Scalable Asymptotically-Optimal Multi-Robot Motion Planning

Andrew Dobson   Kiril Solovey   Rahul Shome   Dan Halperin   Kostas E. Bekris

***Abstract*— Discovering high-quality paths for multi-robot problems can be achieved, in principle, by exploring the composite space of all robots. For instance, sampling-based algorithms that build either roadmaps or tree data structures achieve asymptotic optimality. The hardness of motion planning, however, which implies an exponential dependence on problem dimensionality, renders the explicit construction of such structures in the composite space of all robots impractical. This work proposes a scalable, sampling-based planner for coupled multi-robot problems that provides desirable path-quality guarantees. The proposed `dRRT`* is an informed, asymptotically-optimal extension of a prior method `dRRT`, which introduced the idea of building roadmaps for each robot and implicitly searching the tensor product of these structures in the composite space. The paper describes the conditions for convergence to optimal paths in multi-robot problems, which is not feasible for the prior method. Moreover, simulations indicate `dRRT`* converges to high-quality paths and scales to higher numbers of robots where various alternatives fail. It can also be used on high-dimensional challenges, such as planning for robot manipulators.**

I. Introduction and Prior Work

Many multi-robot planning applications [24], [5], [4] require high-dimensional platforms to simultaneously move in a shared workspace, where high-quality paths must be computed quickly, as demonstrated in Fig. 1. Preprocessing given knowledge of the static scene can help the online computation of high-quality paths. For example, sampling-based roadmaps help with such high-dimensional instances and provide primitives for preprocessing a static scene [9], [10]. These methods converge to optimal solutions given sufficient density [8], i.e., at least $O(n \log(n))$ edges are needed for a roadmap with n vertices, while near-optimality is achieved after finite computation [3], [7] or asymptotically [13], [1].

Naïvely constructing a sampling-based roadmap or tree in the composite configuration space of multiple robots provides asymptotic optimality but is not a scalable solution. In particular, memory requirements depend exponentially on the problem's dimensionality [17]. The alternative is decoupled planning, where paths for robots are computed independently and then coordinated [11]. These methods, however, typically lack completeness and optimality guarantees. While some hybrid approaches can automatically take advantage of the inherent decoupling between robots and provide guarantees [23], they are often limited to discrete domains. The problem is more complex when the robots exhibit non-trivial dynamics [14]. Collision avoidance or control methods can scale to many robots, but typically lack global path quality guarantees [22], [21].

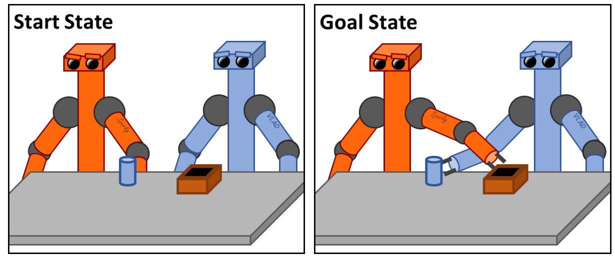

Fig. 1. Simultaneous planning for multiple high-dimensional systems is a difficult, motivating challenge for this work.

The previously proposed `dRRT` approach [18] is a scalable sampling-based approach, which is probabilistically complete. It searches an implicit tensor product of roadmaps, which are constructed independently for each robot [20]. The current work proposes `dRRT`* and shows that it is an efficient asymptotically optimal extension of the previous method. The `dRRT`* framework is an anytime algorithm, which quickly finds initial solutions and then refines them, while ensuring asymptotic convergence to optimal solutions. Simulations show that the method practically generates high-quality paths while scaling to complex, high-dimensional problems, where alternatives fail.

II. Problem Setup and Notation

Consider a shared workspace with $R \geq 2$ robots, each operating in a configuration space $\mathbb{C}_i$ for $1 \leq i \leq R$. Let $\mathbb{C}_i^{\mathrm{f}} \subset \mathbb{C}_i$ be each robot's free space, where it does not collide with the static scene, and $\mathbb{C}_i^{\mathrm{o}} = \mathbb{C}_i \setminus \mathbb{C}_i^{\mathrm{f}}$ is the obstacle space for robot i. The

A. Dobson, R. Shome and K. Bekris are with the Computer Science Dept. of Rutgers Univ., NJ, USA, {chuples.dobson, rahul.shome, kostas.bekris}@cs.rutgers.edu. They were supported by NSF IIS 1451737, IIS-1617744 and CCF 1330789.
K. Solovey and D. Halperin are with the Computer Science Dept. of Tel Aviv University, Israel, {kirilsol,danha}@post.tau.ac.il. Their work has been supported in part by the Israel Science Foundation (grant no. 825/15) and by the Blavatnik Computer Science Research Fund. K. S. has also been supported by the Clore Israel Foundation.

composite configuration space $\mathbb{C} = \prod_{i=1}^{R} \mathbb{C}_i$ is the Cartesian product of each robot's configuration space. A composite configuration $Q = (q_1, \ldots, q_R) \in \mathbb{C}$ is an R-tuple of robot configurations. For two distinct robots i, j, denote by $I_i^j(q_j) \subset \mathbb{C}_i$ the set of configurations of robot i, which lead into collision with robot j at its configuration q_j. Then, the composite free space $\mathbb{C}^{\mathrm{f}} \subset \mathbb{C}$ consists of configurations $Q = (q_1, \ldots, q_R)$ subject to:

- $q_i \in \mathbb{C}_i^{\mathrm{f}}$ for every $1 \leq i \leq R$;
- $q_i \notin I_i^j(q_j), q_j \notin I_j^i(q_i)$ for every $1 \leq i < j \leq R$.

Each $Q \in \mathbb{C}^{\mathrm{f}}$ requires robots to not collide with obstacles or pairwise. The composite obstacle space is defined as $\mathbb{C}^{\mathrm{o}} = \mathbb{C} \setminus \mathbb{C}^{\mathrm{f}}$.

Given $S, T \in \mathbb{C}^{\mathrm{f}}$, where $S = (s_1, \ldots, s_R), T = (t_1, \ldots, t_R)$, a *trajectory* $\Sigma : [0,1] \rightarrow \mathbb{C}^{\mathrm{f}}$ is a continuous curve in $\mathbb{C}^{\mathrm{f}}$, such that $\Sigma(0) = S, \Sigma(1) = T$, where the R robots move simultaneously. Σ is an R-tuple $(\sigma_1, \ldots, \sigma_R)$ of robot paths, such that $\sigma_i : [0,1] \rightarrow \mathbb{C}_i^{\mathrm{f}}$.

