configuration $T\,(U)$ with stars. Each of the $m = 5$ robots is denoted by a different color. If each star falls inside its corresponding ball then the balls' (common) radius corresponds to a valid translation. The ε-congruence is the minimal valid radius.



Note that ε-congruence satisfies all the properties of a pseudosemimetric, and in case L satisfies the triangle inequality (which is the case for L_2 and L_∞) then ε-congruence is a pseudometric and therefore can be used with any nearest-

neighbor data structure.

Shape-based metric. Let $U = (u_1, \ldots, u_m)$ and $V = (v_1, \ldots, v_m)$ be two joint configurations for m robots. Denote by x_i and y_i the x and y coordinates (respectively) of $v_i - u_i$. The *Centroid* distance is defined as the sum of squared Euclidean distances between $v_i - u_i$ and the common centroid of $\{v_i - u_i\}_{i=1}^m$. The centroid distance is calculated using the following equation:

$$\mathrm{Ctd}(U,V) = \sum_{i=1}^{m} \left(x_i^2 + y_i^2\right) - \frac{\left(\sum_{i=1}^m x_i\right)^2 + \left(\sum_{i=1}^m y_i\right)^2}{m}. \quad (1)$$

The development of Equation (1) is based on the notion of *shape space* [28]. Refer to the extended version of this paper for intuition and full details.

In summary, we have presented five metrics for MRMP: the more traditional ΣL_2 and $\max L_2$, and the novel metrics ε_2, ε_∞, Ctd. We will evaluate these five metrics below.

V. Canonical Substructures in $\mathcal{C}$-space

Here we introduce a new approach to better conquer the intricate problem of MRMP. We identify several "gadgets", which represent local instances of the problem, and which force the robots to coordinate in a specific and prescribed manner. Those gadgets can be viewed as a set of representative tasks that need to be carried out in typical scenarios of MRMP. Examining these substructures, rather than the entire complex problem, has two benefits. Firstly, such substructures can be straightforwardly decomposed into a small number of equivalence classes (ECs) of (joint) configurations, which can be viewed as a discrete summary of the continuous problem. We conjecture that a metric which maximizes the number of explored ECs by a given planner also leads to better performance of the planner. Secondly, those ECs of a given substructure, and the relations between them, induce a *natural distance metric*, which faithfully quantifies how difficult it is to move between any given pair of joint configurations. This gives an additional method to assess the quality of a given metric by comparing it to the natural metric.

In the remainder of this section we describe three such canonical substructures, which we refer to as Permutations, Partitions, and Pebbles, and denote them by $\mathbb{X}_{\text{Permutations}}, \mathbb{X}_{\text{Partitions}}, \mathbb{X}_{\text{Pebbles}}$. We also describe their corresponding natural metrics. In Section VI we describe tools for analysis of metrics. Of course there could be many more useful substructures—see comment in the concluding section.

Each such substructure $\mathbb{X}$ is a subset of the joint $\mathcal{C}$-space $\mathcal{X}^m$. For every $\mathbb{X}$ we identify a finite collection of $e > 0$ disjoint subsets $X_1, \ldots, X_e$ of $\mathbb{X}$ termed *equivalence classes* (ECs). Note that each EC is a subset of the joint $\mathcal{C}$-space. We say that two joint configurations $U, V \in \mathbb{X}$ are *equivalent* if they belong to the same EC X_i. If robots can also leave one EC X_i and enter another $X_{i'}$, without going through any other EC then we say that the ECs $X_i, X_{i'}$ are *neighbors*. This gives rise to the *equivalence graph* $G_{\mathbb{X}}$ whose vertices are the ECs of $\mathbb{X}$, and there is an edge between every two neighboring ECs.

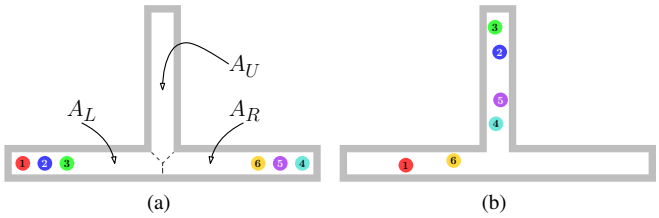


Fig. 2. Tunnel scenario. The environment consists of a T-shaped free space and requires the robots in one side to exchange places with the robots on the other side. There are 6 translating disc robots of radius 2 and the width of each arm is 5, so the robots cannot exchange places within an arm without leaving it. (a) Start configuration. The red, blue and green robots lie on the left arm, and the yellow, purple and cyan robots lie on the right arm. In the goal configuration the red, blue and green robots lie on the right arm and the yellow, purple and cyan robots lie on the left arm. More specifically, the red robot exchanges places with the cyan robot, the blue robot with the purple robot and the green robot with the yellow robot. (b) A configuration for which the permutation in A_U is $(3,2,5,4)$, in A_R is $(\,)$ and in A_L is $(1,6)$. The corresponding EC is denoted by $[(3,2,5,4)\,,(\,)\,,(1,6)]$.

We are now ready to define the *natural distance* $\mathrm{d}_{\mathcal{K}}$ between two given joint configurations $U, V \in \mathbb{X}$. For a given $U \in \mathbb{X}$ denote by $\mathrm{EC}(U)$ the EC of $\mathbb{X}$ in which it resides. Then the natural distance $\mathrm{d}_{\mathcal{K}}(U,V)$ is the graph distance over $G_{\mathbb{X}}$ between $\mathrm{EC}(U)$ and $\mathrm{EC}(V)$, namely the number of edges along the shortest path in the graph between the vertices corresponding to $\mathrm{EC}(U), \mathrm{EC}(V)$.

