Informed RRT*: Optimal Incremental Path Planning Focused through an Admissible Ellipsoidal Heuristic

3 authors, including:

Siddhartha Srinivasa

University of Washington Seattle

223 PUBLICATIONS   **6,363** CITATIONS

Some of the authors of this publication are also working on these related projects:

Project   Recurrent Predictive State Policy Networks View project

Project   YCB Benchmarking Project View project

Informed RRT*: Optimal Sampling-based Path Planning Focused via Direct Sampling of an Admissible Ellipsoidal Heuristic

Jonathan D. Gammell[1], Siddhartha S. Srinivasa[2], and Timothy D. Barfoot[1]

Abstract— **Rapidly-exploring random trees (RRTs) are popular in motion planning because they find solutions efficiently to single-query problems. Optimal RRTs (RRT*s) extend RRTs to the problem of finding the optimal solution, but in doing so asymptotically find the optimal path from the initial state to *every* state in the planning domain. This behaviour is not only inefficient but also inconsistent with their single-query nature.**

For problems seeking to minimize path length, the subset of states that can improve a solution can be described by a prolate hyperspheroid. We show that unless this subset is sampled directly, the probability of improving a solution becomes arbitrarily small in large worlds or high state dimensions. In this paper, we present an exact method to focus the search by directly sampling this subset.

The advantages of the presented sampling technique are demonstrated with a new algorithm, *Informed* RRT*. This method retains the same probabilistic guarantees on completeness and optimality as RRT* while improving the convergence rate and final solution quality. We present the algorithm as a simple modification to RRT* that could be further extended by more advanced path-planning algorithms. We show experimentally that it outperforms RRT* in rate of convergence, final solution cost, and ability to find difficult passages while demonstrating less dependence on the state dimension and range of the planning problem.

I. INTRODUCTION

The motion-planning problem is commonly solved by first discretizing the continuous state space with either a grid for graph-based searches or through random sampling for stochastic incremental searches. Graph-based searches, such as A* [1], are often *resolution complete* and *resolution optimal*. They are guaranteed to find the optimal solution, if a solution exists, and return failure otherwise (up to the resolution of the discretization). These graph-based algorithms do not scale well with problem size (e.g., state dimension or problem range).

Stochastic searches, such as Rapidly-exploring Random Trees (RRTs) [2], Probabilistic Roadmaps (PRMs) [3], and Expansive Space Trees (ESTs) [4], use sampling-based methods to avoid requiring a discretization of the state space. This allows them to scale more effectively with problem size and to directly consider kinodynamic constraints; however, the result is a less-strict completeness guarantee. RRTs are *probabilistically complete*, guaranteeing that the probability of finding a solution, if one exists, approaches unity as the number of iterations approaches infinity.

[1] J. D. Gammell and T. D. Barfoot are with the Autonomous Space Robotics Lab at the University of Toronto Institute for Aerospace Studies, Toronto, Ontario, Canada. Email: {jon.gammell, tim.barfoot}@utoronto.ca

[2] S. S. Srinivasa is with The Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA. Email: siddh@cs.cmu.edu
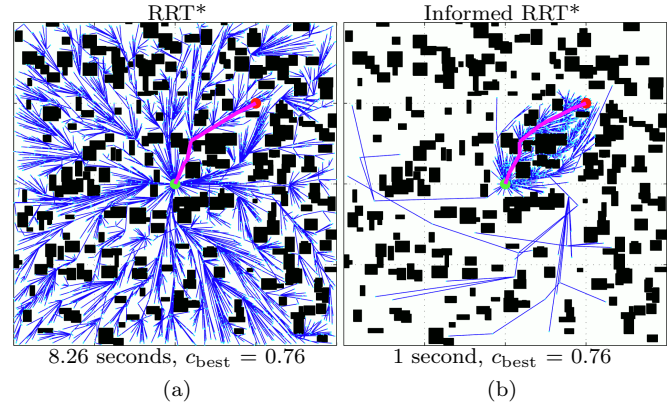
Fig. 1. Solutions of equivalent cost found by RRT* and Informed RRT* on a random world. After an initial solution is found, Informed RRT* focuses the search on an ellipsoidal informed subset of the state space, $X_{\hat{f}} \subseteq X$, that contains all the states that can improve the current solution regardless of homotopy class. This allows Informed RRT* to find a better solution faster than RRT* without requiring any additional user-tuned parameters.

Until recently, these sampling-based algorithms made no claims about the optimality of the solution. Urmson and Simmons [5] had found that using a heuristic to bias sampling improved RRT solutions, but did not formally quantify the effects. Ferguson and Stentz [6] recognized that the length of a solution bounds the possible improvements from above, and demonstrated an iterative anytime RRT method to solve a series of subsequently smaller planning problems. Karaman and Frazzoli [7] later showed that RRTs return a suboptimal path with probability one, demonstrating that all RRT-based methods will almost surely be suboptimal and presented a new class of optimal planners. They named their optimal variants of RRTs and PRMs, RRT* and PRM*, respectively. These algorithms are shown to be *asymptotically optimal*, with the probability of finding the optimal solution approaching unity as the number of iterations approaches infinity.

