Fast Marching Trees: a Fast Marching Sampling-Based Method for Optimal Motion Planning in Many Dimensions

Lucas Janson and Marco Pavone

Abstract In this paper we present a novel probabilistic sampling-based motion planning algorithm called the Fast Marching Tree algorithm (FMT*). The algorithm is specifically aimed at solving complex motion planning problems in high-dimensional configuration spaces. This algorithm is proven to be asymptotically optimal and is shown to converge to an optimal solution faster than its state-of-the-art counterparts, chiefly PRM* and RRT*. An additional advantage of FMT* is that it builds and maintains paths in a tree-like structure (especially useful for planning under differential constraints). The FMT* algorithm essentially performs a "lazy" dynamic programming recursion on a set of probabilistically-drawn samples to grow a tree of paths, which moves steadily outward in cost-to-come space. As such, this algorithm combines features of both single-query algorithms (chiefly RRT) and multiple-query algorithms (chiefly PRM), and is conceptually related to the Fast Marching Method for the solution of eikonal equations. As a departure from previous analysis approaches that are based on the notion of almost sure convergence, the FMT* algorithm is analyzed under the notion of convergence in probability: the extra mathematical flexibility of this approach allows for significant algorithmic advantages and provides convergence rate bounds – a first in the field of optimal sampling-based motion planning. Numerical experiments over a range of dimensions and obstacle configurations confirm our theoretical and heuristic arguments by showing that FMT*, for a given execution time, returns substantially better solutions than either PRM* or RRT*, especially in high-dimensional configuration spaces.

Lucas Janson
Department of Statistics, Stanford University, Stanford, CA 94305, e-mail: ljanson@stanford.edu

Marco Pavone
Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, e-mail: pavone@stanford.edu

A preliminary version of this work has been *orally* presented at the workshop on "Robotic Exploration, Monitoring, and Information Collection: Nonparametric Modeling, Information-based Control, and Planning under Uncertainty" at the Robotics: Science and Systems 2013 conference. This work has neither appeared elsewhere for publication, nor is under review for another refereed publication.

1 Introduction

Probabilistic sampling-based algorithms represent a particularly successful approach to robotic motion planning problems in high-dimensional configuration spaces, which naturally arise, e.g., when controlling the motion of high degree-of-freedom robots or planning under uncertainty [19, 13]. Accordingly, the design of rapidly converging sampling-based algorithms with sound performance guarantees has emerged as a central topic in robotic motion planning, and represents the main thrust of this paper.

Specifically, the key idea behind probabilistic sampling-based algorithms is to avoid the explicit construction of the configuration space (which is prohibitive in the high-dimensional case), and instead conduct a search that probabilistically probes the configuration space with a sampling scheme. This probing is enabled by a collision detection module, which the motion planning algorithm considers as a "black box" [13]. Probabilistic sampling-based algorithms can be divided between multiple-query and single-query. Multiple-query algorithms construct a topological graph called a roadmap, which allows a user to efficiently solve multiple initial-state/goal-state queries. This family of algorithms includes the probabilistic roadmap algorithm (PRM) [10] and its variants, e.g., Lazy-PRM [3], dynamic PRM [7], and PRM* [9]. On the contrary, in single-query algorithms, a single initial-state/goal-state pair is given, and the algorithm must search until it finds a solution (or it may report early failure). This family of algorithms includes the rapidly exploring random trees algorithm (RRT) [14], the rapidly exploring dense trees algorithm (RDT) [13], and their variants, e.g., RRT* [9]. Other notable sampling-based planners include expansive space trees (EST) [5, 16], sampling-based roadmap of trees (SRT) [17], rapidly-exploring roadmap (RRM) [1], and the "cross-entropy" planner in [11]. Analysis in terms of convergence to feasible or even optimal solutions for multiple-query and single-query algorithms is provided in [5, 12, 2, 6, 9]. A central result is that these algorithms provide *probabilistic completeness* guarantees in the sense that the probability that the planner fails to return a solution, if one exists, decays to zero as the number of samples approaches infinity [2]. Recently, it has been proven that both RRT* and PRM* are asymptotically optimal, i.e. the cost of the returned solution converges almost surely to the optimum [9]. Building upon the results in [9], the work in [15] presents an algorithm with provable "sub-optimality" guarantees, which trades "optimality" with faster computation.

Statement of Contributions: The objective of this paper is to propose and analyze a novel probabilistic motion planning algorithm that is asymptotically optimal and improves upon state-of-the-art asymptotically-optimal algorithms (namely RRT* and PRM*) in terms of the convergence rate to the optimal solution (convergence rate is interpreted with respect to execution time). The algorithm, named Fast Marching Tree algorithm (FMT*), is designed to be particularly efficient in high-dimensional environments cluttered with obstacles. FMT* essentially combines some of the features of multiple-query algorithms with those of single-query algorithms, by performing a "lazy" dynamic programming recursion on a set of probabilistically-drawn samples in the configuration space. As such, this algorithm combines features of PRM and SRT (similar to RRM), and grows a tree of trajectories like RRT. Additionally, FMT* is conceptually similar to the Fast Marching Method, one of the main methods for the solution of stationary eikonal equations [18]. As in the Fast Marching Method, the main idea is to exploit a heapsort technique to systematically locate the proper sample point to update and to incrementally build the solution in an "outward" direction, so that one needs never backtrack over previously evaluated sample points. Such a Dijkstra-like one-pass property is what makes both the Fast Marching Method and FMT* particularly efficient.

The end product of the FMT* algorithm is a tree, which, together with the connection to the Fast Marching Method, gives the algorithm its name. An advantage of FMT* with respect to PRM* (in addition to a faster convergence rate) is the fact that FMT* builds and maintains paths in a tree-like structure, which is especially useful when planning under differential or integral constraints. Our simulations across 2, 5, 7, and 10 dimensions, both without obstacles and with 50% obstacle coverage, show that FMT* substantially outperforms PRM* and RRT*.

It is important to note that in this paper we use a notion of asymptotic optimality (AO) different from the one used in [9]. In [9], AO is defined through the notion of convergence almost everywhere (a.e.). Explicitly, in [9], an algorithm is considered AO if the cost of the solution it returns converges a.e. to the optimal cost as the number of samples n approaches infinity. This definition is completely justified when the algorithm is sequential in n, such as RRT* [9], in the sense that it requires that with probability 1, the sequence of solutions converges to an optimal one, with the solution at $n+1$ heavily related to that at n. However, for non-sequential algorithms such as PRM* and FMT*, there is no connection between the solutions at n and $n+1$. Since these algorithms process all the nodes at once, the solution at $n+1$ is based on $n+1$ new nodes, sampled independently of those used in the solution at n. This motivates the definition of AO used in this paper, which is that the cost of the solution returned by an algorithm must converge *in probability* to the optimal cost. Although mathematically convergence in probability is a weaker notion than convergence a.e. (the latter implies the former), in practice there is no distinction when an algorithm is only run on a predetermined, fixed number of nodes. In this case, all that matters is that the probability that the cost of the solution returned by the algorithm is less than an ε fraction greater than the optimal cost goes to 1 as $n \to \infty$, for any $\varepsilon > 0$, which is exactly the statement of convergence in probability. Since this is a mathematically weaker, but practically identical condition, we sought to capitalize on the extra mathematical flexibility, and indeed find that our proof of AO for FMT* allows for an improved implementation of PRM* in [9], providing a speed-up linear in the number of dimensions. Our proof of AO also gives a *convergence rate bound* both for FMT* and PRM* – a first in the field of optimal sampling-based motion planning. Hence, an additional important contribution of this paper is the analysis of AO under the notion of convergence in probability, which is of independent interest and could enable the design and analysis of other AO sampling-based algorithms.