A. Permutations

As an example of $\mathbb{X}_{\text{Permutations}}$ consider the "Tunnel" scenario depicted in Figure 2. The workspace consists of three portions corresponding to the three "arms" of the workspace: upper arm, right arm and left arm, denoted by $\mathcal{A} = \{A_U, A_R, A_L\}$. In this substructure we define the ECs to correspond to the assignment of robots to portions of the tunnel, and to the specific order of the robots within each portion. The order in the upper arm A_U is calculated according to the y coordinate, and the order in the right and left arms A_R, A_L is determined according to the x coordinate. See Figure 2b for an illustration.

Two ECs are neighbors if they correspond to a transition of a single robot that leaves one arm and enters another. For instance, $[(3,4,2)\,,(5,6,1)\,,(\,)]$ and $[(3,4,2,1)\,,(5,6)\,,(\,)]$ are neighbors. This condition implicitly induces the equivalence graph $G_{\mathbb{X}_{\text{Permutations}}}$ and the corresponding natural metric $\mathrm{d}_{\mathcal{K}}$. For instance, for any two configurations U, V which lie in the ECs $[(3,4,2,5,6,1)\,,(\,)\,,(\,)]$, $[(3,4,1,6,5,2)\,,(\,)\,,(\,)]$, respectively, it follows[1] that $\mathrm{d}_{\mathcal{K}}(U,V) = 10$.

An illustration for the equivalence graph for the case of $m = 2$ robots is depicted in Figure 3.

B. Partitions

As an example of $\mathbb{X}_{\text{Partitions}}$ we consider the "Chambers" scenario depicted in Figure 4. Each EC is associated with

[1]The shortest path over $G_{\mathbb{X}_{\text{Permutations}}}$ can be obtained in the following manner: (1) $r_1 : A_U \to A_R$ (namely, r_1 moves from the upper arm to the right arm) , (2) $r_6 : A_U \to A_R$, (3) $r_5 : A_U \to A_R$, (4) $r_2 : A_U \to A_L$, (5) $r_5 : A_R \to A_L$, (6) $r_6 : A_R \to A_L$, (7) $r_1 : A_R \to A_U$, (8) $r_6 : A_L \to A_U$, (9) $r_5 : A_L \to A_U$ and (10) $r_2 : A_L \to A_U$.

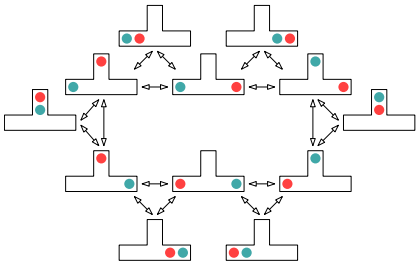
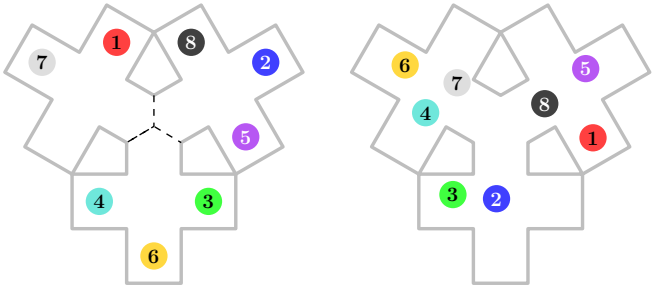


Fig. 3. $G_{\mathbb{X}_{\text{Permutations}}}$ for two robots ($m = 2$). Each vertex of $G_{\mathbb{X}_{\text{Permutations}}}$ represents an EC in the joint $\mathcal{C}$-space $\mathcal{X}^m$.



(a) (b)

Fig. 4. Chambers scenario. The environment consists of three chambers. The structure of each chamber allows the robots to exit from the chamber in any order, not necessarily in the order they entered to the chamber (as opposed to the arms in the Tunnel scenario). (a) Start configuration. (b) A configuration that corresponds to the assignment $[\{1,5,8\},\{4,6,7\},\{2,3\}]$. The natural distance between it and the configuration in Figure 4a is 4.

a partitioning of the robots to the chambers. Each robot is mapped to the chamber that has the largest intersection with the robot and we choose a chamber at random in case that there is a tie. See Figure 4b. Two ECs are neighbors if exactly one robot changes its mapped chamber. Unlike the previous substructure, here the exact order of the robots inside one chamber does not matter.

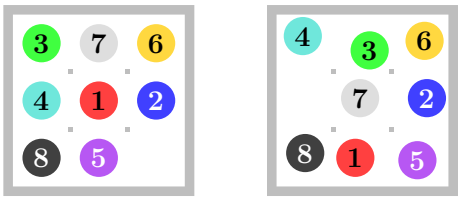
C. Pebbles

The "8-Puzzle" scenario, which is a geometric variation of the classic 15-Puzzle [4], is used as an example for $\mathbb{X}_{\text{Pebbles}}$. The problem is depicted in Figure 5. Unlike the discrete version of the puzzle, where each robot can occupy only one of nine possible places, in the geometric generalization the robots can lie in any collision-free configuration.