RRTs are not asymptotically optimal because the existing state graph biases future expansion. RRT* overcomes this by introducing incremental rewiring of the graph. New states are not only added to a tree, but also considered as replacement parents for existing nearby states in the tree. With uniform global sampling, this results in an algorithm that asymptotically finds the optimal solution to the planning problem by *asymptotically finding the optimal paths from the initial state to every state in the problem domain*. This is inconsistent with their single-query nature and becomes expensive in high dimensions.

In this paper, we present the *focused* optimal planning problem as it relates to the minimization of path length in $\mathbb{R}^n$. For such problems, a necessary condition to improve the solution at any iteration is the addition of states from
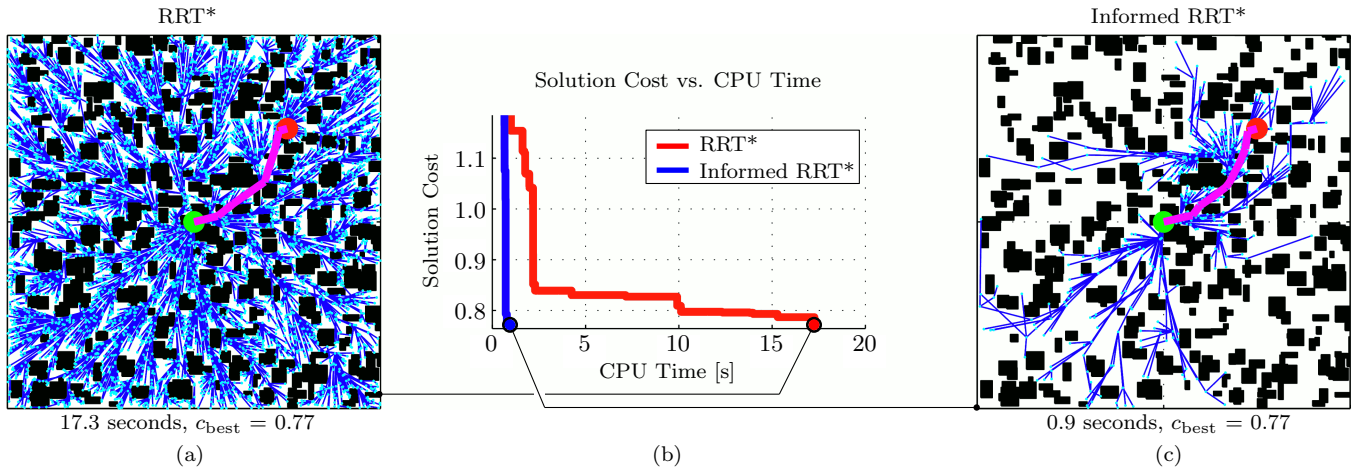
Fig. 2. The solution cost versus computational time for RRT* and Informed RRT* on a random world problem. Both planners were run until they found a solution of the same cost. Figs. (a, c) show the final result, while Fig. (b) shows the solution cost versus computational time. From Fig. (a), it can be observed that RRT* spends significant computational resources exploring regions of the planning problem that cannot possibly improve the current solution, while Fig. (c) demonstrates how Informed RRT* focuses the search. .

an ellipsoidal subset of the planning domain [6], [8]–[10]. We show that the probability of adding such states through uniform sampling becomes arbitrarily small as the size of the planning problem increases or the solution approaches the theoretical minimum, and present an exact method to sample the ellipsoidal subset directly. It is also shown that with strict assumptions (i.e., no obstacles) that this direct sampling results in linear convergence to the optimal solution.

This direct-sampling method allows for the creation of informed-sampling planners. Such a planner, Informed RRT*, is presented to demonstrate the advantages of *informed* incremental search (Fig. 1). Informed RRT* behaves as RRT* until a first solution is found, after which it only samples from the subset of states defined by an admissible heuristic to possibly improve the solution. This subset implicitly balances exploitation versus exploration and requires no additional tuning (i.e., there are no additional parameters) or assumptions (i.e., all relevant homotopy classes are searched). While heuristics may not always improve the search, their prominence in real-world planning demonstrates their practicality. In situations where they provide no additional information (e.g., when the informed subset includes the entire planning problem), Informed RRT* is equivalent to RRT*.

Informed RRT* is a simple modification to RRT* that demonstrates a clear improvement. In simulation, it performs as well as existing RRT* algorithms on simple configurations, and demonstrates order-of-magnitude improvements as the configurations become more difficult (Fig. 2). As a result of its focused search, the algorithm has less dependence on the dimension and domain of the planning problem as well as the ability to find better topologically distinct paths sooner. It is also capable of finding solutions within tighter tolerances of the optimum than RRT* with equivalent computation, and in the absence of obstacles can find the optimal solution to within machine zero in finite time (Fig. 3). It could also be used in combination with other algorithms, such as path-smoothing, to further reduce the search space.