The objective is to find a trajectory, which minimizes a *cost function* $c(\cdot)$. The analysis assumes the cost is the sum of robot path lengths, i.e., $c(\Sigma) = \sum_{i=1}^{R} |\sigma_i|$, where $|\sigma_i|$ denotes the standard *arc length* of σ_i. The arguments also work for $\max_{i=1:R} |\sigma_i|$.[1] This work provides sufficient conditions for dRRT* to converge to optimal trajectories over the cost function c.

III. Methods for Composite Space Planning

First construct for every robot $i \in [1, R]$ a roadmap $\mathbb{G}_i = (\mathbb{V}_i, \mathbb{E}_i)$ over $\mathbb{C}_i^{\mathrm{f}}$ according to the asymptotically optimal PRM* [8], i.e., two roadmap nodes/configurations are connected with an edge if they are within radius $r(|\mathbb{V}_i|)$ and the local path connecting them is in $\mathbb{C}_i^{\mathrm{f}}$. Then: $\hat{\mathbb{G}} = (\hat{\mathbb{V}}, \hat{\mathbb{E}}) = \mathbb{G}_1 \times \ldots \times \mathbb{G}_R$ is the *tensor product roadmap* in $\mathbb{C}$ (see Figure 2). Formally, $\hat{\mathbb{V}} = \{(v_1, v_2, \ldots, v_R) : \forall\ i,\ v_i \in \mathbb{V}_i\}$ is the Cartesian product of the nodes from each roadmap $\mathbb{G}_i$. For two vertices $V = (v_1, \ldots, v_m) \in \hat{\mathbb{V}}, V' = (v_1', \ldots, v_m') \in \hat{\mathbb{V}}$, the edge set $\hat{\mathbb{E}}$ contains edge (V, V') if $\forall i \in [1, R] :\ v_i = v_i'$ or $(v_i, v_i') \in \mathbb{E}_i$.[2]

Similar to the previous dRRT technique, the proposed dRRT* expands a tree of paths over the tensor product roadmap $\hat{\mathbb{G}}$. In contrast to dRRT, however:

- dRRT* performs a rewiring step to refine paths in the tree, reducing costs to reach particular nodes.
- dRRT* is anytime, employing branch and bound pruning after an initial solution is reached.
- dRRT* uses a heuristic to guide its expansion.

[1]The types of distances the arguments hold are more general, but proofs for alternative metrics are left as future work.

[2]Notice this slight difference from dRRT [18] so as to allow edges where some robots remain motionless while others move.

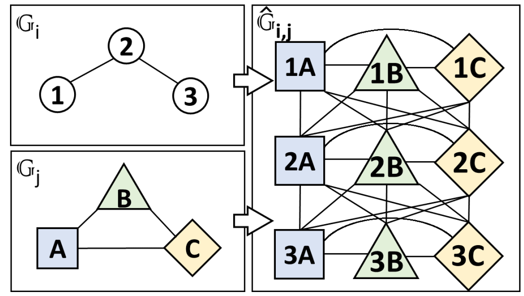

Fig. 2. An illustration of a two-robot tensor product roadmap $\hat{\mathbb{G}}_{i,j}$ between roadmaps $\mathbb{G}_i$ and $\mathbb{G}_j$. Two nodes in the tensor-product roadmap share an edge if all the individual robot configurations share an edge in the individual robot roadmaps.

Algorithm 1 outlines dRRT*, which grows a tree $\mathbb{T} = (\mathbb{V}_{\mathbb{T}}, \mathbb{E}_{\mathbb{T}})$ over $\hat{\mathbb{G}}$, rooted at start configuration S and initializes path π_{best} (Line 1). The method stores the node added at each iteration V (Line 2), so as to guide the expansion of $\mathbb{T}$ towards the goal. The method iteratively expands $\mathbb{T}$ given a time budget (Line 3), according to function Expand_dRRT* (Line 4) as detailed in Algorithm 2, and stores the newly added node V. Then, the method tries to trace the path π that connects the start S with the target T (Line 5). If such a path π is found, it is stored in π_{best}, as long as it improves upon the cost of the previous solution (Lines 6, 7). The best path π_{best} is returned (Line 8).

Algorithm 1: dRRT*$(\hat{\mathbb{G}}, S, T)$

1 $\pi_{\text{best}} \leftarrow \emptyset$, $\mathbb{T}$.init(S);
2 $V \leftarrow S$;
3 **while** time.elapsed() $<$ time_limit **do**
4 $\quad V \leftarrow$ Expand_dRRT*$(\hat{\mathbb{G}}, \mathbb{T}, V, T)$;
5 $\quad \pi \leftarrow$ Trace_Path$(\mathbb{T}, S, T)$;
6 $\quad$ **if** $\pi \neq \emptyset$ *and cost*$(\pi) <$ *cost*(π_{best}) **then**
7 $\quad\quad \pi_{\text{best}} \leftarrow \pi$;
8 **return** π_{best}

Algorithm 2 outlines the expansion step. The default behavior is summarized in Lines 1-4, i.e., when no V^{last} is passed as argument (Line 1). This operation corresponds to an exploration step similar to RRT, i.e., a random sample Q^{rand} is generated in $\mathbb{C}$ (Line 2) and its nearest neighbor V^{near} in $\mathbb{T}$ is found (Line 3); then, the oracle function $\mathbb{O}_d(\cdot, \cdot)$ returns the implicit graph node V^{new} that is a neighbor of V^{near} on the implicit graph in the direction of Q^{rand} (Line 4). If V^{last} is provided (Line 5), which happens when the last iteration generates a node closer to the goal relative to its parent, then V^{new} is greedily generated so as to be a neighbor of V^{last} in the direction of the goal T (Line 6).