Organization: This paper is structured as follows. In Section 2 we formally define the optimal path planning problem. In Section 3 we present FMT*, and we prove some basic properties, e.g., termination. In Section 4 we prove the asymptotic optimality of FMT*. In Section 5 we first conceptually discuss the advantages of FMT* and then we present results from numerical experiments supporting our statements. Finally, in Section 6, we draw some conclusions and we discuss directions for future work.

Notation: Consider the Euclidean space in d dimensions, i.e., $\mathbb{R}^d$. Given a point $x \in \mathbb{R}$, a ball of radius $r > 0$ centered at $\bar{x} \in \mathbb{R}^d$ is defined as $B(\bar{x};\, r) := \{x \in \mathbb{R}^d \mid \|x - \bar{x}\| < r\}$. Given a subset $\mathscr{X}$ of $\mathbb{R}^d$, its boundary is denoted by $\partial \mathscr{X}$. Given two points x and z in $\mathbb{R}^d$, the line connecting them is denoted by $\overline{xz}$. Let ζ_d denote the volume of the unit ball in the d-dimensional Euclidean space. The cardinality of a set S is written card$S$. Given a set $\mathscr{X} \subseteq \mathbb{R}^d$, $\mu(\mathscr{X})$ denotes its d-dimensional Lebesgue measure. In this paper, we will interchangeably refer to points in $\mathscr{X}$ as nodes, samples, or vertices.

2 Problem setup

The problem formulation follows closely the problem formulation in [9]. Let $\mathscr{X} = [0,\, 1]^d$ be the configuration space, where $d \in \mathbb{N}$, $d \geq 2$. Let $\mathscr{X}_{\text{obs}}$ be the obstacle region, such that $\mathscr{X} \setminus \mathscr{X}_{\text{obs}}$ is an open set (we consider $\partial \mathscr{X} \subset \mathscr{X}_{\text{obs}}$), and denote the obstacle-free space as $\mathscr{X}_{\text{free}} = \text{cl}(\mathscr{X} \setminus \mathscr{X}_{\text{obs}})$, where cl$(\cdot)$ denotes the closure of a set. The initial condition x_{init} is an element of $\mathscr{X}_{\text{free}}$, and the goal region $\mathscr{X}_{\text{goal}}$ is an open subset of $\mathscr{X}_{\text{free}}$. A path planning problem is denoted by a triplet $(\mathscr{X}_{\text{free}}, x_{\text{init}}, \mathscr{X}_{\text{goal}})$. A function of *bounded variation* $\sigma : [0,1] \to \mathbb{R}^d$ is called a *path* if it is continuous. A path is said to be *collision-free* if $\sigma(\tau) \in \mathscr{X}_{\text{free}}$ for all $\tau \in [0,\, 1]$. A path is said to be a *feasible path* for the planning problem $(\mathscr{X}_{\text{free}}, x_{\text{init}}, \mathscr{X}_{\text{goal}})$ if it is collision-free, $\sigma(0) = x_{\text{init}}$, and $\sigma(1) \in cl(\mathscr{X}_{\text{goal}})$.

A goal region $\mathscr{X}_{\text{goal}}$ is said to be *regular* if there exists $\xi > 0$ such that $\forall y \in \partial \mathscr{X}_{\text{goal}}$, there exists $z \in \mathscr{X}_{\text{goal}}$ with $B(z;\xi) \subseteq \mathscr{X}_{\text{goal}}$ and $y \in \partial B(z;\xi)$. In other words, a regular goal region is a "well-behaved" set where the boundary has bounded curvature. We will say $\mathscr{X}_{\text{goal}}$ is ξ-regular if $\mathscr{X}_{\text{goal}}$ is regular for the parameter ξ. Let Σ be the set of all paths. A cost function for the planning problem $(\mathscr{X}_{\text{free}}, x_{\text{init}}, \mathscr{X}_{\text{goal}})$ is a function $c : \Sigma \to \mathbb{R}_{\geq 0}$ from the set of paths to the nonnegative real numbers; in this paper we will consider as cost functions $c(\sigma)$ the *arc length* of σ with respect to the Euclidean metric in $\mathscr{X}$ (recall that σ is, by definition, rectifiable). Extension to more general cost functions (possibly not satisfying the triangle inequality) are possible and are deferred to future work. The optimal path planning problem is then defined as follows:

Optimal path planning problem: Given a path planning problem $(\mathscr{X}_{\text{free}}, x_{\text{init}}, \mathscr{X}_{\text{goal}})$ with a regular goal region and an arc length function $c : \Sigma \to \mathbb{R}_{\geq 0}$, find a feasible path σ^* such that $c(\sigma^*) = \min\{c(\sigma) : \sigma \text{ is feasible}\}$. If no such path exists, report failure.

Finally, we introduce some definitions concerning the *clearance* of a path, i.e., its "distance" from $\mathscr{X}_{\text{obs}}$ [9]. For a given $\delta > 0$, the δ-interior of $\mathscr{X}_{\text{free}}$ is defined as the set of all states that are at least a distance δ away from any point in $\mathscr{X}_{\text{obs}}$. A collision-free path σ is said to have strong δ-clearance if it lies entirely inside the δ-interior of $\mathscr{X}_{\text{free}}$. A collision-free path σ is said to have weak δ-clearance if there exists a path σ' that has strong δ-clearance and there exists a homotopy ψ, with $\psi(0) = \sigma$, $\psi(1) = \sigma'$, and for all $\alpha \in (0,1]$ there exists $\delta_\alpha > 0$ such that $\psi(\alpha)$ has strong δ_α-clearance.

3 The Fast Marching Tree algorithm (FMT*)

In this section we present the Fast Marching Tree algorithm, FMT*, described in pseudocode in Algorithm 1. We first introduce the algorithm, then discuss its termination properties, and finally discuss implementation details and computational complexity. The proof of its (asymptotic) optimality will be presented in Section 4.

3.1 The Algorithm

Let $\mathtt{SampleFree}(k)$ be a function that returns a set of $k \in \mathbb{N}$ points sampled independently and identically from the uniform distribution on $\mathcal{X}_{\text{free}}$. We use the uniform distribution in this paper for simplicity, but any distribution supported on $\mathcal{X}_{\text{free}}$ would yield identical theoretical results, in particular AO. Given a vertex $x \in \mathcal{X}$ and a set of vertices V, let $\mathtt{Save}(V,x)$ be a function that stores in memory a set of vertices V associated with a vertex x. Given a set of vertices V and a positive number r, let $\mathtt{Near}(V,x,r)$ be a function that returns the set of vertices $\{v \in V : \|v-x\| < r\}$. Given a graph $G = (V,E)$, where V is the vertex set and E is the edge set, and a vertex $x \in V$, let $\mathtt{Cost}(x,G)$ be a function that returns the cost of the shortest path in the graph G between the vertices x_{init} and x. With a slight abuse of notation, we define the function $\mathtt{Cost}(\overline{vz})$ as the function that returns the cost of the line $\overline{vz}$ (note that $\mathtt{Cost}(\overline{vz})$ is well defined regardless of $\overline{vz}$ being collision free). Given two vertices x and z in $\mathcal{X}_{\text{free}}$, let $\mathtt{CollisionFree}(x,z)$ denote the boolean function which is true if and only if $\overline{xz}$ does not intersect an obstacle. Given two sets of points $\mathcal{S}_1$ and $\mathcal{S}_2$ in $\mathcal{X}_{\text{free}}$, let $\mathtt{Intersect}(\mathcal{S}_1,\mathcal{S}_2)$ be the function that returns the set of points that belong to both $\mathcal{S}_1$ and $\mathcal{S}_2$. Given a tree $T = (V,E)$, where V is the vertex set and E is the edge set, and a vertex $x \in T$, let $\mathtt{Path}(x,T)$ be the function that returns the unique path in the tree T from x_{init} to vertex x.