The remainder of this paper is organized as follows. Section II presents a formal definition of the focused optimal planning problem and reviews the existing literature.

Section III presents a closed-form estimate of the subset of states that can improve a solution for problems seeking to minimize path length in $\mathbb{R}^n$ and analyzes the implications on RRT*-style algorithms. Section IV presents a method to sample this subset directly. Section V presents the Informed RRT* algorithm and Section VI presents simulation results comparing RRT* and Informed RRT* on simple planning problems of various size and configuration and random problems of various dimension. Section VII concludes the paper with a discussion of the technique and some related ongoing work.

II. BACKGROUND

A. Problem Definition

We define the optimal planning problem similarly to [7]. Let $X \subseteq \mathbb{R}^n$ be the state space of the planning problem. Let $X_{\mathrm{obs}} \subsetneq X$ be the states in collision with obstacles and $X_{\mathrm{free}} = X \setminus X_{\mathrm{obs}}$ be the resulting set of permissible states. Let $\mathbf{x}_{\mathrm{start}} \in X_{\mathrm{free}}$ be the initial state and $\mathbf{x}_{\mathrm{goal}} \in X_{\mathrm{free}}$ be the desired final state. Let $\sigma : [0, 1] \mapsto X$ be a sequence of states (a path) and Σ be the set of all nontrivial paths.

The optimal planning problem is then formally defined as the search for the path, σ^*, that minimizes a given cost function, $c : \Sigma \mapsto \mathbb{R}_{\geq 0}$, while connecting $\mathbf{x}_{\mathrm{start}}$ to $\mathbf{x}_{\mathrm{goal}}$ through free space,

$$\sigma^* = \arg\min_{\sigma \in \Sigma} \left\{ c\left(\sigma\right) \mid \sigma(0) = \mathbf{x}_{\mathrm{start}}, \sigma(1) = \mathbf{x}_{\mathrm{goal}}, \right.$$
$$\left. \forall s \in [0, 1], \sigma\left(s\right) \in X_{\mathrm{free}} \right\},$$

where $\mathbb{R}_{\geq 0}$ is the set of non-negative real numbers.

Let $f\left(\mathbf{x}\right)$ be the cost of an optimal path from $\mathbf{x}_{\mathrm{start}}$ to $\mathbf{x}_{\mathrm{goal}}$ constrained to pass through $\mathbf{x}$. Then the subset of states that can improve the current solution, $X_f \subseteq X$, can be expressed in terms of the current solution cost, c_{best},

$$X_f = \left\{ \mathbf{x} \in X \mid f\left(\mathbf{x}\right) < c_{\mathrm{best}} \right\}. \tag{1}$$

The problem of focusing RRT*'s search in order to increase the convergence rate is equivalent to increasing the probability of adding a random state from X_f.

As $f\left(\cdot\right)$ is generally unknown, a heuristic function, $\widehat{f}\left(\cdot\right)$,

| After an initial solution is found all possible improvements lie within an ellipse. | As the solution is improved the area of the ellipse decreases. | In the absence of obstacles the ellipse degenerates to a line. |

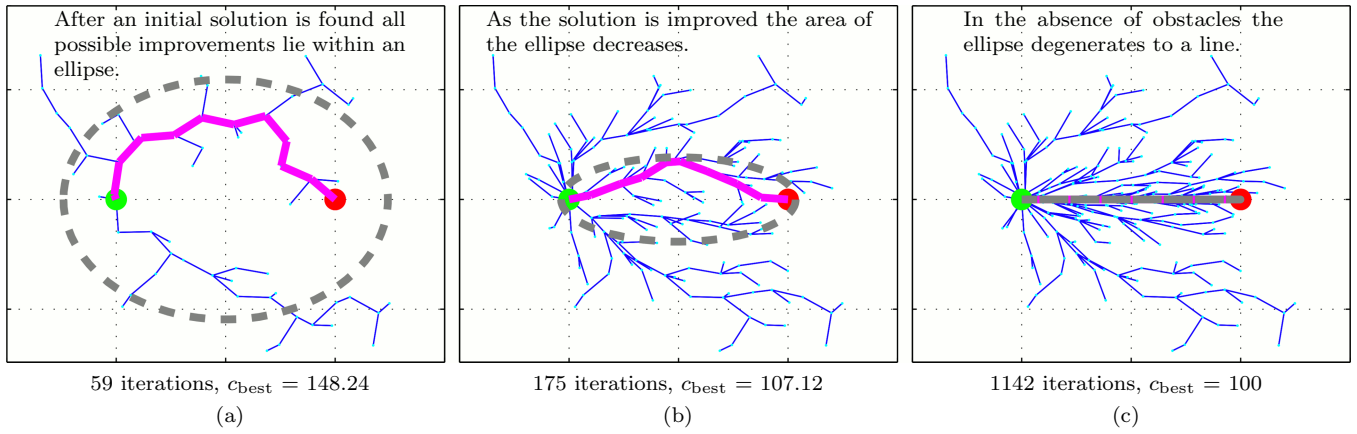| 59 iterations, $c_{\text{best}} = 148.24$ | 175 iterations, $c_{\text{best}} = 107.12$ | 1142 iterations, $c_{\text{best}} = 100$ |
| (a) | (b) | (c) |