In either case, the method next finds neighbors N, which are adjacent to V^{new} in $\hat{\mathbb{G}}$ and have also been

Algorithm 2: $\texttt{Expand\_dRRT}^*(\hat{\mathbb{G}}, \mathbb{T}, V^{\text{last}}, T)$

1 **if** $V^{\text{last}} == NULL$ **then**
2 $\quad Q^{\text{rand}} \leftarrow \texttt{Random\_Sample}()$;
3 $\quad V^{\text{near}} \leftarrow \texttt{Nearest\_Neighbor}(\mathbb{T}, Q^{\text{rand}})$;
4 $\quad V^{\text{new}} \leftarrow \mathbb{O}_{\text{d}}(V^{\text{near}}, Q^{\text{rand}})$;
5 **else**
6 $\quad V^{\text{new}} \leftarrow \mathbb{O}_{\text{d}}(V^{\text{last}}, T)$;
7 $N \leftarrow \texttt{Adjacent}(V^{\text{new}}, \hat{\mathbb{G}}) \cap \mathbb{V}_{\mathbb{T}}$;
8 $V^{\text{best}} \leftarrow \operatorname{argmin}_{v \in N, \mathbb{L}(v, V^{\text{new}}) \subset \mathbb{C}^{\text{f}}} c(v) + c(\mathbb{L}(v, V^{\text{new}}))$;
9 **if** $V^{\text{best}} == NULL$ **then**
10 $\quad$ **return** $NULL$;
11 **if** $V^{\text{new}} \notin \mathbb{T}$ **then**
12 $\quad \mathbb{T}.\texttt{Add\_Vertex}(V^{\text{new}})$;
13 $\quad \mathbb{T}.\texttt{Add\_Edge}(V^{\text{best}}, V^{\text{new}})$;
14 **else**
15 $\quad \mathbb{T}.\texttt{Rewire}(V^{\text{best}}, V^{\text{new}})$;
16 **for** $v \in N$ **do**
17 $\quad$ **if** $c(V^{\text{new}}) + c(\mathbb{L}(V^{\text{new}}, v)) < c(v)$ *and* $\mathbb{L}(V^{\text{new}}, v) \subset \mathbb{C}^{\text{f}}$ **then**
18 $\quad\quad \mathbb{T}.\texttt{Rewire}(V^{\text{new}}, v)$;
19 **if** $h(V^{\text{new}}) < h(V^{\text{best}})$ **then**
20 $\quad$ **return** V^{new};
21 **else**
22 $\quad$ **return** $NULL$;

added to $\mathbb{T}$ (Line 7). Among N, the best node V^{best} is chosen, for which the local path $\mathbb{L}(V^{\text{best}}, V^{\text{new}})$ is collision-free and that the total path cost to V^{new} is minimized (Line 8). If no such parent can be found (Line 9), the expansion fails and no node is returned (Line 10). Then, if V^{new} is not in $\mathbb{T}$, it is added (Lines 11-13). Otherwise, if it exists, the tree is rewired so as to contain edge $(V^{\text{best}}, V^{\text{new}})$, and the cost of the V^{new}'s sub-tree (if any) is updated (Lines 14, 15). Then, for all nodes in N (Line 16), the method tests whether $\mathbb{T}$ should be rewired through V^{new} to reach this neighbor. Given that $\mathbb{L}(V^{\text{new}}, v)$ is collision-free and is of lower cost than the existing path to v (Line 17), the tree is rewired to make V^{new} be the parent of v (line 18).

Finally, if the heuristic for V^{new} is lower than its parent V^{best} (line 19), the method returns V^{new} (Line 20), prompting greedy expansion of V^{new} during the next iteration. Otherwise, $NULL$ is returned, prompting exploration. The implementation employs branch-and-bound, which is not reflected in the algorithmics.

V^{new} is determined via the dRRT oracle function, which in conjunction with a simple rewiring scheme is sufficient for showing asymptotic optimality for dRRT* (see Section IV). The oracle function $\mathbb{O}_d$ for a two-robot case is illustrated in Figure 3. First, let $\rho(Q, Q')$ be the ray from configuration Q terminating at Q'. Then, denote $\angle_Q(Q', Q'')$ as the minimum angle between $\rho(Q, Q')$ and $\rho(Q, Q'')$. When Q^{rand} is drawn in $\mathbb{C}$, its nearest neighbor V^{near} in $\mathbb{T}$ is found. Then, project the points Q^{rand} and V^{near} into each robot space $\mathbb{C}_i$, i.e., ignore the configurations of other robots.

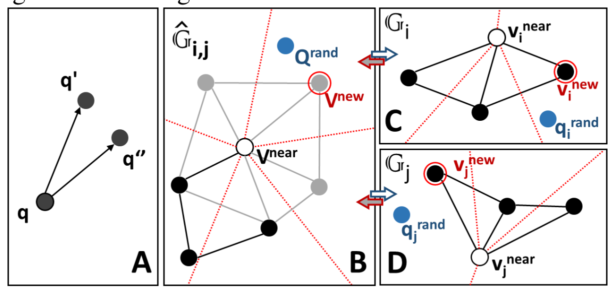

Fig. 3. (A) The method reasons over all neighbors q' of q so as to minimize the angle $\angle_q(q', q'')$. (B) $\mathbb{O}_d(\cdot,\cdot)$ finds graph vertex V^{new} by minimizing angle $\angle_{V^{\text{near}}}(V^{\text{new}}, Q^{\text{rand}})$. (C,D) $V^{\text{near}}$and $Q^{\text{rand}}$are projected into each robot's $\mathbb{C}$-space so as to find nodes $v_{\text{i}}^{\text{new}}$ and $v_{\text{j}}^{\text{new}}$, respectively, which minimize angle $\angle_{v_{\text{i/j}}^{\text{near}}}(v_{\text{i/j}}^{\text{new}}, q_{\text{i/j}}^{\text{rand}})$.

The method separately searches the single-robot roadmaps to discover V^{new}. Denote $V^{\text{near}} = (v_1, \ldots, v_R)$ and $Q^{\text{rand}} = (\tilde{q}_1, \ldots, \tilde{q}_R)$. For every robot i, let $N_i \subset \mathbb{V}_i$ be the neighborhood of $v_i \in \mathbb{V}_i$, and identify $v'_i = \operatorname{argmin}_{v \in N_i} \angle_{v_i}(\tilde{q}_i, v)$. The oracle function returns node $V^{\text{new}} = (v'_1, \ldots, v'_R)$.

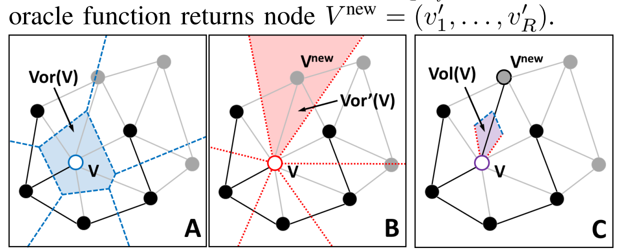

Fig. 4. (A) The Voronoi region $\text{Vor}(V)$ of vertex V is shown where if Q^{rand} is drawn, vertex V is selected for expansion. (B) When Q^{rand} lies in the directional Voronoi region $\text{Vor}'(V)$, the expand step expands to V^{new}. (C) Thus, when Q^{rand} is drawn within volume $\text{Vol}(V) = \text{Vor}(V) \cap \text{Vor}'(V)$, the method will generate V^{new} via V.