Algorithm 1 Fast Marching Tree Algorithm (FMT*)

```
$V \leftarrow \{x_{\text{init}}\} \cup \mathtt{SampleFree}(n)$; $E \leftarrow \emptyset$
$W \leftarrow V \backslash \{x_{\text{init}}\}$; $H \leftarrow \{x_{\text{init}}\}$
$z \leftarrow x_{\text{init}}$
$N_z \leftarrow \mathtt{Near}(V \backslash \{z\}, z, r_n)$
$\mathtt{Save}(N_z, z)$
while $z \notin \mathcal{X}_{\text{goal}}$ do
    $H_{\text{new}} \leftarrow \emptyset$
    $X_{\text{near}} = \mathtt{Intersect}(N_z, W)$
    for $x \in X_{\text{near}}$ do
        $N_x \leftarrow \mathtt{Near}(V \backslash \{x\}, x, r_n)$
        $\mathtt{Save}(N_x, x)$
        $Y_{\text{near}} \leftarrow \mathtt{Intersect}(N_x, H)$
        $y_{\min} \leftarrow \arg\min_{y \in Y_{\text{near}}} \{\mathtt{Cost}(y, T = (V,E)) + \mathtt{Cost}(\overline{yx})\}$
        if $\mathtt{CollisionFree}(y_{\min}, x)$ then
            $E \leftarrow E \cup \{(y_{\min}, x)\}$    // $\overline{y_{\min}x}$ is collision-free
            $H_{\text{new}} \leftarrow H_{\text{new}} \cup \{x\}$
            $W \leftarrow W \backslash \{x\}$
        end if
    end for
    $H \leftarrow (H \cup H_{\text{new}}) \backslash \{z\}$
    if $H = \emptyset$ then
        return Failure
    end if
    $z \leftarrow \arg\min_{y \in H} \{\mathtt{Cost}(y, T = (V,E))\}$
end while
return $\mathtt{Path}(z, T = (V,E))$
```

Essentially, the FMT* algorithm executes a forward dynamic programing recursion over the vertices sampled with the $\mathtt{SampleFree}(n)$ function. FMT* maintains dual sets H and W, where H keeps track of nodes which have already been added to the tree, although it drops nodes that are not near enough to the edge of the expanding tree to actually have any new connections made. The set W maintains the nodes which have not been added to the tree. The algorithm generates a tree by moving steadily outward in cost-to-come space (see Figure 1). To give some intuition, when there are no obstacles and the cost is Euclidean distance, FMT* reports the exact same solution (or failure) as PRM*. This is because, without obstacles, FMT* is indeed using dynamic programming to build the minimum-cost spanning tree, with x_{init} as the root, of the set of nodes in the PRM* graph which are connected to x_{init}. The only difference is that by not starting with the entire PRM* graph itself, FMT* is able to find the solution much faster (see Section 5). In the presence of obstacles, FMT* and PRM* no longer return the same solution in general, and this is due to how FMT* deals with obstructing obstacles. In particular, when FMT* searches for a connection for a node v, it will leave v unconnected (and let it remain in W to be checked again in later iterations) if the node $x \in H$ whose connection (if obstacle-free) would produce the smallest cost-to-come for v is blocked from connecting to v by an obstacle. Assuming x_0 is the optimal parent of v with respect to the PRM* graph, v will never be connected to x_0 in FMT* only if when x_0 is the minimum-cost node in H, there is another node $x_1 \in H$ such that (a) x_1 has (necessarily, by the structure of H) greater cost-to-come than x_0, (b) x_1 is within a radius r_n of v, (c) x_1 is blocked from connecting to v by an obstacle, and (d) obstacle-free connection of v to x_1 would have lower cost-to-come than connection to x_0. If any of these conditions fail, then on some iteration (possibly not the first), v will be connected optimally with respect to PRM*. Note that the combination of conditions (a), (b), (c), and (d) ought to make such suboptimal connections quite rare.

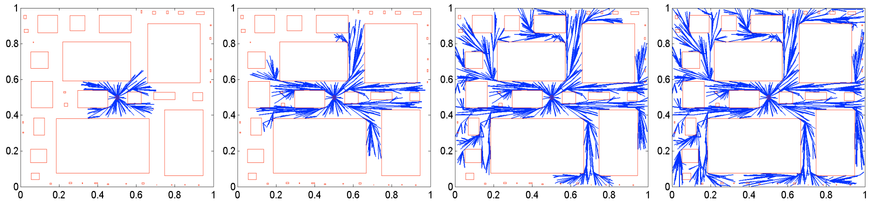

Fig. 1 The FMT* algorithm generates a tree by moving steadily outward in cost-to-come space. This figure portrays the growth of the tree in a 2D environment with 1,000 nodes (not shown).

3.2 Termination

One might wonder if, in the first place, the FMT* algorithm always terminates, i.e., it does not cycle indefinitely through the sets H and W. The following theorem shows that indeed FMT* always terminates.

Theorem 1 (Termination). *Consider a path planning problem* $(\mathscr{X}_{free}, x_{init}, \mathscr{X}_{goal})$ *and any* $n \in \mathbb{N}$. *The FMT* algorithm always terminates in at most n iterations of the while loop.*

Proof. To prove that FMT* always terminates, note two key facts: (i) FMT* terminates and reports failure if H is ever empty, and (ii) the minimum-cost node in H is removed from H at each while loop iteration. Therefore, to prove the theorem it suffices to prove the invariant that any node that has ever been added to H can never be added again (this, in fact, would imply that the while loop goes through at most n iterations). To establish the invariant, observe that at a given iteration only nodes in X_{near} are added to H. However, $X_{\text{near}} \subseteq W$, so only nodes in W can be added to H, and each time a node is added, it is removed from W. Finally, since W never has nodes added to it, a node can only be added to H once. This proves the invariant, and, in turn, the claim. □

3.3 Implementation Details

The set H should be implemented as a binary min heap, ordered by cost-to-come, with a parallel set of nodes H' which exactly tracks the nodes in H (but in no particular order) for use when the intersection operation in line 12 of the algorithm is executed. For many of the nodes $x \in V$, FMT* saves the associated set N_x of r_n-neighbors for that node. Instead of just saving a reference for each node $y \in N_x$, N_x can also have memory allocated for the real value $\texttt{Cost}(\overline{yx})$ and the boolean value $\texttt{CollisionFree}(y,x)$. Saving both of these values whenever they are first computed guarantees that FMT* will never compute them more than once for a given pair of nodes. Finally, the first `Cost` function (the one that takes in a vertex and graph) should not be a computation; the costs should just be saved for every node that is added to the tree. Since the cost is only queried for nodes that are already in the tree, and that cost never changes, it would suffice to just add a step which saves the cost-to-come of x.