Fig. 3. Informed RRT* converging to within machine zero of the optimum in the absence of obstacles. The start and goal states are shown as green and red, respectively, and are 100 units apart. The current solution is highlighted in magenta, and the ellipsoidal sampling domain, $X_{\widehat{f}}$, is shown as a grey dashed line for illustration. Improving the solution decreases the size of the sampling domain, creating a feedback effect that converges to within machine zero of the theoretical minimum. Fig. (a) shows the first solution at 59 iterations, (b) after 175 iterations, and (c), the final solution after 1142 iterations, at which point the ellipse has degenerated to a line between the start and goal.

may be used as an estimate. This heuristic is referred to as *admissible* if it never overestimates the true cost of the path, i.e., $\forall \mathbf{x} \in X$, $\widehat{f}(\mathbf{x}) \leq f(\mathbf{x})$. An estimate of X_f, $X_{\widehat{f}}$, can then be defined analogously to (1). For admissible heuristics, this estimate is guaranteed to completely contain the true set, $X_{\widehat{f}} \supseteq X_f$, and thus inclusion in the estimated set is also a necessary condition to improving the current solution.

B. Prior Work

Prior work to focus RRT and RRT* has relied on sample biasing, heuristic-based sample rejection, heuristic-based graph pruning, and/or iterative searches.

1) Sample Biasing: Sample biasing attempts to increase the frequency that states are sampled from X_f by biasing the distribution of samples drawn from X. This continues to add states from outside of X_f that cannot improve the solution. It also results in a nonuniform density over the problem being searched, violating a key RRT* assumption.

a) Heuristic-biased Sampling: Heuristic-biased sampling attempts to increase the probability of sampling X_f by weighting the sampling of X with a heuristic estimate of each state. It is used to improve the quality of a regular RRT by Urmson and Simmons [5] in the Heuristically Guided RRT (hRRT) by selecting states with a probability inversely proportional to their heuristic cost. The hRRT was shown to find better solutions than RRT; however, the use of RRTs means that the solution is almost surely suboptimal [7].

Kiesel et al. [11] use a two-stage process to create an RRT* heuristic in their *f-biasing* technique. A coarse abstraction of the planning problem is initially solved to provide a heuristic cost for each discrete state. RRT* then samples new states by randomly selecting a discrete state and sampling inside it with a continuous uniform distribution. The discrete sampling is biased such that states belonging to the abstracted solution have the highest probability of selection. This technique provides a heuristic bias for the full duration of the RRT* algorithm; however, to account for the discrete abstraction it maintains a nonzero probability of selecting every state. As a result, states that cannot improve the current solution are still sampled.

b) Path Biasing: Path-biased sampling attempts to increase the frequency of sampling X_f by sampling around the current solution path. This assumes that the current solution is either homotopic to the optimum or separated only by small obstacles. As this assumption is not generally true, path-biasing algorithms must also continue to sample globally to avoid local optima. The ratio of these two sampling methods is frequently a user-tuned parameter.

Alterovitz et al. [12] use path biasing to develop the Rapidly-exploring Roadmap (RRM). Once an initial solution is found, each iteration of the RRM either samples a new state or selects an existing state from the current solution and refines it. Path refinement occurs by connecting the selected state to its neighbours resulting in a graph instead of a tree.

Akgun and Stilman [13] use path biasing in their dual-tree version of RRT*. Once an initial solution is found, the algorithm spends a user-specified percentage of its iterations refining the current solution. It does this by randomly selecting a state from the solution path and then explicitly sampling from its Voronoi region. This increases the probability of improving the current path at the expense of exploring other homotopy classes. Their algorithm also employs sample rejection in exploring the state space (Section II-B.2).

Nasir et al. [14] combine path biasing with smoothing in their RRT*-Smart algorithm. When a solution is found, RRT*-Smart first smooths and reduces the path to its minimum number of states before using these states as biases for further sampling. This adds the complexity of a path-smoothing algorithm to the planner while still requiring global sampling to avoid local optima. While the path smoothing quickly reduces the cost of the current solution, it may also reduce the probability of finding a different homotopy class by removing the number of bias points about which samples are drawn and further violates the RRT* assumption of uniform density.

Kim et al. [15] use a visibility analysis to generate an initial bias in their Cloud RRT* algorithm. This bias is updated as a solution is found to further concentrate sampling near the path.