As in RRT and in dRRT, dRRT* has a Voronoi-bias property [12]. It is, however, slightly more involved to observe as shown in Figure 4. To generate an edge (V, V^{new}), random sample Q^{rand} must be drawn within the Voronoi cell of V, denoted as $\text{Vor}(V)$ (Figure 4(A)) and in the general direction of V^{new}, denoted as $\text{Vor}'(V)$ (Figure 4(B)). The intersection of these two volumes: $\text{Vol}(V) = \text{Vor}(V) \cap \text{Vor}'(V)$, is the volume to be sampled so as to generate V^{new} via V^{near} (Figure 4(C)).

IV. Analysis

This section examines the properties of dRRT* starting with the asymptotic convergence of the implicit roadmap $\hat{\mathbb{G}}$ to containing a path in $\mathbb{C}^{\text{f}}$ with optimum cost. Then, it is shown that dRRT* eventually discovers the shortest path in $\hat{\mathbb{G}}$. The combination of these two facts proves the asymptotic optimality of dRRT*.

For simplicity, the analysis considers robots operating in Euclidean space, i.e., $\mathbb{C}_i$ is a d-dim. Euclidean hypercube $[0,1]^d$ for fixed $d \geq 2$. Robots are also assumed to have the same number of degrees of freedom d. Furthermore, the analysis assumes the cost function of *total distance*, i.e., $|\Sigma| := \sum_{i=1}^{R} |\sigma_i|$. Discussion on these restrictions is provided in Section VI.

A. Optimal Convergence of $\hat{\mathbb{G}}$

For each robot, an asymptotically optimal PRM* roadmap $\mathbb{G}_i$ is constructed having $n = |V_i|$ samples and using a connection radius $r(n)$ necessary for asymptotic convergence to the optimum [8]. This work, similar to previous work [19], focuses on the notion of a robust optimum path and shows that the tensor product roadmap $\hat{\mathbb{G}}$ converges to such a path as n increases.

Definition 1: A trajectory $\Sigma : [0,1] \rightarrow \mathbb{C}^{\mathrm{f}}$ is *robust* if there exists a fixed $\delta > 0$ such that for every $\tau \in [0,1], X \in \mathbb{C}^{\mathrm{o}}$ it holds that $\|\Sigma(\tau) - X\|_2 \geq \delta$, where $\|\cdot\|_2$ denotes the standard Euclidean distance.

Definition 2: A value $c > 0$, which denotes a path cost, is robust if for every fixed $\epsilon > 0$, there exists a robust path Σ, such that $|\Sigma| \leq (1+\epsilon)c$. The *robust optimum* c^*, is the infimum over all such values.

For any fixed $n \in \mathbb{N}^+$, and a specific instance of $\hat{\mathbb{G}}$ constructed from R roadmaps, having n samples each, denote by $\Sigma^{(n)}$ the shortest path from S to T over $\hat{\mathbb{G}}$.

Definition 3: $\hat{\mathbb{G}}$ is asymp. optimal (AO) if for every fixed $\epsilon > 0$ it holds that $|\Sigma^{(n)}| \leq (1+\epsilon)c^*$ asymptotically almost surely [3], where the probability is over all the instantiations of $\hat{\mathbb{G}}$ with n samples for each PRM.

Using this definition, the following theorem is proven. Recall that d denotes the dimension of a single-robot configuration space.

Theorem 1: $\hat{\mathbb{G}}$ is AO when

$$r(n) \geq r^*(n) = (1+\eta)2\left(\frac{1}{d}\right)^{\frac{1}{d}}\left(\frac{\log n}{n}\right)^{\frac{1}{d}},$$

where η is any constant larger than 0.

Remark. Note that $r^*(n)$ was developed in [7, Theorem 4.1], and guarantees AO of PRM* for a single robot. The proof technique described in that work will be one of the ingredients used to prove Theorem 1.[4]

This proof shows that for a clearance-robust optimal path for any given $\epsilon > 0$ having cost c^*, there exists a robust trajectory $\Sigma : [0,1] \rightarrow \mathbb{C}^{\mathrm{f}}$, and a fixed $\delta > 0$, such that the cost of Σ is at most $(1+\epsilon/2)c^*$ and for every $X \in \mathbb{C}^{\mathrm{o}},\ \tau \in [0,1]$ it holds that $\|\Sigma(\tau) - X\| \geq \delta$. This is predicated on drawing a sufficient number of samples $n \geq n_0$. For the proposed proof, it is shown that $\hat{\mathbb{G}}$ contains a trajectory $\Sigma^{(n)}$, such that:

$$|\Sigma^{(n)}| \leq (1 + o(1)) \cdot |\Sigma|, \quad (1)$$

asymptotically almost surely (a.a.s.), which immediately implies that $|\Sigma^{(n)}| \leq (1+\epsilon)c^*$, proving Theorem 1.

Thus, it remains to show that there exists a trajectory on $\hat{\mathbb{G}}$, which satisfies Eq. 1 a.a.s.. As a first step, it will be shown that the robustness of $\Sigma = (\sigma_1, \ldots, \sigma_R)$ in the composite space implies robustness in the single-robot setting, i.e., robustness along σ_i.

For $\tau \in [0,1]$ define the forbidden space parameterized by τ as

$$\mathbb{C}_i^{\mathrm{o}}(\tau) = \mathbb{C}_i^{\mathrm{o}} \cup \bigcup_{j=1, j\neq i}^{R} I_i^j(\sigma_j(\tau)).$$

Claim 1: For every robot i, $\tau \in [0,1]$, and $q_i \in \mathbb{C}_i^{\mathrm{o}}(\tau)$, $\|\sigma_i(\tau) - q_i\|_2 \geq \delta$.

Proof: Fix a robot i, and fix some $\tau \in [0,1]$ and a configuration $q_i \in \mathbb{C}_i^{\mathrm{o}}(\tau)$. Next, define the following composite configuration

$$Q = (\sigma^1(\tau), \ldots, q_i, \ldots, \sigma^R(\tau)).$$

Note that it differs from $\Sigma(\tau)$ only in the i-th robot's configuration. By the robustness of Σ it follows that

$$\begin{aligned}\delta &\leq \|\Sigma(\tau) - Q\|_2 \\ &= \left(\|\sigma_i(\tau) - q_i\|_2^2 + \sum_{j=1, j\neq i}^{R} \|\sigma_j(\tau) - \sigma_j(\tau)\|_2^2\right)^{\frac{1}{2}} \\ &\leq \|\sigma_i(\tau) - q_i\|_2.\end{aligned}$$

■

The result of Claim 1 is that the paths $\sigma_1, \ldots, \sigma_R$ are robust in the sense that there is sufficient clearance for the individual robots to not collide with each other given a fixed location of a single robot. A Lemma is derived using proof techniques from the literature [7], and it implies every $\mathbb{G}_i$ contains a single-robot path $\sigma_i^{(n)}$ that converges to σ_i.