Each EC of $\mathbb{X}_{\text{Pebbles}}$ is associated with an assignment of robots to the nine cells. The cell corresponding to each robot is the one that has the largest intersection with the robot, with the restriction that at most one robot is assigned to a single cell, and we choose a cell at random in case that there is a tie. An example for a configuration along with its correspondent assignment is described in Figure 5b. Two ECs of $\mathbb{X}_{\text{Pebbles}}$ are neighbors if exactly one robot changes its cell assignment.

VI. Analysis of Metrics

In this section we introduce two novel tools for analyzing metrics, which rely on the concept of canonical substructures, described in Section V. The following tools assess the quality of a given metric d by quantifying its similarity to the natural



(a) (b)

Fig. 5. 8-Puzzle scenario. The environment can be naturally partitioned into nine cells that form 3×3 grid. A robot can translate only between adjacent cells. (a) Start configuration. The goal is to arrange the robots in the order $r_1,\ldots,r_m$, i.e., r_1 is situated in the cell in the top left corner, and so on. (b) A configuration that corresponds to the assignment $[\{4\},\{3\},\{6\},\{\},\{7\},\{2\},\{8\},\{1\},\{5\}]$. The natural distance between it and the configuration in Figure 5a is 5 since there is a discrete motion with 5 steps that transforms one configuration to the other (the motion involves the purple, red, grey, green and cyan robots).

metric $\mathrm{d}_{\mathcal{K}}$, and by counting number of explored ECs by a planner that is paired with d.

In addition to the tools described in this section, we have a visualization tool that automatically generates an animation for the expanded tree. Some properties of the metrics can be inferred by perusing the animations. This tool was essential in the screening phase and guiding our choice of metrics. Links to example videos can be found in the extended version of this paper.

A. Distributions separation

The following technique requires as an input, after fixing a specific canonical substructure $\mathbb{X}$, a set of ℓ randomly sampled joint configurations $\mathbb{C} = \{C_1,\ldots,C_\ell\}$ from $\mathbb{X}$. Each such sample is then classified according to its EC in $\mathbb{X}$.

Our working hypothesis is that a good metric should faithfully reflect the natural distance, and in the rest of the subsection we spell out what it means to have this property.

When incorporating the metric into a sampling-based planner, the role of the metric is to compare distances between different pairs of sampled configurations. Given two pairs of configurations (U_1,V_1) and (U_2,V_2), the planner favors to check the continuous motion between the first pair in case the distance between U_1,V_1 is smaller than the distance between U_2,V_2 (Note that in the case of an RRT-style planner, the compared pairs always satisfy $U_1 = U_2$.). How much a metric reflects the natural distance can be measured by how well the relation between distances of different pairs of configurations is preserved when compared to the natural distance. Preserving the natural distance can be measured by Γ_{d}:

$$\Gamma_{\mathrm{d}} = \Pr_{U_1,U_2,V_1,V_2\in\mathbb{X}}\Big[\mathrm{d}(U_1,V_1) < \mathrm{d}(U_2,V_2) \;\Big|\; \mathrm{d}_{\mathcal{K}}(U_1,V_1) < \mathrm{d}_{\mathcal{K}}(U_2,V_2)\Big].$$

In one extreme case, if we use the natural distance as d we have $\Gamma_{\mathrm{d}} = 1$. In the other extreme case, if a metric d has no correlation with the natural distance we have $\Gamma_{\mathrm{d}} = 0.5$. We are interested in a metric that gives a large value of Γ_{d}.

In the rest of the subsection we formalize the discussion above and explain how to calculate and compare Γ_{d} between

different metrics. For every possible (discrete) value of the natural distance $\alpha \in \operatorname{Im} \mathrm{d}_{\mathcal{K}}$ we compute the set ${\mathcal{D}_{\mathrm{d}}}^{\alpha}$ of metric distances given that the natural distance is α:

$$ {\mathcal{D}_{\mathrm{d}}}^{\alpha} = \{\mathrm{d}(U,V) \mid U,V \in \mathbb{C}, \mathrm{d}_{\mathcal{K}}(U,V) = \alpha\}. $$

With a slight abuse of notation, we treat ${\mathcal{D}_{\mathrm{d}}}^{\alpha}$ as a distribution over pairs of configurations from $\mathbb{X}$. Here we use the fact that $\mathbb{C}$ captures the structure of $\mathbb{X}$. Furthermore, we define $\mathcal{D}_{\mathrm{d}} = \{{\mathcal{D}_{\mathrm{d}}}^{\alpha} \mid \alpha \in \operatorname{Im} \mathrm{d}_{\mathcal{K}}\}$. Consequently, Γ_{d} can be represented as

$$ \Gamma_{\mathrm{d}} = \Pr\Big[\alpha_0 < \beta_0 \;\Big|\; \alpha_0 \sim {\mathcal{D}_{\mathrm{d}}}^{\alpha}, \beta_0 \sim {\mathcal{D}_{\mathrm{d}}}^{\beta}, \alpha < \beta\Big], $$

where the notation $\alpha_0 \sim {\mathcal{D}_{\mathrm{d}}}^{\alpha}$ indicates that α_0 is sampled from the distribution ${\mathcal{D}_{\mathrm{d}}}^{\alpha}$.