Previous work has often characterized the computational complexity of an algorithm by how long it takes to run the algorithm on n nodes. However the computational complexity that ultimately matters is how long it takes for an algorithm to return a solution of a certain quality. We focus on this concept of computational complexity and rely on simulations as a guide, although we note here that FMT* is $O(n\log(n))$, with the proof omitted due to space limitations. In order to compute this more relevant computational complexity, we would need a characterization of how fast the solution improves with the number of nodes – more on this in Remark 2 at the end of the next section.

4 Asymptotic Optimality of FMT*

The following theorem presents the main result of this paper.

Theorem 2 (Asymptotic optimality of FMT*). *Let $(\mathscr{X}_{free}, x_{init}, \mathscr{X}_{goal})$ be a path planning problem in d dimensions, with $\mathscr{X}_{goal}$ ξ-regular, such that there exists an optimal path σ^* with weak δ-clearance for some $\delta > 0$. Let c^* denote the arc length of σ^*, and let c_n denote the cost of the path returned by FMT* (or ∞ if FMT* returns failure) with n vertices using the following radius,*

$$r_n = (1+\eta)\cdot 2\left(\frac{1}{d}\right)^{\frac{1}{d}}\left(\frac{\mu(\mathscr{X}_{free})}{\zeta_d}\right)^{\frac{1}{d}}\left(\frac{\log(n)}{n}\right)^{\frac{1}{d}}, \tag{1}$$

for some $\eta > 0$. Then $\lim_{n\to\infty} \mathbb{P}(c_n > (1+\varepsilon)c^*) = 0$ for all $\varepsilon > 0$.

Proof. Note that $c^* = 0$ implies $x_{\text{init}} \in cl(\mathscr{X}_{\text{goal}})$, and the result is trivial, therefore assume $c^* > 0$. Fix $\theta \in (0, 1/4)$ and define the sequence of paths σ_n such that $\lim_{n\to\infty} c(\sigma_n) = c^*$, $\sigma_n(1) \in \partial\mathscr{X}_{\text{goal}}$, $\sigma_n(\tau) \notin \mathscr{X}_{\text{goal}}$ for all $\tau \in (0,1)$, $\sigma_n(0) = x_{\text{init}}$, and σ_n has strong δ_n-clearance, where $\delta_n = \min\{\delta, \frac{3+\theta}{2+\theta} r_n\}$. A proof that such a sequence of paths exists can be found in [9] as Lemma 50. It requires a slew of extra notation and metric space results, and so the proof is omitted here.

Let σ_n' be the concatenation of σ_n with the line that extends from $\sigma_n(1)$ in the direction perpendicular to the tangent hyperplane of $\partial\mathscr{X}_{\text{goal}}$ at $\sigma_n(1)$ of length $\min\{\xi, \frac{r_n}{2(2+\theta)}\}$. Note that this tangent hyperplane is well-defined, since the regularity assumption for $\mathscr{X}_{\text{goal}}$ ensures that its boundary is differentiable. Note that, trivially, $\lim_{n\to\infty} c(\sigma_n') = \lim_{n\to\infty} c(\sigma_n) = c^*$.

Fix $\varepsilon \in (0,1)$, suppose $\alpha, \beta \in (0, \theta\varepsilon/8)$, and pick $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$ the following conditions hold: (1) $\frac{r_n}{2(2+\theta)} < \xi$, (2) $\frac{3+\theta}{2+\theta} r_n < \delta$, (3) $c(\sigma_n') < (1+\frac{\varepsilon}{4})c^*$, and (4) $\frac{r_n}{2+\theta} < \frac{\varepsilon}{8}c^*$.

For the remainder of this proof, assume $n \geq n_0$. From conditions (1) and (2), σ_n' has strong $\frac{3+\theta}{2+\theta} r_n$-clearance. Letting $\kappa(\alpha,\beta,\theta) := 1 + (2\alpha+2\beta)/\theta$, conditions (3) and (4) imply,

$$\begin{aligned}\kappa(\alpha,\beta,\theta)\, c(\sigma_n') + \frac{r_n}{2+\theta} &\leq \kappa(\alpha,\beta,\theta)\left(1+\frac{\varepsilon}{4}\right)c^* + \frac{\varepsilon}{8}c^* \\ &\leq \left(\left(1+\frac{\varepsilon}{2}\right)\left(1+\frac{\varepsilon}{4}\right)+\frac{\varepsilon}{8}\right)c^* \leq (1+\varepsilon)c^*.\end{aligned}$$

Therefore,

$$\mathbb{P}\left(c_n > (1+\varepsilon)c^*\right) = 1 - \mathbb{P}\left(c_n \leq (1+\varepsilon)c^*\right) \leq 1 - \mathbb{P}\left(c_n \leq \kappa(\alpha,\beta,\theta)\, c(\sigma_n') + \tfrac{r_n}{2+\theta}\right). \tag{2}$$

Define the sequence of balls $B_{n,1}, \ldots, B_{n,M_n} \subseteq \mathscr{X}_{\text{free}}$ parameterized by θ as follows. For $m = 1$ we define $B_{n,1} := B\left(\sigma_n(\tau_{n,1}); \frac{r_n}{2+\theta}\right)$, with $\tau_{n,1} = 0$. For $m = 2,3,\ldots$, let $\Gamma_m = \left\{\tau \in (\tau_{n,m-1}, 1) : \|\sigma_n(\tau) - \sigma_n(\tau_{n,m-1})\| = \frac{\theta r_n}{2+\theta}\right\}$; if $\Gamma_m \neq \emptyset$ we define $B_{n,m} := B\left(\sigma_n(\tau_{n,m}); \frac{r_n}{2+\theta}\right)$, with $\tau_{n,m} = \min_\tau \Gamma_m$. Let M_n be the first m such that $\Gamma_m = \emptyset$, then, $B_{n,M_n} := B\left(\sigma_n'(1); \frac{r_n}{2(2+\theta)}\right)$, and we stop the process, i.e., B_{n,M_n} is the last ball placed along the path σ_n (note that the center of the last ball is $\sigma_n'(1)$). Considering the construction of σ_n' and condition (1) above, we conclude that $B_{n,M_n} \subseteq \mathscr{X}_{\text{goal}}$.

Recall that V is the set of nodes available to algorithm FMT* (see line 1 in Algorithm 1). We define the event $A_{n,\theta} := \bigcap_{m=1}^{M_n}\{B_{n,m} \cap V \neq \emptyset\}$; $A_{n,\theta}$ is the event that each ball contains at least one (not necessarily unique) node in V (for clarity, we made the event's dependence on θ, due to the dependence on θ of the balls, explicit). Further, for all $m \in \{1,\ldots,M_n - 1\}$, let $B_{n,m}^{\beta}$ be the ball with the same center as $B_{n,m}$ and radius $\frac{\beta r_n}{2+\theta}$, and let K_n^{β} be the number of smaller balls $B_{n,m}^{\beta}$ not containing any of the nodes in V, i.e., $K_n^{\beta} := \text{card}\{m \in \{1,\ldots,M_n-1\} : B_{n,m}^{\beta} \cap V = \emptyset\}$.

We now present three important lemmas, the first of which is proved in the Appendix, and the other two of which are proved in the extended version of this paper available on arXiv [8].

Lemma 1. *Under the assumptions of Theorem 2 and assuming $n \geq n_0$, the following inequality holds:*

$$\mathbb{P}\left(c_n \leq \kappa(\alpha,\beta,\theta)\,c(\sigma_n') + \tfrac{r_n}{2+\theta}\right) \geq 1 - \mathbb{P}(K_n^\beta \geq \alpha(M_n-1)) - \mathbb{P}(A_{n,\theta}^c).$$

Lemma 2. *Under the assumptions of Theorem 2, for all $\alpha \in (0,1)$ and $\beta \in (0,\theta/2)$, it holds that:* $\lim_{n\to\infty}\mathbb{P}(K_n^\beta \geq \alpha(M_n-1)) = 0$.