Lemma 1: For every robot i, $\mathbb{G}_i$ is constructed with n samples and a connection radius $r(n) \geq r^*(n)$ contains a path $\sigma_i^{(n)}$ with the following attributes a.a.s.:

(i) $\sigma_i^{(n)}(0) = s_i$, $\sigma_i^{(n)}(1) = t_i$;
(ii) $|\sigma_i^{(n)}| \leq (1 + o(1))|\sigma_i|$;
(iii) $\forall q \in \mathrm{Im}(\sigma_i^{(n)})$, $\exists \tau \in [0,1]$ s.t. $\|q - \sigma_i(\tau)\|_2 \leq r^*(n)$, where $\mathrm{Im}(\cdot)$ is some function image.

Proof: The first property (i) follows from the fact that s_i, t_i are directly added to $\mathbb{G}_i$. The rest follows from the proof of Theorem 4.1 in [7], which is applicable here since $r(n) \geq r^*(n)$. ■

Lemma 1 also implies that $\hat{\mathbb{G}}$ contains a path in $\mathbb{C}$, that represents robot-to-obstacle collision-free motions, and minimizes the multi-robot metric cost. In particular, define $\Sigma^{(n)} = (\sigma_1^{(n)}, \ldots, \sigma_R^{(n)})$, where $\sigma_i^{(n)}$ are obtained

[3] Let $A_1, A_2, \ldots$ be random variables in some probability space and let B be an event depending on A_n. B occurs *asymptotically almost surely* (a.a.s.) if $\lim_{n\rightarrow\infty} \Pr[B(A_n)] = 1$.

[4] Note that $r^*(n)$ can be refined to incorporate the proportion of $\mathbb{C}_i^{\mathrm{f}}$, which would reduce this expression.

from Lemma 1. Then

$$|\Sigma^{(n)}| = \sum_{i=1}^{R} |\sigma_i^{(n)}| \leq (1+o(1)) \sum_{i=1}^{R} |\sigma_i| \leq (1+o(1))|\Sigma|.$$

Nevertheless, it is not clear whether this ensures the existence of a path where robot-robot collisions are avoided. That is, although $\mathrm{Im}(\sigma_i^{(n)}) \subset \mathbb{C}_i^{\mathrm{f}}$, it might be the case that $\mathrm{Im}(\Sigma^{(n)}) \cap \mathbb{C}^{\mathrm{o}} \neq \emptyset$. Next, it is shown that $\sigma_1^{(n)}, \ldots, \sigma_R^{(n)}$ can be reparametrized to induce a composite-space path whose image is fully contained in $\mathbb{C}^{\mathrm{f}}$, with length equivalent to $\Sigma^{(n)}$.

For each robot i, denote by $V_i = (v_i^1, \ldots, v_i^{\ell_i})$ the chain of $\mathbb{G}_i$ vertices traversed by $\sigma_i^{(n)}$. For every $v_i^j \in V_i$ assign a timestamp τ_i^j of the closest configuration along σ^i, i.e.,

$$\tau_i^j = \operatorname*{argmin}_{\tau \in [0,1]} \|v_i^j - \sigma_i(\tau)\|_2.$$

Also, define $\mathcal{T}_i = (\tau_i^1, \ldots, \tau_i^{\ell_i})$ and denote by $\mathcal{T}$ the ordered list of $\bigcup_{i=1}^{R} \mathcal{T}_i$, according to the timestamp values. Now, for every i, define a global timestamp function $TS_i : \mathcal{T} \rightarrow V_i$, which assigns to each global timestamp in $\mathcal{T}$ a single-robot configuration from V_i. It thus specifies in which vertex robot i resides at time $\tau \in \mathcal{T}$. For $\tau \in \mathcal{T}$, let j be the largest index, such that $\tau_i^j \leq \tau$. Then simply assign $TS_i(\tau) = \tau_i^j$. From property (iii) in Lemma 1 and Claim 1 it follows that no robot-robot collisions are induced by the reparametrization, concluding the proof of Theorem 1.

B. *Asymptotic Optimality of* dRRT*

Finally, dRRT* is shown to be AO. Denote by m the time budget in Algorithm 1, i.e., the number of iterations of the loop. Denote by $\Sigma^{(n,m)}$ the solution returned by dRRT* for n and m.

Theorem 2: If $r(n) > r^*(n)$ then for every fixed $\epsilon > 0$ it holds that

$$\lim_{n,m \rightarrow \infty} \Pr\left[|\Sigma^{(n,m)}| \leq (1+\epsilon)c^*\right] = 1.$$

Since $\hat{\mathbb{G}}$ is AO (Theorem 1), it suffices to show that for any fixed n, and a fixed instance of $\hat{\mathbb{G}}$, defined over R PRMs with n samples each, dRRT* eventually (as m tends to infinity), finds the optimal trajectory over $\hat{\mathbb{G}}$. This can be shown using the properties of a Markov chain with absorbing states [6, Theorem 11.3]. While a full proof is omitted here, the idea is similar to what is presented in previous work [18, Theorem 3], and examined in Appendix A. By restricting the states of the Markov chain to being the graph vertices along the optimal path, setting the target vertex to be an absorbing vertex, and showing that the probability of transitioning along any edge in this path is nonzero (i.e., the probability is proportional to $\frac{\mu(\mathrm{Vol}(V_k))}{\mu(\mathbb{C}^{\mathrm{f}})} > 0$), then the probability that this process does not reach the target state along the optimal path converges to 0 as the number of dRRT* iterations tends to infinity. The final step is to show that the above statements hold when both m and n tend to ∞. A proof for this phenomenon can be found in [18, Theorem 6].