Sampling-based planners usually attempt to connect nearby configurations. Thus, it is more important to identify close configurations than remote ones. Pairs of far-away configurations (with respect to the natural distance) are practically ignored by a sampling-based planner that uses a reasonable metric d. We restrict Γ_{d} to natural distances of at most a threshold parameter τ, using the following definition[2] of ${\Gamma_{\mathrm{d}}}^{\tau}$:

$$ {\Gamma_{\mathrm{d}}}^{\tau} = \Pr\Big[\alpha_0 < \beta_0 \;\Big|\; \alpha_0 \sim {\mathcal{D}_{\mathrm{d}}}^{\alpha}, \beta_0 \sim {\mathcal{D}_{\mathrm{d}}}^{\beta}, \alpha < \beta, \alpha \leq \tau\Big]. $$

We expect that a metric d_1 will be more effective than a metric d_2 if ${\Gamma_{\mathrm{d}_1}}^{\tau} > {\Gamma_{\mathrm{d}_2}}^{\tau}$.

B. Explored equivalence classes

RRT-style planners, as the one used and described later on in Section VII, explore the $\mathcal{C}$-space from a starting configuration. A desirable property of such planners is to reach various regions of interest in the $\mathcal{C}$-space. In our setting, we measure the quality of exploration by the number of different ECs reached, where a larger number of explored ECs means that the planner explores the $\mathcal{C}$-space more exhaustively. Since the planner cannot foresee which parts of the $\mathcal{C}$-space can lead to a solution, we expect that an effective metric will result in a larger number of explored ECs when compared to an ineffective one.

We propose the following experiment to assess d with respect to the quality of exploration. A single-tree RRT-style planner is used to build a tree with N vertices. The set of explored configurations is denoted by $\mathcal{U}_{\mathrm{d}}$. For each configuration $U \in \mathcal{U}_{\mathrm{d}}$ we identify its representative EC denoted by $\mathrm{EC}(U)$. We count the number of distinct explored ECs, i.e. the number of distinct ECs in the set $\{\mathrm{EC}(U) \mid U \in \mathcal{U}_{\mathrm{d}}\}$, and denote it by $|\mathcal{U}_{\mathrm{d}}/\mathrm{EC}|$. We anticipate that a metric d_1 will be more effective than a metric d_2 if $|\mathcal{U}_{\mathrm{d}_1}/\mathrm{EC}| > |\mathcal{U}_{\mathrm{d}_2}/\mathrm{EC}|$.

VII. Experimental Results

In this section we make use of the tools developed in Section VI to analyze the properties of the metrics in the scenarios described in Section V. Then we compare the effectiveness of the metrics as used by dRRT [44] to solve instances of MRMP. As mentioned in Section III, dRRT is an extension of RRT, which allows it to cope with a greater number of robots and more complex scenarios. Later on we show the effectiveness of the planner incorporated with different metrics in a general environment that consists of several substructures.

[2]We require that $\alpha \leq \tau$, and not β, since we only care that pairs of configurations with small value of $\mathrm{d}_{\mathcal{K}}$ will remain so with respect to d. A similar correlation is not assumed between large distances.

A. Implementation details

All the metrics defined in Section IV can be implemented with running time linear in the number of robots. Refer to the extended version of this paper for full description of the implementation.

All the experiments were performed using a cluster of 40 single-core virtual machines running over Google Compute Engine [18]. The planning is based on the *Open Motion Planning Library (*OMPL*)* [46].

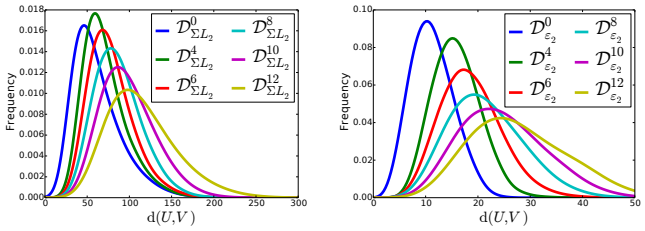
B. Analyzing properties of the metrics

We show and analyze the results of the experiments described in Section VI using the scenarios described in Section V.

For each scenario, we show results for the value of ${\Gamma_{\mathrm{d}}}^{\tau}$ defined in Section VI-A. Then, we count the number of distinct explored ECs, as suggested in Section VI-B. In order to do so, we use a dRRT-tree with 10,000 vertices rooted at the start configuration (see Figures 2a, 4a and 5a) . Finally, we show the effectiveness of an entire planning algorithm that uses each of the metrics and show how it correlates with the results of the analysis tools. We measure the effectiveness of the planner by inspecting both (i) the number of explored vertices when a solution is found—the lower the number, the more effective we consider the metric to be; and (ii) the success rate of the planner. We mention that the success rate of the local planner (and not the motion planner) is similar among all the metrics, and therefore we do not report it. We do not measure running times since we are interested only in the analytic effectiveness of each metric. As noted earlier, all metrics require similar computation time.

Next, for each typical scenario we describe (i) the results of the distributions-separation predicates, (ii) the results of the ECs exploration, and finally (iii) the actual behavior of the planner and its relation to the predictions. These are also summarized in Table I, Figure 7 and Figure 8, respectively.

Permutations substructure. Figure 6 shows subsets of the sets of distributions $\mathcal{D}_{\varepsilon_2}$ and $\mathcal{D}_{\Sigma L_2}$ for the Tunnel scenario: observe that the distributions in $\mathcal{D}_{\varepsilon_2}$ are better separated than the distributions in $\mathcal{D}_{\Sigma L_2}$. This separation is expressed by the dissimilarities between the different distributions. For example, the common area bounded by the blue and green distributions (representing ${\mathcal{D}_{\mathrm{d}}}^{0}$ and ${\mathcal{D}_{\mathrm{d}}}^{4}$ respectively) is smaller for ε_2 when compared to ΣL_2. This is also the case for the green and red distributions (representing ${\mathcal{D}_{\mathrm{d}}}^{4}$ and ${\mathcal{D}_{\mathrm{d}}}^{6}$ respectively). The value ${\Gamma_{\mathrm{d}}}^{\tau}$ quantifies the distribution separation. For this scenario we set $\tau = 4$. The values of ${\Gamma_{\mathrm{d}}}^{\tau}$ are given in Table I. The values for ε_2, ε_∞ and Ctd are similar to each other, and are larger than the values for ΣL_2 and $\max L_2$.