Lemma 3. *Under the assumptions of Theorem 2, assume that $r_n = \gamma(\log n/n)^{1/d}$, where $\gamma = (1+\eta)\cdot 2\,(1/d)^{\frac{1}{d}}\,(\mu(\mathscr{X}_{free})/\zeta_d)^{1/d}$ and $\eta > 0$. Then for all $\theta < 2\eta$,* $\lim_{n\to\infty}\mathbb{P}(A_{n,\theta}^c) = 0$.

Essentially, Lemma 1 provides a lower bound for the cost of the solution delivered by FMT* in terms of the probabilities that the “big” balls and “small” balls do not contain vertices in V. Lemma 2 states that the probability that the fraction of small balls not containing vertices in V is larger than an α fraction of the total number of balls is asymptotically zero. Finally, Lemma 3 states that the probability that at least one “big” ball does not contain any of the vertices in V is asymptotically zero.

The asymptotic optimality claim of the theorem then follows easily. Let $\varepsilon \in (0,1)$ and pick $\theta \in (0,\min\{2\eta,1/4\})$ and $\alpha,\beta \in (0,\theta\varepsilon/8) \subset (0,\theta/2)$. From equation (2) and Lemma 1, one can write

$$\lim_{n\to\infty}\mathbb{P}(c_n > (1+\varepsilon)c^*) \leq \lim_{n\to\infty}\mathbb{P}\left(K_n^\beta \geq \alpha(M_n-1)\right) + \lim_{n\to\infty}\mathbb{P}\left(A_{n,\theta}^c\right).$$

The right hand-side of this equation equals zero by Lemmas 2 and 3, and the claim is proven. The case with general ε follows by monotonicity in ε of the above probability. □

Remark 1. Since the solution returned by FMT* is never better than the one returned by PRM*, the exact same result holds for PRM*. Note that this proof uses a γ which is a factor of $(d+1)^{1/d}$ smaller (and thus a r_n which is $(d+1)^{1/d}$ smaller) than that in [9]. Since the number of cost computations and checks to collision-free scale approximately as r_n^d, this factor should reduce run time substantially for a given number of nodes, especially in high dimensions. This is due to the difference in definitions of AO mentioned earlier which, again, makes no practical difference for PRM* or FMT*. Furthermore, the result holds for PRM* under the more general cost definition used in [9] under the new definition of AO by replacing Lemma 52 in [9] with Lemma 3 from this paper.

Remark 2. We now note a very intriguing and promising side-effect of the new definition of AO. In proving that $\mathbb{P}(c_n > (1+\varepsilon)c^*) \to 0$ for all $\varepsilon > 0$, we actually had to characterize this probability. Although the details of the functional form are scattered throughout the proof of Theorem 2 (much of which is presented in the extended version of this paper in [8]), a convergence rate bound for FMT*, and thus also PRM*, can be obtained as a function of n. As far as the authors are aware, this would be the first such convergence rate result for an optimal sampling-based motion-planning algorithm, and would be an important step towards understanding the behavior of this class of algorithms. An experimental verification of the tightness of this bound is left for future work.

5 Numerical Experiments and Discussion

In this section we discuss the advantages of FMT* over previous sampling-based motion planning algorithms. To the best of our knowledge, the only other asymptotically optimal algorithms are PRM*, RRG, and RRT* [9], so it is with these state-of-the-art methods that we draw comparison. We first present a conceptual comparison between FMT* and such algorithms, and then we present results from numerical experiments.

5.1 Conceptual Comparison with Existing AO Algorithms

As compared to RRT*, we expect FMT* to show some improvement in solution quality per number of nodes placed. This is because for a given set of nodes, FMT* creates connections nearly optimally (exactly optimally when there are no obstacles) within the radius constraints, while RRT*, even with its rewiring step, is ultimately a greedy algorithm. It is however hard to conceptually compare how long the algorithms might take to run on a given set of nodes, given how differently they generate paths.

One advantage of FMT* over the graph-based methods such as PRM* and RRG is that FMT* builds paths in a tree-like structure at all times. This is important when differential or integral constraints are added to the paths. If, for example, a smoothness constraint is placed on paths, then since FMT* builds its paths outwards, each new branch solves the differentially constrained system with initial conditions starting from its parent node/state, whose optimal path has already been determined. In graph-based methods, many edges are added at once with no well-defined parent, so that it is not clear what the initial conditions are that need to be satisfied for each edge. Even in RRT*, the rewiring process reroutes an already differentially constrained path in the middle of the path, a more challenging problem than just concatenating to the end.

5.2 Results from Numerical Experiments

All simulations were run in C++, using a Linux operating system with a 2.3 GHz processor and 3.6 GB of RAM. The implementation of RRT* was taken from the Open Motion Planning Library (OMPL) [4], and these simulations rely on the quality of the OMPL implementation. We did adjust the search radius to match the lower bound in [9] plus 10%, and note that no steering parameter was used. Both PRM* and FMT* were implemented in OMPL, and for comparison purposes, we used the radius suggested in [9] for PRM* for both (in particular, we used an r_n of 10% over the lower bound given there, as opposed to the smaller r_n lower bound presented in this paper). Figures 2 - 5 show the results of simulations run in 2, 5, 7, and 10 dimensions, respectively, with no obstacles and 50% obstacle coverage (the obstacles are hyperrectangles). The initial state x_{init} was set to be the center of the unit hypercube (which was the configuration space), and $\mathscr{X}_{\text{goal}}$ was set to be the ball of radius $0.001^{1/d}$ centered at the 1-vector. The points represent simulations of the three algorithms on node sets increasing in size to the right. The error bars represent plus and minus one standard error of the mean, reflecting that some plots were run with 100 simulations at each point and others at 50 simulations per point. Note that PRM* was not simulated in 10D because it took prohibitively long to run.

In all the figures, FMT* dominates the other two algorithms, in that the FMT* curve is below and to the left of the other two. This difference becomes particularly prominent in higher dimensions and with more obstacles, which is exactly the regime in

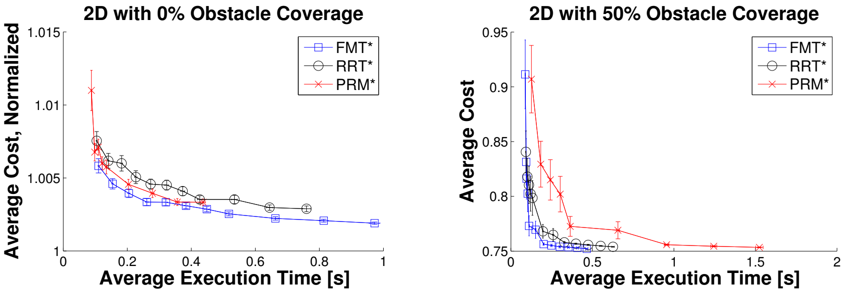

Fig. 2 Simulation results in 2 dimensions with and without obstacles. Left figure: (normalized) cost versus time with 0% obstacle coverage. Right figure: cost versus time with 50% obstacle coverage.

which sampling-based motion planning algorithms are particularly useful. Specifically, note that in Figures 4 and 5, the solution of RRT* never even dips below the solution of FMT* on the smallest node set, making it hard to estimate the speedup, but it is clearly multiple orders of magnitude. As compared to PRM*, FMT* still provides substantial speedups in the plots with 50% obstacle coverage, and even in the 0% obstacles plots for 5D and 7D, while both curves seem to plateau, FMT* reaches that plateau in about half the time of PRM*.