V. Experimental Validation

This section provides an experimental evaluation of dRRT* by demonstrating practical convergence, scalability for disk robots, and applicability to dual-arm manipulation. The approach and alternatives are executed on a cluster with Intel(R) Xeon(R) CPU E5-4650 @ 2.70GHz processors, and 128GB of RAM. [5]

2 Disk Robots among 2D Polygons: This base-case test involves 2 disks ($\mathbb{C}_i := \mathbb{R}^2$) of radius 0.2 with bounded velocity, in a 10×10 region, inflated by the radius, as in Figure 5. The disks have to swap positions between $(0, 0)$ and $(9, 9)$. This is a setup where it is possible to compute the explicit roadmap, which is not practical in more involved scenarios. In particular, dRRT* is tested against: a) running A* on the implicit tensor roadmap $\hat{\mathbb{G}}$ (referred to as "Implicit A*"), where $\hat{\mathbb{G}}$ is defined over the same individual roadmaps with N nodes as those used by dRRT*; and b) an explicitly constructed PRM* roadmap with N^2 nodes in $\mathbb{C}$.

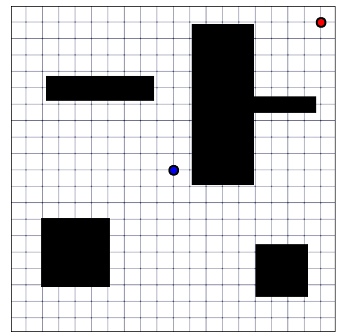

Fig. 5. The 2D environment where the 2 disk robots operate.

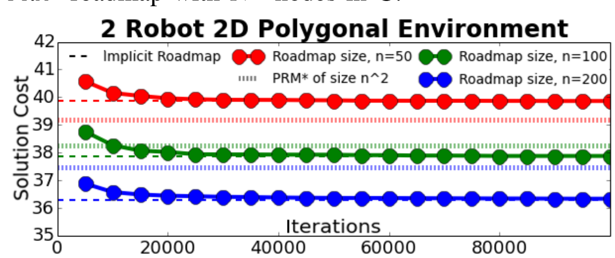

Fig. 6. For every n, 10 randomly generated pairs of roadmaps are generated. dRRT* runs 5 random experiments for every roadmap pair, and the implicit A* searches these 10 tensor combinations. PRM* is run 10 times for every n. Average solution cost is reported over iterations. Data averaged over 10 roadmap pairs. dRRT* (solid line) converges to the optimal path through $\hat{\mathbb{G}}$ (dashed line).

Results are shown in Figure 6. dRRT* converges to the optimal path over $\hat{\mathbb{G}}$, similar to the one discovered by Implicit A*, while quickly finding an initial solution of high quality. Furthermore, the implicit tensor product

[5] Additional data are provided in Appendix B.

roadmap $\hat{\mathbb{G}}$ is of comparable quality to the explicitly constructed roadmap.

Table I presents running times. dRRT* and implicit A* construct 2 N-sized roadmaps (row 3), which are faster to construct than the PRM* roadmap in $\mathbb{C}$ (row 1). PRM* becomes very costly as N increases. For $N = 500$, the explicit roadmap contains $250,000$ vertices, taking 1.7GB of RAM to store, which was the upper limit for the machine used. When the roadmap can be constructed, it is quicker to query (row 2). dRRT* quickly returns an initial solution (row 5), and converges within 5% of the optimum length (row 6) well before Implicit A* returns a solution as N increases (row 4). The next benchmark further emphasizes this point.

TABLE I

CONSTRUCTION AND QUERY TIMES (SECS) FOR 2 DISK ROBOTS.

Number of nodes: N =	50	100	200
N^2-PRM* construction	3.427	13.293	69.551
N^2-PRM* query	0.002	0.004	0.023
2 N-size PRM* construction	0.1351	0.274	0.558
Implicit A* search over $\hat{\mathbb{G}}$	0.684	2.497	10.184
dRRT* over $\hat{\mathbb{G}}$ (initial)	0.343	0.257	0.358
dRRT* over $\hat{\mathbb{G}}$ (converged)	3.497	4.418	5.429

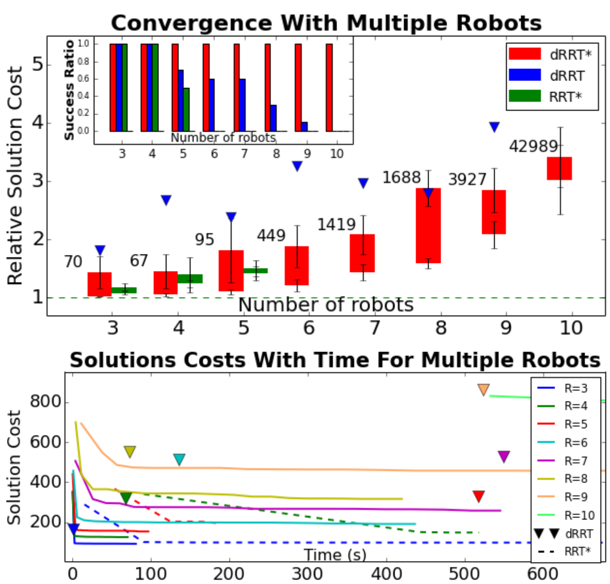

Fig. 7. Data averaged over 10 runs for 3 to 10 robots. (*Top*): Relative solution cost and success ratio of dRRT*, dRRT and RRT* for increasing R. dRRT*: average iteration and variance for initial solution (top of box), and solution cost and variance after $100,000$ iterations (bottom). Similar results for RRT*. Single data point for dRRT (no quality improvement after first solution). (*Bottom*): Solution costs over time.

Many Disk Robots among 2D Polygons: In the same environment as above, the number of robots R is increased to evaluate scalability. Each robot starts on the perimeter of the environment and is tasked with reaching the opposite side. An $N = 50$ roadmap is constructed for every robot. It quickly becomes intractable to construct a PRM* roadmap in the composite space of many robots.

Figure 7 shows the inability of alternatives to compete with dRRT* in scalability. Solution costs are normalized by an optimistic estimate of the path cost for each case, which is the sum of the optimal solutions for each robot, disregarding robot-robot interactions. Implicit A* fails to return solutions even for 3 robots. Directly executing RRT* in the composite space fails to do so for $R \geq 6$. The original dRRT method (without the informed search component) starts suffering in success ratio for $R \geq 5$ and returns worse solutions than dRRT*. The average solution times for dRRT may decrease as R increases but this is due to the decreasing success ratio, i.e., dRRT begins to only succeed at easy problems.