(a) $\mathcal{D}_{\Sigma L_2}$ (b) $\mathcal{D}_{\varepsilon_2}$

Fig. 6. Distributions from $\mathcal{D}_{\mathrm{d}}$ for ΣL_2 and ε_2 metrics in the Tunnel scenario. Better reflection of the natural distance is expressed by higher level of separability between the distributions.

Scenario	τ	**Metric (d)**				
		ΣL_2	$\max L_2$	ε_2	ε_∞	Ctd
Tunnel	4	0.810	0.843	0.904	0.904	0.907
Chambers	1	0.858	0.983	0.971	0.962	0.938
8-Puzzle	7	0.953	0.938	0.951	0.921	0.971

TABLE I The value of $\Gamma_{\mathrm{d}}{}^{\tau}$ for different metrics in different scenarios. Each entry in the table is the value of $\Gamma_{\mathrm{d}}{}^{\tau}$ for the corresponding d, τ and scenario. Larger values mean higher distributions separation, and in turn better effectiveness is expected.

The number of distinct explored ECs is shown in Figure 7a: observe that Ctd and ε-congruence-type metrics show better results when compared to the standard metrics. In addition, we expect that ε_2 and Ctd will be more effective than ε_∞. Furthermore, ΣL_2 shows better results than $\max L_2$.

As described in Figure 8a, the effectiveness of the metrics correlates with the analysis of Section VI. As expected, ε_2, ε_∞ and Ctd are more effective than ΣL_2 and $\max L_2$.

Partitions substructure. For the distributions separation we use $\tau = 1$. The values of $\Gamma_{\mathrm{d}}{}^{\tau}$ are given in Table I. $\max L_2$ has the largest value, then come ε_2, ε_∞ and Ctd, while ΣL_2 is far behind.

Figure 7b shows the number of distinct explored ECs. $\max L_2$ shows the best results, ε_2 and ε_∞ have comparable results, which are better than Ctd, and ΣL_2 yields the poorest results.

For this scenario, by looking at the results of the experiments described in Section VI, one can foresee that $\max L_2$, ε_∞ and ε_2 will be more effective than Ctd, which in turn, will be more effective than ΣL_2. This is indeed the case when measuring the effectiveness of the planner, as can be seen in Figure 8b.

Pebbles substructure. For the calculation of $\Gamma_{\mathrm{d}}{}^{\tau}$ we use $\tau = 7$. The values are given in Table I. The best value is achieved by Ctd, then ε_2 and ΣL_2 have comparable values, then comes $\max L_2$ and finally ε_∞ with the smallest value.

The number of distinct explored ECs is shown in Figure 7c. Here again, the largest number of explored ECs is achieved with Ctd, followed by ε_2 and ΣL_2. Then ε_∞, and the lowest value is for $\max L_2$.

The effectiveness of the planner incorporated with each metric is expressed in Figure 8c. The results are with accordance to the analysis: Ctd is the most effective metric, ΣL_2 and ε_2 have comparable effectiveness, and ε_∞ and $\max L_2$ are the less effective metrics.

C. Putting it all together

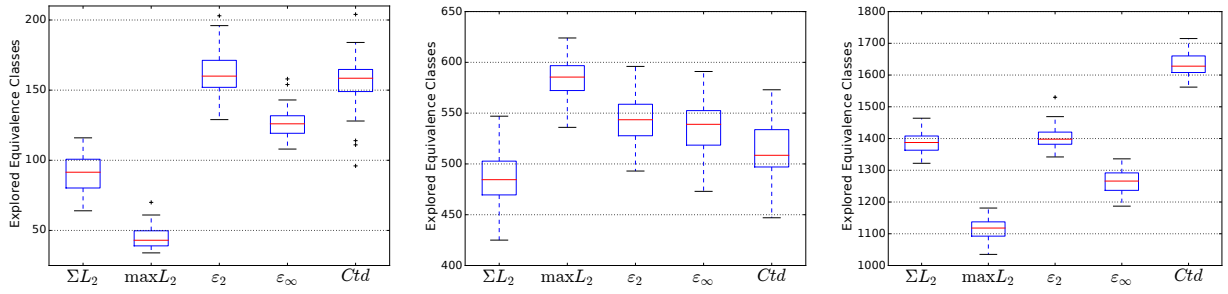
The $\mathcal{C}$-space of a general MRMP problem may consist of several substructures. This is the case for the scenario depicted in Figure 9a, which contains $m = 8$ robots. Figure 9c shows the effectiveness of planning with each metric. As can be inferred from the results, even in more general scenarios, the novel metrics are more effective than the standard ones. In some cases, it may be beneficial to alternate between several metrics—the planner maintains several nearest-neighbors data-structures, each for a metric. Each time the tree is expanded, a different data-structure is used in a round-robin fashion.