We also note that, although it is not exactly clear from these plots because the points are not labeled with the number of nodes, when all three algorithms are run on the same number of nodes, the curve for PRM* looks very much like that for FMT* but shifted to the right (same solution quality, but slower), while that of RRT* looks very much like that of FMT* but shifted up (lower solution quality in the same amount of time). This agrees with our heuristic analysis. It is of some interest to note that in 10D with 50% obstacle coverage, RRT* only returned a solution about 90% of the time across the number of nodes simulated, while FMT* had a 100% success rate for all but the smallest two node sets (200 nodes = 94%, 300 nodes = 96%). Finally, although the error bars depend on the number of simulations, each graph used the same number of simulations for each point, and thus it is noteworthy that RRT*'s error bars are uniformly larger than those of FMT* . This means that the solutions returned by FMT* are both higher quality and more consistent.

6 Conclusions

In this paper we have introduced and analyzed a novel probabilistic sampling-based motion planning algorithm called Fast Marching Tree algorithm (FMT*). This algorithm is asymptotically optimal and appears to converge *significantly* faster then its state-of-the-art counterparts. We used the weaker notion of convergence in probability, as opposed to convergence almost surely, and showed that the extra mathematical flexibility provides substantial theoretical and algorithmic benefits, including convergence rate bounds.

This paper leaves numerous important extensions open for further research. First, it is of interest to extend the FMT* algorithm to address problems with differential motion constraints and in non-metric spaces (relevant, e.g., for information-planning). Second, we plan to explore the convergence rate bounds provided by

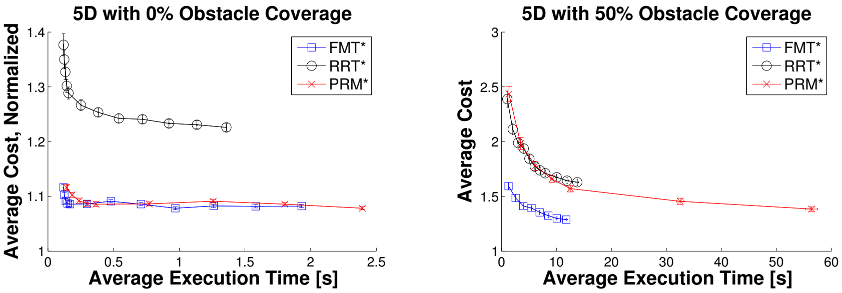

Fig. 3 Simulation results in 5 dimensions with and without obstacles. Left figure: (normalized) cost versus time with 0% obstacle coverage. Right figure: cost versus time with 50% obstacle coverage.

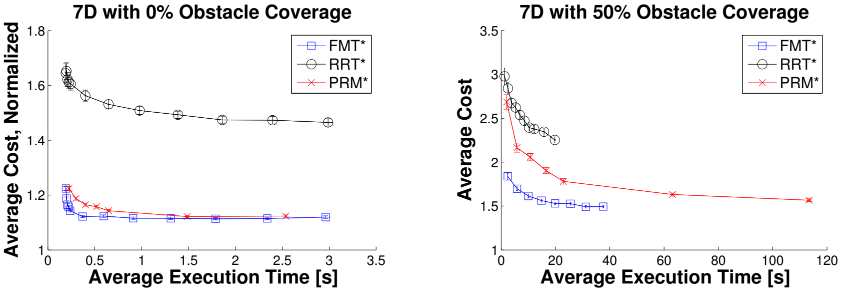

Fig. 4 Simulation results in 7 dimensions with and without obstacles. Left figure: (normalized) cost versus time with 0% obstacle coverage. Right figure: cost versus time with 50% obstacle coverage.

the proof of AO given here. Third, we plan to use this algorithm as the backbone for scalable information-theoretic planning algorithms. Fourth, we plan to extend the FMT* algorithm for solving the eikonal equation. Finally, we plan to test the performance of FMT* on mobile ground robots operating in dynamic environments.

Acknowledgements The authors gratefully acknowledge the contributions of Ashley Clark and Wolfgang Pointner to the implementation of FMT* and for the numerical experiments. This research was supported in part by NASA under the Space Technology Research Grants Program, Grant NNX12AQ43G.

Appendix

Proof (Proof of Lemma 1). To start, note that $\mathbb{P}(K_n^\beta \geq \alpha(M_n - 1)) + \mathbb{P}(A_n^c) \geq \mathbb{P}(\{K_n^\beta \geq \alpha(M_n - 1)\} \cup A_n^c) = 1 - \mathbb{P}(\{K_n^\beta < \alpha(M_n - 1)\} \cap A_n)$, where the first inequality follows from the union bound and the second equality follows from De Morgan's laws. Note that the event $\{K_n^\beta < \alpha(M_n - 1)\} \cap A_n$ is the event that each $B_{n,m}$ contains at least one node, and more than a $1 - \alpha$ fraction of the $B_{n,m}^\beta$ balls also contains at least one node.

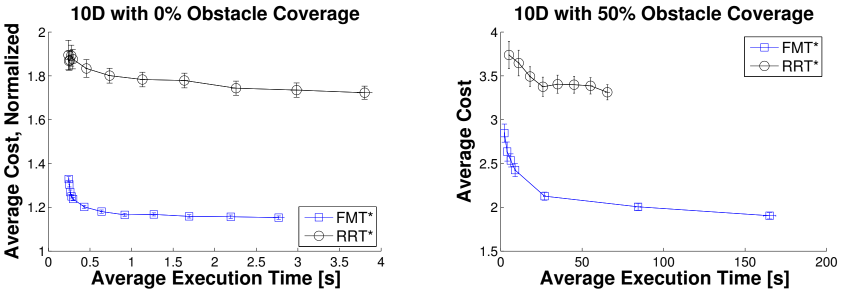

Fig. 5 Simulation results in 10 dimensions with and without obstacles. Left figure: (normalized) cost versus time with 0% obstacle coverage. Right figure: cost versus time with 50% obstacle coverage.

When two nodes x_i and x_{i+1}, $i \in \{1,\dots,M_n-2\}$, are contained in adjacent balls $B_{n,i}$ and $B_{n,i+1}$, respectively, their distance apart $\|x_{i+1}-x_i\|$ can be upper bounded by,

$$\begin{cases} \frac{\theta r_n}{2+\theta} + \frac{\beta r_n}{2+\theta} + \frac{\beta r_n}{2+\theta} & : \text{if } x_i \in B^{\beta}_{n,i} \text{ and } x_{i+1} \in B^{\beta}_{n,i+1} \\ \frac{\theta r_n}{2+\theta} + \frac{\beta r_n}{2+\theta} + \frac{r_n}{2+\theta} & : \text{if } x_i \in B^{\beta}_{n,i} \text{ or } x_{i+1} \in B^{\beta}_{n,i+1} \\ \frac{\theta r_n}{2+\theta} + \frac{r_n}{2+\theta} + \frac{r_n}{2+\theta} & : \text{otherwise,} \end{cases}$$

where the three bounds have been suggestively divided into a term for the distance between ball centers and a term each for the radii of the two balls containing the nodes. This bound also holds for $\|x_{M_n}-x_{M_n-1}\|$, although necessarily in one of the latter two bounds, since B^{β}_{n,M_n} being undefined precludes the possibility of the first bound. Thus we can rewrite the above bound, for $i \in \{1,\dots,M_n-1\}$, as $\|x_{i+1}-x_i\| \le \bar{c}(x_i)+\bar{c}(x_{i+1})$, where

$$\bar{c}(x_k) := \begin{cases} \frac{\theta r_n}{2(2+\theta)} + \frac{\beta r_n}{2+\theta} & : x_k \in B^{\beta}_{n,k}, \\ \frac{\theta r_n}{2(2+\theta)} + \frac{r_n}{2+\theta} & : x_k \notin B^{\beta}_{n,k}. \end{cases} \tag{3}$$