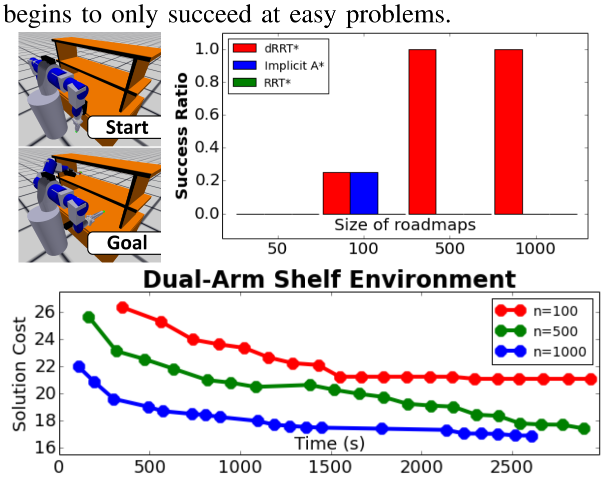

Fig. 8. (*Top*): dRRT* is run for a dual-arm manipulator to go from its home position (above) to a reaching configuration (below). 5 random experiments are run for 4 random roadmap pairs for every n. dRRT* achieves perfect success ratio as n increases. (*Bottom*): dRRT* solution quality over time. Here, larger roadmaps provide benefits in terms of running time and solution quality.

Dual-arm manipulator: This test shows the benefits of dRRT* when planning for two 7-dimensional arms. Figure 8 shows that RRT* fails to return solutions within $100K$ iterations. Using small roadmaps is also insufficient for this problem. Both dRRT* and Implicit A* require larger roadmaps to begin succeeding. But with $N \geq 500$, Implicit A* always fails, while dRRT* maintains a 100% success ratio. As expected, roadmaps of increasing size result in higher quality path. The informed nature of dRRT* also allows to find initial solutions fast, which together with the branch-and-bound primitive allows for good convergence.

VI. DISCUSSION

This work proves asymptotic optimality of sampling-based multi-robot planning over implicit structures using the dRRT* approach, which performs efficient, informed search. It constructs asymptotically optimal roadmaps for each robot, but searches over an implicitly defined tensor product, and avoids the need to construct large, dense roadmaps in the composite space of many robots.

Future work includes the consideration of dynamics given recent results [16], [15]. Furthermore, the analysis should be valid for a wide array of cost functions, but this must be shown. This approach can be leveraged toward efficiently solving simultaneous task and motion planning for many robot manipulators [2].

REFERENCES

[1] A. Dobson and K. E. Bekris. Sparse Roadmap Spanners for Asymptotically Near-Optimal Motion Planning. *Intern. Journal of Robotics Research (IJRR)*, 33(1):18–47, 2014.

[2] A. Dobson and K. E. Bekris. Planning Representations and Algorithms for Prehensile Multi-Arm Manipulation. In *IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems (IROS)*, 2015.

[3] A. Dobson, G. Moustakides, and K. E. Bekris. Geometric Probability Results For Bounding Path Quality In Sampling-Based Roadmaps After Finite Computation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[4] M. Gharbi, J. Cortés, and T. Siméon. Roadmap Composition for Multi-Arm Systems Path Planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009.

[5] F. Gravot and R. Alami. A Method for Handling Multiple Roadmaps and Its Use for Complex Manipulation Planning. In *IEEE Intern. Conf. on Robotics and Automation (ICRA)*, 2003.

[6] C. Grinstead and J. Snell. *Introduction to Probability*. American Mathmatical Society, Providence, RI, 2012.

[7] L. Janson, A. Schmerling, A. Clark, and M. Pavone. Fast Marching Tree: a Fast marching Sampling-Based Method for Optimal Motion Planning in Many Dimensions. *International Journal of Robotics Research (IJRR)*, 34(7):883–921, 2015.

[8] S. Karaman and E. Frazzoli. Sampling-based Algorithms for Optimal Motion Planning. *International Journal of Robotics Research (IJRR)*, 30(7):846–894, June 2011.

[9] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars. Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

[10] S. M. LaValle and J. J. Kuffner. Randomized Kinodynamic Planning. *International Journal of Robotics Research (IJRR)*, 20:378–400, May 2001.

[11] S. Leroy, J. Laumond, and Siméon. Multiple Path Coordination for Mobile Robots: A Geometric Algorithm. In *Intern. Joint Conf. on Artificial Intelligence (IJCAI)*, 1999.

[12] S. R. Lindemann and S. M. LaValle. Incrementally reducing dispersion by increasing voronoi bias in rrts. In *IEEE Intern. Conf. on Robotics and Automation (ICRA)*, 2004.

[13] J. Marble and K. E. Bekris. Asymptotically Near-Optimal Planning with Probabilistic Roadmap Spanners. *IEEE Transactions on Robotics*, 29(2):432–444, 2013.

[14] J. Peng and S. Akella. Coordinating Multiple Robots with Kinodynamic Constraints Along Specified Paths. *The International Journal of Robotics Research*, 24(4):295–310, 2005.

[15] E. Schmerling, L. Janson, and M. Pavone. Optimal sampling-based motion planning under differential constraints: The drift case with linear affine dynamics. In *IEEE Conference on Decision and Control (CDC)*, 2015.

[16] E. Schmerling, L. Janson, and M. Pavone. Optimal sampling-based motion planning under differential constraints: The driftless case. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[17] J. T. Schwartz and M. Sharir. On the Piano Movers' Problem: III. Coordinating the Motion of Several Independent Bodies: The Special Case of Circular Bodies Moving Amidst Polygonal Barriers. *The International Journal of Robotics Research*, 2(3):46–75, 1983.

[18] K. Solovey, O. Salzman, and D. Halperin. Finding a needle in an exponential haystack: Discrete RRT for exploration of implicit roadmaps in multi-robot motion planning. *The International Journal of Robotics Research*, 35(5):501–513, 2016.

[19] K. Solovey, O. Salzman, and D. Halperin. New Perspective on Sampling-based Motion Planning via Random Geometric Graphs. In *Robotics Science and Systems (RSS)*, 2016.

[20] P. Svestka and M. Overmars. Coordinated Path Planning for Multiple Robots. *Robotics and Autonomous Systems*, 23:125–152, 1998.

[21] S. Tang and V. Kumar. A Complete Algorithm for Generating Safe Trajectories for Multi-Robot Teams. In *International Symposium on Robotics Research (ISRR)*, Sestri Levante, Italy, September 2015.

[22] J. Van Den Berg, S. Guy, M. Lin, and D. Manocha. *Reciprocal n-body collision avoidance*, volume 70 of *Springer Tracts in Advanced Robotics*, pages 3–19. Star edition, 2011.

[23] J. van den Berg, J. Snoeyink, M. Lin, and D. Manocha. Centralized Path Planning for Multiple Robots: Optimal Decoupling into Sequential Plans. In *Robotics: Science and Systems (RSS)*, 2009.