We have tested the scenario depicted in Figure 9a with 4, 6 and 8 robots (for 4 and 6 robots we eliminate from the scenario the robots $r_5, \ldots, r_8$ and r_7, r_8 respectively). We used each of the five metrics, along with all the combinations of two out of the five (total of 15) metrics. For the scenario with $m = 4$ robots, the effectiveness of all the metrics and their alternation was comparable. The results for the scenario with $m = 6$ robots (see Figure 9b) support the fact that it may be better to alternate between different metrics. Note the interesting fact that when alternating between ε_2 and ΣL_2 or Ctd, better effectiveness is obtained than when using each metric solely. For the scenario with $m = 8$ robots (Figure 9c) the novel metrics are more effective when compared to the standard ones. Alternating between novel and standard metrics does not make the planner more effective for the case of 8 robots. As we move from 4 robots (easier) to 8 robots (considerably harder), the effectiveness of the metrics becomes more noticeable.

VIII. Conclusions and Future Work

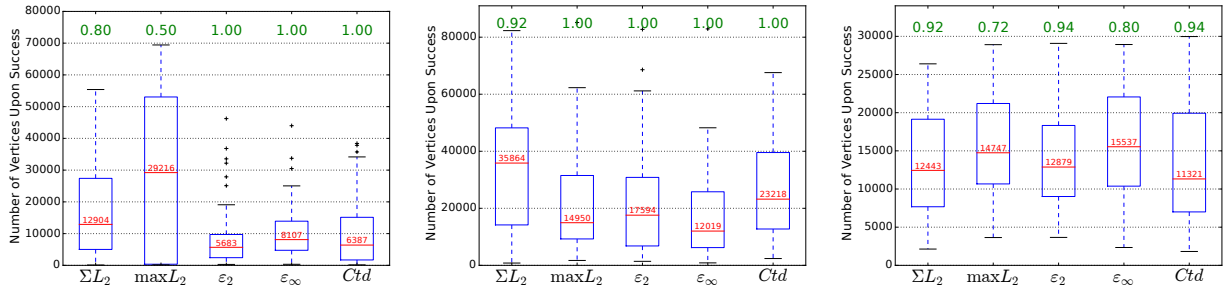
Our conclusion on how to effectively solve MRMP using sampling-based planners is to use tailored multi-robot metrics, possibly side-by-side with more traditional metrics. Already with three substructures we could test and suggest new effective metrics. There could be many more substructures, in particular larger, more elaborate ones. We plan to look for additional such substructures, which could help in finding more relevant metrics and useful combinations thereof.

Metrics are relevant for other settings of MRMP, including moving rigid bodies in 3D, and robots with differential constraints. The proposed metrics and analysis tools can be extended to such settings as well. In fact, we already applied our ideas to simple settings with robots that can translate and rotate. Another notable variant is the *unlabeled* setting in which all the robots are identical and interchangeable. There are similarity measures for unlabeled point sets that can be adapted for MRMP [2, 6, 14, 20, 24]. Unlabeled planning involves matching functions as well, which have common properties with metrics but make the problem considerably harder. We began to explore the unlabeled case, and have some promising initial results in this direction as well. A demonstration of our initial results for rotating robots and unlabeled disc robots is provided in the extended version of this paper.



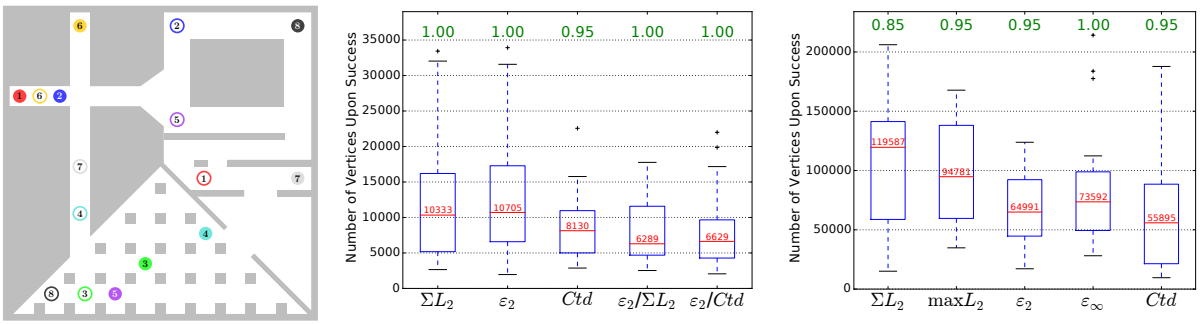
(a) Permutations Substructure (b) Partitions Substructure (c) Pebbles Substructure

Fig. 7. Different explored equivalence classes experiment. A dRRT tree is expanded until it contains 10,000 vertices. For each vertex in the tree we find its representative EC, and count the number of different ECs (denoted by $|\mathcal{U}_{\mathrm{d}}/\mathrm{EC}|$). Higher value means that we expect the metric to be more effective. The experiment is repeated 50 times for each metric. The figure depicts the values of $|\mathcal{U}_{\mathrm{d}}/\mathrm{EC}|$ for each metric.



(a) Permutations Substructure (b) Partitions Substructure (c) Pebbles Substructure

Fig. 8. Number of expanded vertices when a solution is found. The experiment is repeated 50 times per metric. The planner success rate is depicted in the green labels on top of each boxplot. The red labels are the median value. Effectiveness is expressed by high success rate and low number of vertices.