Again, $\bar{c}(x_{M_n})$ is still well-defined, but always takes the second value in equation (3) above. Let $L_{n,\alpha,\beta}$ be the length of a path that sequentially connects a set of nodes $\{x_1 = x_{\text{init}}, x_2, \dots, x_{M_n}\}$, such that $x_m \in B_{n,m}\ \forall m \in \{1,\dots,M_n\}$, and more than a $(1-\alpha)$ fraction of the nodes $x_1,\dots,x_{M_n-1}$ are also contained in their respective $B^{\beta}_{n,m}$ balls. The length $L_{n,\alpha,\beta}$ can then be upper bounded as follows

$$\begin{aligned} L_{n,\alpha,\beta} &= \sum_{k=1}^{M_n-1} \|x_{k+1}-x_k\| \le \sum_{k=1}^{M_n-1} 2\bar{c}(x_k) - \bar{c}(x_1) + \bar{c}(x_{M_n}) \\ &\le (M_n-1)\frac{\theta r_n}{2+\theta} + \lceil (1-\alpha)(M_n-1) \rceil \frac{2\beta r_n}{2+\theta} + \lfloor \alpha(M_n-1) \rfloor \frac{2 r_n}{2+\theta} + \frac{(1-\beta) r_n}{2+\theta} \\ &\le (M_n-1)\, r_n \frac{\theta + 2\alpha + 2(1-\alpha)\beta}{2+\theta} + \frac{(1-\beta) r_n}{2+\theta} \end{aligned}$$

$$\le M_n\, r_n \frac{\theta + 2\alpha + 2\beta}{2+\theta} + \frac{r_n}{2+\theta}. \tag{4}$$

In equation 4, $\lceil x \rceil$ denotes the smallest integer not less than x, while $\lfloor x \rfloor$ denotes the largest integer not greater than x. Furthermore, we can upper bound M_n as follows,

$$c(\sigma'_n) \ge \sum_{k=1}^{M_n-2} \|\sigma_n(\tau_{k+1}) - \sigma_n(\tau_k)\| + \|\sigma'_n(1) - \sigma_n(\tau_{M_n-1})\| \ge (M_n - 2)\frac{\theta r_n}{2+\theta} + \frac{r_n}{2(2+\theta)}$$
$$= M_n \frac{\theta r_n}{2+\theta} + \left(\frac{1}{2} - 2\theta\right)\frac{r_n}{2+\theta} \ge M_n \frac{\theta r_n}{2+\theta}, \tag{5}$$

where the last inequality follows from the assumption that $\theta < 1/4$. Combining equations (4) and (5) gives

$$L_{n,\alpha,\beta} \le c(\sigma'_n)\left(1 + \frac{2\alpha + 2\beta}{\theta}\right) + \frac{r_n}{2+\theta} = \kappa(\alpha,\beta,\theta)\, c(\sigma'_n) + \frac{r_n}{2+\theta}. \tag{6}$$

We will now show that when A_n occurs, c_n is no more than the length of the path connecting any sequence of M_n vertices tracing through the balls $B_{n,1}, \dots, B_{n,M_n}$ (this of course also implies $c_n < \infty$). Coupling this fact with equation (6), one can then conclude that the event $\{K_n^\beta < \alpha(M_n - 1)\} \cap A_n$ implies that $c_n \le \kappa(\alpha,\beta,\theta)\, c(\sigma'_n) + \frac{r_n}{2+\theta}$, which, in turn, would prove the lemma.

Let $x_1 = x_{\text{init}}$, $x_2 \in B_{n,2}$, $\dots$, $x_{M_n} \in B_{n,M_n} \subseteq \mathscr{X}_{\text{goal}}$. Note that the x_i's need not all be distinct. The following property holds for all $m \in \{2, \dots, M_n - 1\}$:

$$\|x_m - x_{m-1}\| \le \|x_m - \sigma_n(\tau_m)\| + \|\sigma_n(\tau_m) - \sigma_n(\tau_{m-1})\| + \|\sigma_n(\tau_{m-1}) - x_{m-1}\|$$
$$\le \frac{r_n}{2+\theta} + \frac{\theta r_n}{2+\theta} + \frac{r_n}{2+\theta} = r_n.$$

Similarly, one can write $\|x_{M_n} - x_{M_n-1}\| \le \frac{r_n}{2+\theta} + \frac{(\theta+1/2)r_n}{2+\theta} + \frac{r_n}{2(2+\theta)} = r_n$. Furthermore, we can lower bound the distance to the nearest obstacle for $m \in \{2, \dots, M_n - 1\}$ by:

$$\inf_{w \in X_{\text{obs}}} \|x_m - w\| \ge \inf_{w \in X_{\text{obs}}} \|\sigma_n(\tau_m) - w\| - \|x_m - \sigma_n(\tau_m)\| \ge \frac{3+\theta}{2+\theta} r_n - \frac{r_n}{2+\theta} = r_n,$$

where the second inequality follows from the assumed δ_n-clearance of the path σ_n. Again, similarly, one can write $\inf_{w \in X_{\text{obs}}} \|x_{M_n} - w\| \ge \inf_{w \in X_{\text{obs}}} \|x_m - \sigma_n(1)\| - \|\sigma_n(1) - w\| \ge \frac{3+\theta}{2+\theta} r_n - \frac{r_n}{2+\theta} = r_n$. Together, these two properties imply that, for $m \in \{2, \dots, M_n\}$, when a connection is attempted for x_m, x_{m-1} will be in the search radius and there will be no obstacles in that search radius. In particular, this implies that either the algorithm will return a feasible path before considering x_{M_n}, or it will consider x_{M_n} and connect it. Therefore, FMT* is guaranteed to return a feasible solution when the event A_n occurs. Since the remainder of this proof assumes that A_n occurs, we will also assume $c_n < \infty$.

Finally, assuming x_m is contained in an edge, let $c(x_m)$ denote the (unique) cost-to-come of x_m in the graph generated by FMT* at the end of the algorithm, just before the path is returned. If x_m is not contained in an edge, we set $c(x_m) = \infty$. Note that $c(\cdot)$ is well-defined, since if x_m is contained in any edge, it must be connected through a unique path to x_{init}. We claim that for all $m \in \{2, \dots, M_n\}$, either $c_n \leq \sum_{k=1}^{m-1} \|x_{k+1} - x_k\|$, or $c(x_m) \leq \sum_{k=1}^{m-1} \|x_{k+1} - x_k\|$. In particular, taking $m = M_n$, this would imply that $c_n \leq \min\{c(x_{M_n}), \sum_{k=1}^{M_n-1} \|x_{k+1} - x_k\|\} \leq \sum_{k=1}^{M_n-1} \|x_{k+1} - x_k\|$, which, as argued before, would imply the claim.