2) Heuristic-based Sample Rejection: Heuristic-based sample rejection attempts to increase the real-time rate of sampling X_f by using rejection sampling on X to sample

$X_{\widehat{f}}$. Samples drawn from a larger distribution are either kept or rejected based on their heuristic value. Akgun and Stilman [13] use such a technique in their algorithm. While this is computationally inexpensive for a single iteration, the number of iterations necessary to find a single state in $X_{\widehat{f}}$ is proportional to its size relative to the sampling domain. This becomes nontrivial as the solution approaches the theoretical minimum or the planning domain grows.

Otte and Correll [8] draw samples from a subset of the planning domain in their parallelized C-FOREST algorithm. This subset is defined as the hyperrectangle that bounds the prolate hyperspheroidal informed subset. While this improves the performance of sample rejection, its utility decreases as the dimension of the problem increases (Remark 2).

3) Graph Pruning: Graph pruning attempts to increase the real-time exploration of X_f by using a heuristic function to limit the graph to $X_{\widehat{f}}$. States in the planning graph with a heuristic cost greater than the current solution are periodically removed while global sampling is continued. The space-filling nature of RRTs biases the expansion of the pruned graph towards the perimeter of $X_{\widehat{f}}$. After the subset is filled, only samples from within $X_{\widehat{f}}$ itself can add new states to the graph. In this way, graph pruning becomes a rejection-sampling method after greedily filling the target subset. As adding a new state to an RRT requires a call to a nearest-neighbour algorithm, graph pruning will be more computationally expensive than simple sample rejection while still suffering from the same probabilistic limitations.

Karaman et al. [16] use graph pruning to implement an anytime version of RRT* that improves solutions during execution. They use the current vertex cost plus a heuristic estimate of the cost from the vertex to the goal to periodically remove states from the tree that cannot improve the current solution. As RRT* asymptotically approaches the optimal cost of a vertex *from above*, this is an inadmissible heuristic for the cost of a solution through a vertex (Section III). This can overestimate the heuristic cost of a vertex resulting in erroneous removal, especially early in the algorithm when the tree is coarse. Jordan and Perez [17] use the same inadmissible heuristic in their bidirectional RRT* algorithm.

Arslan and Tsiotras [18] use a graph structure and lifelong planning A* (LPA*) [19] techniques in the RRT# algorithm to prune the existing graph. Each existing state is given a LPA*-style key that is updated after the addition of each new state. Only keys that are less than the current best solution are updated, and only up-to-date keys are available for connection with newly drawn samples.

4) Anytime RRTs: Ferguson and Stentz [6] recognized that a solution bounds the subset of states that can provide further improvement from above. Their iterative RRT method, Anytime RRTs, solves a series of independent planning problems whose domains are defined by the previous solution. They represent these domains as ellipses [6, Fig. 2], but do not discuss how to generate samples. Restricting the planning domain encourages each RRT to find a better solution than the previous; however, to do so they must discard the states already found in $X_{\widehat{f}}$.

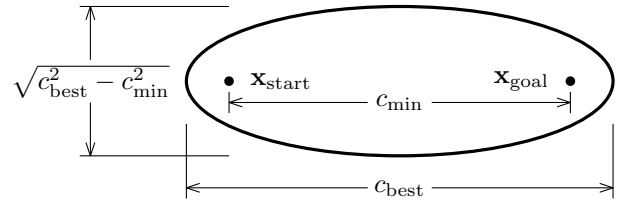The algorithm presented in this paper calculates $X_{\widehat{f}}$

Fig. 4. The heuristic sampling domain, $X_{\widehat{f}}$, for a $\mathbb{R}^2$ problem seeking to minimize path length is an ellipse with the initial state, $\mathbf{x}_{\mathrm{start}}$, and the goal state, $\mathbf{x}_{\mathrm{goal}}$ as focal points. The shape of the ellipse depends on both the initial and goal states, the theoretical minimum cost between the two, c_{min}, and the cost of the best solution found to date, c_{best}. The eccentricity of the ellipse is given by $c_{\mathrm{min}}/c_{\mathrm{best}}$.

explicitly and samples from it directly. Unlike path biasing it makes no assumptions about the homotopy class of the optimum and unlike heuristic biasing does not explore states that cannot improve the solution. As it is based on RRT*, it is able to keep all states found in $X_{\widehat{f}}$ for the duration of the search, unlike Anytime RRTs. By sampling $X_{\widehat{f}}$ directly, it always samples potential improvements regardless of the relative size of $X_{\widehat{f}}$ to X. This allows it to work effectively regardless of the size of the planning problem or the relative cost of the current solution to the theoretical minimum, unlike sample rejection and graph pruning methods. In problems where the heuristic does not provide any additional information, it performs identically to RRT*.