(a) (b) Effectiveness with 6 robots (c) Effectiveness with 8 robots

Fig. 9. A general scenario. We test the scenario with 8 robots, and the scenario with 4 or 6 robots which we get by eliminating $r_5, \ldots, r_8$ or r_7, r_8, respectively. (a) Start and goal configuration, drawn in solid and empty discs, respectively. (b) Effectiveness of metrics and alternation between metrics summarized over 20 runs for the case of 6 robots. As in the previous plots, the green labels indicate the success rate. (c) Effectiveness of each metric summarized over 20 runs of the planner for the case of 8 robots.

In this work we assessed metrics using RRT-style planners, such as dRRT (see Section III). Although we do not believe that our reported results are biased towards these specific types of planners, it would be interesting to see whether the conclusions can be reproduced for other planners, that operate differently than RRT, e.g., PRM*, RRT* [27] and FMT* [25]. This also leads to the question of the effect metrics have on the quality of the solution in MRMP.

Currently, we experimentally fine-tune the parameter τ. It will be interesting to come up with a theoretical analysis of the choice of this parameter.

References

[1] Aviv Adler, Mark de Berg, Dan Halperin, and Kiril Solovey. Efficient multi-robot motion planning for unlabeled discs in simple polygons. *IEEE Trans. Automation Science and Engineering*, 12(4):1309–1317, 2015.

[2] Helmut Alt, Kurt Mehlhorn, Hubert Wagener, and Emo Welzl. Congruence, similarity, and symmetries of geometric objects. *Discrete & Computational Geometry*, 3:237–256, 1988.

[3] Nancy M. Amato, O. Burçhan Bayazit, Lucia K. Dale, Christopher Jones, and Daniel Vallejo. Choosing good distance metrics and local planners for probabilistic roadmap methods. In *IEEE International Conference on Robotics and Automation*, pages 630–637, 1998.

[4] Aaron F. Archer. A modern treatment of the 15 puzzle. *The American Mathematical Monthly*, 106(9):793–799, 1999.

[5] Daman Bareiss and Jur van den Berg. Generalized reciprocal collision avoidance. *I. J. Robotics Res.*, 34(12):1501–1514, 2015.

[6] Serge J. Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, 2002.

[7] Mukunda Bharatheesha, Wouter Caarls, Wouter Jan Wolfslag, and Martijn Wisse. Distance metric approximation for state-space RRTs using supervised learning. In *IEEE/RSJ International Conference on*

Intelligent Robots and Systems, pages 252–257, 2014.

[8] Alexandre Boeuf, Juan Cortés, Rachid Alami, and Thierry Siméon. Enhancing sampling-based kinodynamic motion planning for quadrotors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2447–2452, 2015.

[9] Stina S. Bridgeman and Roberto Tamassia. Difference metrics for interactive orthogonal graph drawing algorithms. *J. Graph Algorithms Appl.*, 4(3):47–74, 2000.

[10] Sergey Brin. Near neighbor search in large metric spaces. In *International Conference on Very Large Data Bases*, pages 574–584, 1995.

[11] Edgar Chávez, Gonzalo Navarro, Ricardo A. Baeza-Yates, and José L. Marroquín. Searching in metric spaces. *ACM Comput. Surv.*, 33(3): 273–321, 2001.

[12] Howie Choset, Kevin Lynch, Seth Hutchinson, George Kantor, Wolfram Burgard, Lydia E. Kavraki, and Sebastian Thrun. *Principles of robot motion: theory, algorithms, and implementations*. MIT Press, 2005.

[13] Paolo Ciaccia, Marco Patella, and Pavel Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *International Conference on Very Large Data Bases*, pages 426–435, 1997.

[14] Alon Efrat and Alon Itai. Improvements on bottleneck matching and related problems using geometry. In *Symposium on Computational Geometry*, pages 301–310, 1996.

[15] Chinwe Ekenna, Sam Ade Jacobs, Shawna L. Thomas, and Nancy M. Amato. Adaptive neighbor connection for PRMs: A natural fit for heterogeneous environments and parallelism. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1249–1256, 2013.

[16] Jacob E. Goodman and Richard Pollack. On the combinatorial classification of nondegenerate configurations in the plane. *J. Comb. Theory, Ser. A*, 29(2):220–235, 1980.

[17] Jacob E. Goodman and Richard Pollack. Multidimensional sorting. *SIAM J. Comput.*, 12(3):484–507, 1983.

[18] Google. Google compute engine, 2017. URL https://cloud.google.com/compute/.

[19] Dan Halperin, Lydia Kavraki, and Kiril Solovey. Robotics. In Jacob E. Goodman, Joseph O'Rourke, and Csaba D. Toth, editors, *Handbook of Discrete and Computational Geometry*, chapter 51. CRC Press LLC, 3rd edition, 2016. URL http://www.csun.edu/~ctoth/Handbook/HDCG3.html.

[20] Felix Hausdorff. *Mengenlehre*. Springer, Berlin, 3rd edition, 1927.

[21] Robert A. Hearn and Erik D. Demaine. PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. *Theor. Comput. Sci.*, 343(1-2): 72–96, 2005.

[22] John E. Hopcroft, Jacob T. Schwartz, and Micha Sharir. On the complexity of motion planning for multiple independent objects; PSPACE-hardness of the "Warehouseman's problem". *I. J. Robotics Res.*, 3(4): 76–88, 1984.

[23] David Hsu, Jean-Claude Latombe, and Rajeev Motwani. Path planning in expansive configuration spaces. *Int. J. Comput. Geometry Appl.*, 9 (4/5):495–512, 1999.

[24] Daniel P. Huttenlocher, Klara Kedem, and Micha Sharir. The upper envelope of Voronoi surfaces and its applications. *Discrete & Computational Geometry*, 9:267–291, 1993.