The claim is proved by induction on m. The case of $m = 1$ is trivial, since the first step in the FMT* algorithm is to make every collision-free connection between $x_{\text{init}} = x_1$ and the nodes contained in $B(x_{\text{init}}; r_n)$, which will include x_2 and, thus, $c(x_2) = \|x_2 - x_1\|$. Now suppose the claim is true for $m - 1$. There are four cases to consider:

1. $c_n \leq \sum_{k=1}^{m-2} \|x_{k+1} - x_k\|$,
2. $c(x_{m-1}) \leq \sum_{k=1}^{m-2} \|x_{k+1} - x_k\|$ and FMT* ends before considering x_m,
3. $c(x_{m-1}) \leq \sum_{k=1}^{m-2} \|x_{k+1} - x_k\|$ and $x_{m-1} \in H$ when x_m is first considered,
4. $c(x_{m-1}) \leq \sum_{k=1}^{m-2} \|x_{k+1} - x_k\|$ and $x_{m-1} \notin H$ when x_m is first considered.

Case 1: $c_n \leq \sum_{k=1}^{m-2} \|x_{k+1} - x_k\| \leq \sum_{k=1}^{m-1} \|x_{k+1} - x_k\|$, thus the claim is true for m.

Case 2: $c(x_{m-1}) < \infty$ implies that x_{m-1} enters H at some point during FMT*. However, if x_{m-1} were ever the minimum-cost element of H, x_m would have been considered, and thus FMT* must have returned a feasible solution before x_{m-1} was ever the minimum-cost element of H. Since the end-node of the solution returned must have been the minimum-cost element of H, $c_n \leq c(x_{m-1}) \leq \sum_{k=1}^{m-2} \|x_{k+1} - x_k\| \leq \sum_{k=1}^{m-1} \|x_{k+1} - x_k\|$, thus the claim is true for m.

Case 3: $x_{m-1} \in H$ when x_m is first considered, $\|x_m - x_{m-1}\| \leq r_n$, and there are no obstacles in $B(x_m; r_n)$. Therefore, x_m must be connected to some parent when it is first considered, and $c(x_m) \leq c(x_{m-1}) + \|x_m - x_{m-1}\| \leq \sum_{k=1}^{m-1} \|x_{k+1} - x_k\|$, thus the claim is true for m.

Case 4: When x_m is first considered, there must exist $z \in B(x_m; r_n)$ such that z is the minimum-cost element of H, while x_{m-1} has not even entered H yet. Note that again, since $B(x_m; r_n)$ intersects no obstacles and contains at least one node in H, x_m must be connected to some parent when it is first considered. Since $c(x_{m-1}) < \infty$, there is a well-defined path $\mathscr{P} = \{v_1, \dots, v_q\}$ from $x_{\text{init}} = v_1$ to $x_{m-1} = v_q$ for some $q \in \mathbb{N}$. Let $w = v_j$, where $j = \max_{i \in \{1,\dots,q\}} \{i : v_i \in H \text{ when } x_m \text{ is first considered}\}$. Then there are two subcases, either $w \in B(x_m; r_n)$ or $w \notin B(x_m; r_n)$. If $w \in B(x_m; r_n)$, then,

$$\begin{aligned} c(x_m) &\leq c(w) + \|x_m - w\| \leq c(w) + \|x_{m-1} - w\| + \|x_m - x_{m-1}\| \\ &\leq c(x_{m-1}) + \|x_m - x_{m-1}\| \leq \sum_{k=1}^{m-1} \|x_{k+1} - x_k\|, \end{aligned}$$

thus the claim is true for m (the second and third inequalities follow from the triangle inequality). If $w \notin B(x_m; r_n)$, then,

$$c(x_m) \leq c(z) + \|x_m - z\| \leq c(w) + r_n \leq c(x_{m-1}) + \|x_m - x_{m-1}\| \leq \sum_{k=1}^{m-1} \|x_{k+1} - x_k\|,$$

where the third inequality follows from the fact that $w \notin B(x_m, r_n)$, which means that any path through w to x_m, in particular the path $\mathscr{P} \cup x_m$, must traverse a distance of at least r_n between w and x_m. Thus, in the final subcase of the final case, the claim is true for m.

Hence, we can conclude that $c_n \leq \sum_{k=1}^{M_n-1} \|x_{k+1} - x_k\|$. As argued before, coupling this fact with equation (6), one can conclude that the event $\{K_n^\beta < \alpha(M_n - 1)\} \cap A_n$ implies that $c_n \leq \kappa(\alpha, \beta, \theta)\, c(\sigma_n') + \frac{r_n}{2+\theta}$, and the claim follows. □

References

[1] R. Alterovitz, S. Patil, and A. Derbakova. Rapidly-exploring roadmaps: Weighing exploration vs. refinement in optimal motion planning. In *Proc. IEEE Conf. on Robotics and Automation*, pages 3706–3712, 2011.

[2] J. Barraquand, L. Kavraki, R. Motwani, J.-C. Latombe, Tsai-Y. Li, and P. Raghavan. A random sampling scheme for path planning. In *International Journal of Robotics Research*, pages 249–264. Springer, 2000.

[3] R. Bohlin and L. E. Kavraki. Path planning using lazy PRM. In *Proc. IEEE Conf. on Robotics and Automation*, volume 1, pages 521–528, 2000.

[4] I. A. Şucan, M. Moll, and L. E. Kavraki. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, December 2012. http://ompl.kavrakilab.org.

[5] D. Hsu, J. C. Latombe, and R. Motwani R. Path planning in expansive configuration spaces. *International Journal of Computational Geometry & Applications*, 9:495–512, 1999.

[6] D. Hsu, J.-C. Latombe, and H. Kurniawati. On the probabilistic foundations of probabilistic roadmap planning. *International Journal of Robotics Research*, 25(7):627–643, 2006.

[7] L. Jaillet and T. Siméon. A PRM-based motion planner for dynamically changing environments. In *Proc. IEEE Conf. on Robotics and Automation*, volume 2, pages 1606–1611, 2004.

[8] L. Janson and M. Pavone. Fast Marching Trees: a fast marching sampling-based method for optimal motion planning in many dimensions – extended version. 2013. Available at: http://arxiv.org/abs/1306.3532.

[9] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *International Journal of Robotics Research*, 30(7):846–894, 2011.

[10] L.E. Kavraki, P. Svestka, J.-C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566 –580, 1996.

[11] M. Kobilarov. Cross-entropy motion planning. *International Journal of Robotics Research*, 31(7):855–871, 2012.

[12] A. M. Ladd and L. E. Kavraki. Measure theoretic analysis of probabilistic path planning. *IEEE Transactions on Robotics and Automation*, 20(2):229–242, 2004.

[13] S. Lavalle. *Planning Algorithms*. Cambridge University Press, 2006.

[14] S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. *International Journal of Robotics Research*, 20(5):378–400, 2001.

[15] J. D. Marble and K. E. Bekris. Towards small asymptotically near-optimal roadmaps. In *Proc. IEEE Conf. on Robotics and Automation*, 2012.

[16] J. M Phillips, N. Bedrossian, and L. E. Kavraki. Guided expansive spaces trees: A search strategy for motion- and cost-constrained state spaces. In *Proc. IEEE Conf. on Robotics and Automation*, volume 4, pages 3968–3973, 2004.

[17] E. Plaku, K. E. Bekris, B. Y. Chen, A. M. Ladd, and L. E. Kavraki. Sampling-based roadmap of trees for parallel motion planning. *IEEE Transactions on Robotics*, 21(4):597–608, 2005.

[18] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[19] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005.