III. ANALYSIS OF THE ELLIPSOIDAL INFORMED SUBSET

Given a positive cost function, the cost of an optimal path from $\mathbf{x}_{\mathrm{start}}$ to $\mathbf{x}_{\mathrm{goal}}$ constrained to pass through $\mathbf{x} \in X$, $f(\mathbf{x})$, is equal to the cost of the optimal path from $\mathbf{x}_{\mathrm{start}}$ to $\mathbf{x}$, $g(\mathbf{x})$, plus the cost of the optimal path from $\mathbf{x}$ to $\mathbf{x}_{\mathrm{goal}}$, $h(\mathbf{x})$. As RRT*-based algorithms asymptotically approach the optimal path to every state *from above*, an admissible heuristic estimate, $\widehat{f}(\cdot)$, must estimate both these terms. A sufficient condition for admissibility is that the components, $\widehat{g}(\cdot)$ and $\widehat{h}(\cdot)$, are individually admissible heuristics of $g(\cdot)$ and $h(\cdot)$, respectively.

For problems seeking to minimize path length in $\mathbb{R}^n$, Euclidean distance is an admissible heuristic for both terms (even with motion constraints). This *informed* subset of states that may improve the current solution, $X_{\widehat{f}} \supseteq X_f$, can then be expressed in closed form in terms of the cost of the current solution, c_{best}, as

$$X_{\widehat{f}} = \left\{ \mathbf{x} \in X \mid ||\mathbf{x}_{\mathrm{start}} - \mathbf{x}||_2 + ||\mathbf{x} - \mathbf{x}_{\mathrm{goal}}||_2 \leq c_{\mathrm{best}} \right\},$$

which is the general equation of an n-dimensional prolate hyperspheroid (i.e., a special hyperellipsoid). The focal points are $\mathbf{x}_{\mathrm{start}}$ and $\mathbf{x}_{\mathrm{goal}}$, the transverse diameter is c_{best}, and the conjugate diameters are $\sqrt{c_{\mathrm{best}}^2 - c_{\mathrm{min}}^2}$ (Fig. 4).

Admissibility of $\widehat{f}(\cdot)$ makes adding a state in $X_{\widehat{f}}$ a necessary condition to improve the solution. With the space-filling nature of RRT, the probability of adding such a state quickly becomes the probability of sampling such a state[1]. Thus, the probability of improving the solution at any iteration by uniformly sampling a larger subset, $\mathbf{x}^{i+1} \sim \mathcal{U}(X_{\mathrm{s}})$, $X_{\mathrm{s}} \supseteq$

---

[1] States may be added to $X_{\widehat{f}}$ with a sample from outside the subset until it is filled to within the RRT growth-limiting parameter, η, of its boundary.

$X_{\widehat{f}}$, is less than or equal to the ratio of set measures $\lambda\left(\cdot\right)$,

$$P\left(c_{\text{best}}^{i+1} < c_{\text{best}}^i\right) \leq P\left(\mathbf{x}^{i+1} \in X_f\right) \qquad (2)$$
$$\leq P\left(\mathbf{x}^{i+1} \in X_{\widehat{f}}\right) = \frac{\lambda(X_{\widehat{f}})}{\lambda(X_{\text{s}})}.$$

Using the volume of a prolate hyperspheroid in $\mathbb{R}^n$ gives

$$P\left(c_{\text{best}}^{i+1} < c_{\text{best}}^i\right) \leq \frac{c_{\text{best}}^i \left(c_{\text{best}}^{i^2} - c_{\text{min}}^2\right)^{\frac{n-1}{2}} \zeta_n}{2^n \lambda(X_{\text{s}})}, \qquad (3)$$

with ζ_n being the volume of a unit n-ball.

Remark 1 (Rejection sampling): From (3) it can be observed that the probability of improving a solution through uniform sampling becomes arbitrarily small for large subsets (e.g., global sampling) or as the solution approaches the theoretical minimum.

Remark 2 (Rectangular rejection sampling): Let X_{s} be a hyperrectangle that tightly bounds the informed subset (i.e., the widths of each side correspond to the diameters of the prolate hyperspheroid) [8]. From (3), the probability that a sample drawn uniformly from X_{s} will be in $X_{\widehat{f}}$ is then $\frac{\zeta_n}{2^n}$, which decreases rapidly with n. For example, with $n = 6$ this gives a maximum 8% probability of improving a solution at each iteration through rejection sampling regardless of the specific solution, problem, or algorithm parameters.

Theorem 1 (Obstacle-free linear convergence): With uniform sampling of the informed subset, $\mathbf{x} \sim \mathcal{U}\left(X_{\widehat{f}}\right)$, the cost of the best solution, c_{best}, converges linearly to the theoretical minimum, c_{min}, in the absence of obstacles.