[25] Lucas Janson, Edward Schmerling, Ashley A. Clark, and Marco Pavone. Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions. *I. J. Robotics Res.*, 34(7):883–921, 2015.

[26] Jeffrey Kane Johnson. A novel relationship between dynamics and complexity in multi-agent collision avoidance. In *Robotics: Science and Systems*, Ann Arbor, Michigan, June 2016.

[27] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *I. J. Robotics Res.*, 30(7):846–894, 2011.

[28] David G. Kendall. Shape manifolds, procrustean metrics and complex projective spaces. *Bull. London Math. Soc.*, 16(2):81–121, 1984.

[29] Stephen Kloder and Seth Hutchinson. Path planning for permutation-invariant multi-robot formations. In *IEEE International Conference on Robotics and Automation*, pages 1797–1802, 2005.

[30] James J. Kuffner. Effective sampling and distance metrics for 3D rigid body path planning. In *IEEE International Conference on Robotics and Automation*, pages 3993–3998, 2004.

[31] Steve M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006.

[32] Steven M. LaValle and James J. Kuffner. Randomized kinodynamic planning. In *IEEE International Conference on Robotics and Automation*, pages 473–479, 1999.

[33] Stephane Leroy, Jean-Paul Laumond, and Thierry Siméon. Multiple path coordination for mobile robots: a geometric algorithm. In *International Joint Conference on Artificial Intelligence*, pages 1118–1123, 1999.

[34] Kelly A. Lyons, Henk Meijer, and David Rappaport. Algorithms for cluster busting in anchored graph drawing. *J. Graph Algorithms Appl.*, 2(1), 1998.

[35] Marco Morales, Lydia Tapia, Roger Pearce, Samuel Rodriguez, and Nancy M. Amato. A machine learning approach for feature-sensitive motion planning. In *Algorithmic Foundations of Robotics*, pages 361–376, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[36] Luigi Palmieri and Kai Oliver Arras. Distance metric learning for RRT-based motion planning with constant-time inference. In *IEEE International Conference on Robotics and Automation*, pages 637–643, 2015.

[37] Amit Pamecha, Imme Ebert-Uphoff, and Gregory S. Chirikjian. Useful metrics for modular robot motion planning. *IEEE Trans. Robotics and Automation*, 13(4):531–545, 1997.

[38] Erion Plaku and Lydia E. Kavraki. Quantitative analysis of nearest-neighbors search in high-dimensional sampling-based motion planning. In *Algorithmic Foundation of Robotics*, pages 3–18, 2006.

[39] Oren Salzman, Michael Hemmer, and Dan Halperin. On the power of manifold samples in exploring configuration spaces and the dimensionality of narrow passages. *IEEE T. Automation Science and Engineering*, 12(2):529–538, 2015.

[40] Gilardo Sanchez and Jean-Claude Latombe. On delaying collision checking in PRM planning – application to multi-robot coordination. *I. J. Robotics Res.*, 21:5–26, 2002.

[41] Kiril Solovey and Dan Halperin. k-Color multi-robot motion planning. *I. J. Robotics Res.*, 33(1):82–97, 2014.

[42] Kiril Solovey and Dan Halperin. On the hardness of unlabeled multi-robot motion planning. In *Robotics: Science and Systems*, 2015.

[43] Kiril Solovey, Jingjin Yu, Or Zamir, and Dan Halperin. Motion planning for unlabeled discs with optimality guarantees. In *Robotics: Science and Systems*, 2015.

[44] Kiril Solovey, Oren Salzman, and Dan Halperin. Finding a needle in an exponential haystack: Discrete RRT for exploration of implicit roadmaps in multi-robot motion planning. *I. J. Robotics Res.*, 35(5): 501–513, 2016.

[45] Paul Spirakis and Chee-Keng Yap. Strong NP-hardness of moving many discs. *Inf. Process. Lett.*, 19(1):55–59, 1984.

[46] Ioan Alexandru Sucan, Mark Moll, and Lydia E. Kavraki. The open motion planning library. *IEEE Robot. Automat. Mag.*, 19(4):72–82, 2012.

[47] Matthew Turpin, Nathan Michael, and Vijay Kumar. Concurrent assignment and planning of trajectories for large teams of interchangeable robots. In *IEEE International Conference on Robotics and Automation*, pages 842–848, 2013.

[48] Jur van den Berg and Mark H. Overmars. Prioritized motion planning for multiple robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 430 – 435, 2005.

[49] Remco C. Veltkamp and Michiel Hagedoorn. State of the art in shape matching. In *Principles of Visual Information Retrieval*, pages 87–119. Springer, 2001.

[50] Petr Švestka and Mark H. Overmars. Coordinated path planning for multiple robots. *Robotics and Autonomous Systems*, 23(3):125–152, 1998.

[51] Glenn Wagner and Howie Choset. Subdimensional expansion for multirobot path planning. *Artif. Intell.*, 219:1–24, 2015.

[52] Lutz Winkler, Heinz Wörn, and Adrian Friebel. A distance and diversity measure for improving the evolutionary process of modular robot organisms. In *IEEE International Conference on Robotics and Biomimetics*, pages 2102–2107, 2011.

[53] Viktor Zykov, Efstathios Mytilinaios, Mark Desnoyer, and Hod Lipson. Evolved and designed self-reproducing modular robotics. *IEEE Trans. Robotics*, 23(2):308–319, 2007.