[24] G. Wagner and H. Choset. Subdimensional Expansion for Multirobot Path Planning. *Artificial Intelligence Journal*, 219:1024, 2015.

APPENDIX

A. Proof of Convergence

This appendix examines the result of Theorem 2, and formally proves the convergence of the dRRT* tree toward containing all optimal paths.

Lemma 2 (Optimal Tree Convergence of dRRT*): Consider an arbitrary optimal path π^* originating from v_0 and ending at v_t, then let $O_k^{(m)}$ be the event such that after m iterations of dRRT*, the search tree $\mathbb{T}$ contains the optimal path up to segment k. Then,

$$\liminf_{m\to\infty} \mathbb{P}\big(O_t^{(m)}\big) = 1.$$

Proof. This is shown using Markov chain results [6, Theorem 11.3]. Specifically, absorbing Markov chains can be leveraged to show that dRRT* will eventually contain the optimal path over $\hat{\mathbb{G}}$. An absorbing Markov chain has some subset of its states in which the transition matrix only allows self-transitions.

The proof follows by showing that the dRRT* method can be described as an absorbing Markov chain, where the target state of a query is represented as an absorbing state in a Markov chain. For completeness, the theorem is re-stated here.

Theorem 3 (Thm 11.3 in Grinstead & Snell): In an absorbing Markov chain, the probability that the process will be absorbed is 1 (i.e., $Q(m) \to 0$ as $n \to \infty$), where $Q(m)$ is the transition submatrix for all non-absorbing states.

The first part is that the dRRT* search is cast as an absorbing Markov chain, and second, that the transition probability from each state to the next is nonzero, i.e., each state eventually connects to the target.

For query (S, T), let the sequence $V = \{v_1, v_2, \ldots, v_t\}$ of length t represent the vertices of $\hat{\mathbb{G}}$ corresponding to the optimal path through the graph which connects these points, where v_t corresponds to the target vertex, and furthermore, let v_t be an absorbing state. Theorem 3 operates under the assumption that each vertex v_i is connected to an absorbing state (v_t in this case).

Then, let the transition probability for each state have two values, one for each state transitioning to itself, which corresponds to the dRRT* search expanding along some other arbitrary path. The other value is a transition probability from v_i to v_{i+1}. This corresponds to the method sampling within the volume $\text{Vol}(v_i)$.

Then, as the second step, it must be shown that this volume has a positive probability of being sampled in each iteration. It is sufficient then to argue that $\frac{\mu(\text{Vol}(s_i))}{\mu(\mathbb{C}^f)} > 0$. Fortunately, for any finite n, previous work has already shown that this is the case given general position assumptions [18, Lemma 2].

Given these results, the dRRT* is cast as an absorbing Markov chain, which satisfies the assumptions of 3, and therefore, the matrix $Q(m) \to 0$. This implies that the optimal path to the goal has been expanded in the tree, and therefore $\liminf_{m\to\infty} \mathbb{P}(O_t^{(m)}) = 1$. $\square$

B. More Experimental Data

This appendix presents additional experimental data.

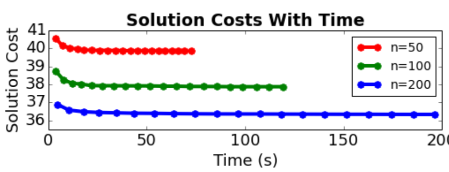

Fig. 9. 2-Robot convergence data over time.

1) 2-Robot Benchmark: For the 2 disk robots case, Figure 9 provides the solution cost found by dRRT*over computation time instead of over iterations.

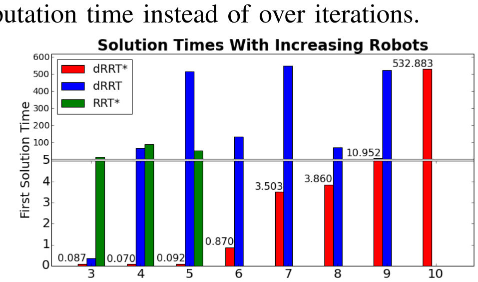

Fig. 10. R-robot solution times for varying R.

2) R Disk Robots: For the R disk robots, Figure 10 shows query resolution times for the various methods.

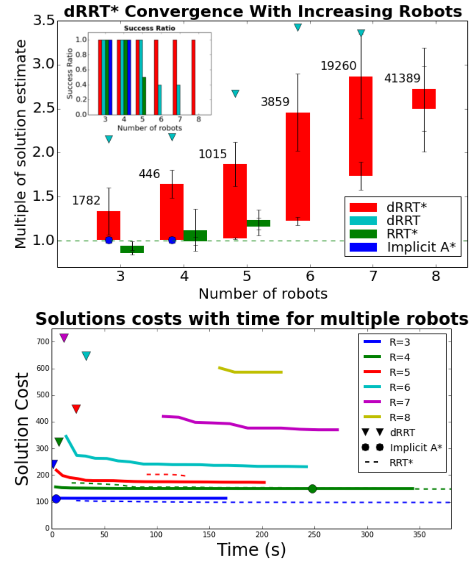

Fig. 11. (*Top*): Convergence rate and success ratio over the minimal 9-node roadmap (*Bottom*): Solution cost over time when using the minimal roadmap.

To emphasize the lack of scalability for alternate methods, additional experiments were run using a smaller, manually crafted 9-node roadmap, shown in Figure 12. Each robot roadmap samples its nodes within the shaded regions indicated in the figure.

Figure 11 indicates that even using a very well-crafted and small roadmap for a problem does not help the alternate methods to scale. While they perform better than with the random roadmaps, Implicit A* times out for $R \geq 5$, and RRT* times out for $R \geq 6$.

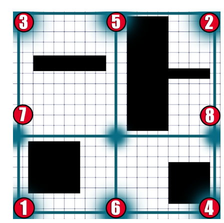

Fig. 12. Minimal graph for the R-robot case.

3) Manipulator Benchmark: For the dual-arm manipulator benchmark, Figure 13 presents solution quality over iterations instead of over time.

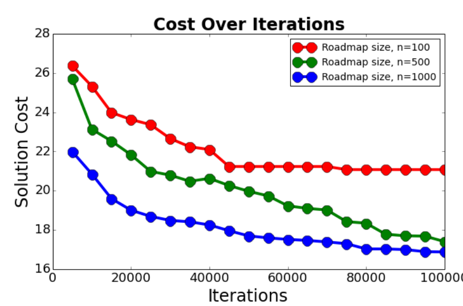

Fig. 13. Motoman benchmark solution quality over iterations.