Proof: The heuristic value of a state is equal to the transverse diameter of a prolate hyperspheroid that passes through the state and has focal points at $\mathbf{x}_{\text{start}}$ and $\mathbf{x}_{\text{goal}}$. With uniform sampling, the expectation is then [20]

$$E\left[\widehat{f}\left(\mathbf{x}\right)\right] = \frac{nc_{\text{best}}^2 + c_{\text{min}}^2}{(n+1)c_{\text{best}}}. \qquad (4)$$

We assume that the RRT* rewiring parameter is greater than the diameter of the informed subset, similarly to how the proof of the asymptotic optimality of RRT* assumes that η is greater than the diameter of the planning problem [7]. The expectation of the solution cost, c_{best}^i, is then the expectation of the heuristic cost of a sample drawn from a prolate hyperspheroid of diameter c_{best}^{i-1}, i.e., $E\left[c_{\text{best}}^i\right] = E\left[\widehat{f}\left(\mathbf{x}^i\right)\right]$. From (4) it follows that the solution cost converges linearly with a rate, μ, that depends only on the state dimension [20],

$$\mu = \left.\frac{\partial E\left[c_{\text{best}}^i\right]}{\partial c_{\text{best}}^{i-1}}\right|_{c_{\text{best}}^{i-1} = c_{\text{min}}} = \frac{n-1}{n+1}. \qquad \blacksquare$$

While the obstacle-free assumption is impractical, Thm. 1 illustrates the fundamental effectiveness of direct informed sampling and provides possible insight for future work.

IV. Direct Sampling of an Ellipsoidal Subset

Uniformly distributed samples in a hyperellipsoid, $\mathbf{x}_{\text{ellipse}} \sim \mathcal{U}\left(X_{\text{ellipse}}\right)$, can be generated by transforming uniformly distributed samples from the unit n-ball, $\mathbf{x}_{\text{ball}} \sim \mathcal{U}\left(X_{\text{ball}}\right)$,

$$\mathbf{x}_{\text{ellipse}} = \mathbf{L}\mathbf{x}_{\text{ball}} + \mathbf{x}_{\text{centre}},$$

where $\mathbf{x}_{\text{centre}} = \left(\mathbf{x}_{f1} + \mathbf{x}_{f2}\right)/2$ is the centre of the hyperellipsoid in terms of its two focal points, $\mathbf{x}_{f1}$ and $\mathbf{x}_{f2}$, and $X_{\text{ball}} = \{\mathbf{x} \in X \mid \|\mathbf{x}\|_2 \leq 1\}$ [21].

This transformation can be calculated by Cholesky decomposition of the hyperellipsoid matrix, $\mathbf{S} \in \mathbb{R}^{n \times n}$,

$$\mathbf{L}\mathbf{L}^T \equiv \mathbf{S},$$

where

$$\left(\mathbf{x} - \mathbf{x}_{\text{centre}}\right)^T \mathbf{S}\left(\mathbf{x} - \mathbf{x}_{\text{centre}}\right) = 1,$$

with $\mathbf{S}$ having eigenvectors corresponding to the axes of the hyperellipsoid, $\{\mathbf{a}_i\}$, and eigenvalues corresponding to the squares of its radii, $\{r_i^2\}$. The transformation, $\mathbf{L}$, maintains the uniform distribution in X_{ellipse} [22].

For prolate hyperspheroids, such as $X_{\widehat{f}}$, the transformation can be calculated from just the transverse axis and the radii. The hyperellipsoid matrix in a coordinate system aligned with the transverse axis is the diagonal matrix

$$\mathbf{S} = \text{diag}\left\{\frac{c_{\text{best}}^2}{4}, \frac{c_{\text{best}}^2 - c_{\text{min}}^2}{4}, \ldots, \frac{c_{\text{best}}^2 - c_{\text{min}}^2}{4}\right\},$$

with a resulting decomposition of

$$\mathbf{L} = \text{diag}\left\{\frac{c_{\text{best}}}{2}, \frac{\sqrt{c_{\text{best}}^2 - c_{\text{min}}^2}}{2}, \ldots, \frac{\sqrt{c_{\text{best}}^2 - c_{\text{min}}^2}}{2}\right\}, \qquad (5)$$

where $\text{diag}\left\{\cdot\right\}$ denotes a diagonal matrix.

The rotation from the hyperellipsoid frame to the world frame, $\mathbf{C} \in SO\left(n\right)$, can be solved directly as a general Wahba problem [23]. It has been shown that a valid solution can be found even when the problem is underspecified [24]. The rotation matrix is given by

$$\mathbf{C} = \mathbf{U} \, \text{diag}\left\{1, \ldots, 1, \det\left(\mathbf{U}\right) \det\left(\mathbf{V}\right)\right\} \mathbf{V}^T, \qquad (6)$$

where $\det\left(\cdot\right)$ is the matrix determinant and $\mathbf{U} \in \mathbb{R}^{n \times n}$ and $\mathbf{V} \in \mathbb{R}^{n \times n}$ are unitary matrices such that $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \equiv \mathbf{M}$ via singular value decomposition. The matrix $\mathbf{M}$ is given by the outer product of the transverse axis in the world frame, $\mathbf{a}_1$, and the first column of the identity matrix, $\mathbf{1}_1$,

$$\mathbf{M} = \mathbf{a}_1 \mathbf{1}_1^T,$$

where

$$\mathbf{a}_1 = \left(\mathbf{x}_{\text{goal}} - \mathbf{x}_{\text{start}}\right) / \|\mathbf{x}_{\text{goal}} - \mathbf{x}_{\text{start}}\|_2.$$

A state uniformly distributed in the informed subset, $\mathbf{x}_{\widehat{f}} \sim \mathcal{U}\left(X_{\widehat{f}}\right)$, can thus be calculated from a sample drawn uniformly from a unit n-ball, $\mathbf{x}_{\text{ball}} \sim \mathcal{U}\left(X_{\text{ball}}\right)$, through a transformation (5), rotation (6), and translation,

$$\mathbf{x}_{\widehat{f}} = \mathbf{C}\mathbf{L}\mathbf{x}_{\text{ball}} + \mathbf{x}_{\text{centre}}. \qquad (7)$$

This procedure is presented algorithmically in Alg. 2.

V. Informed RRT*

An example algorithm using direct informed sampling, Informed RRT*, is presented in Algs. 1 and 2. It is identical to RRT* as presented in [7], with the addition of lines 3, 6, 7, 30, and 31. Like RRT*, it searches for the optimal path, σ^*, to a planning problem by incrementally building a tree in state space, $\mathcal{T} = (V, E)$, consisting of a set of vertices, $V \subseteq X_{\text{free}}$, and edges, $E \subseteq X_{\text{free}} \times X_{\text{free}}$. New vertices are added by growing the graph in free space towards randomly selected states. The graph is rewired with each new vertex such that the cost of the nearby vertices are minimized.

The algorithm differs from RRT* in that once a solution is found, it focuses the search on the part of the planning

Algorithm 1: Informed RRT*$(\mathbf{x}_{\text{start}}, \mathbf{x}_{\text{goal}})$

1   $V \leftarrow \{\mathbf{x}_{\text{start}}\}$;
2   $E \leftarrow \emptyset$;
3   $X_{\text{soln}} \leftarrow \emptyset$;
4   $\mathcal{T} = (V, E)$;
5   **for** iteration $= 1 \ldots N$ **do**
6     $c_{\text{best}} \leftarrow \min_{\mathbf{x}_{\text{soln}} \in X_{\text{soln}}} \{\text{Cost}(\mathbf{x}_{\text{soln}})\}$;
7     $\mathbf{x}_{\text{rand}} \leftarrow \text{Sample}(\mathbf{x}_{\text{start}}, \mathbf{x}_{\text{goal}}, c_{\text{best}})$;
8     $\mathbf{x}_{\text{nearest}} \leftarrow \text{Nearest}(\mathcal{T}, \mathbf{x}_{\text{rand}})$;
9     $\mathbf{x}_{\text{new}} \leftarrow \text{Steer}(\mathbf{x}_{\text{nearest}}, \mathbf{x}_{\text{rand}})$;
10   **if** $\text{CollisionFree}(\mathbf{x}_{\text{nearest}}, \mathbf{x}_{\text{new}})$ **then**
11      $V \leftarrow \cup \{\mathbf{x}_{\text{new}}\}$;
12      $X_{\text{near}} \leftarrow \text{Near}(\mathcal{T}, \mathbf{x}_{\text{new}}, r_{\text{RRT}^*})$;
13      $\mathbf{x}_{\text{min}} \leftarrow \mathbf{x}_{\text{nearest}}$;
14      $c_{\text{min}} \leftarrow \text{Cost}(\mathbf{x}_{\text{min}}) + c \cdot \text{Line}(\mathbf{x}_{\text{nearest}}, \mathbf{x}_{\text{new}})$;
15   **for** $\forall \mathbf{x}_{\text{near}} \in X_{\text{near}}$ **do**
16       $c_{\text{new}} \leftarrow \text{Cost}(\mathbf{x}_{\text{near}}) + c \cdot \text{Line}(\mathbf{x}_{\text{near}}, \mathbf{x}_{\text{new}})$;
17   **if** $c_{\text{new}} < c_{\text{min}}$ **then**
18   **if** $\text{CollisionFree}(\mathbf{x}_{\text{near}}, \mathbf{x}_{\text{new}})$ **then**
19         $\mathbf{x}_{\text{min}} \leftarrow \mathbf{x}_{\text{near}}$;
20         $c_{\text{min}} \leftarrow c_{\text{new}}$;
21      $E \leftarrow E \cup \{(\mathbf{x}_{\text{min}}, \mathbf{x}_{\text{new}})\}$;
22   **for** $\forall \mathbf{x}_{\text{near}} \in X_{\text{near}}$ **do**
23       $c_{\text{near}} \leftarrow \text{Cost}(\mathbf{x}_{\text{near}})$;
24       $c_{\text{new}} \leftarrow \text{Cost}(\mathbf{x}_{\text{new}}) + c \cdot \text{Line}(\mathbf{x}_{\text{new}}, \mathbf{x}_{\text{near}})$;
25   **if** $c_{\text{new}} < c_{\text{near}}$ **then**
26   **if** $\text{CollisionFree}(\mathbf{x}_{\text{new}}, \mathbf{x}_{\text{near}})$ **then**
27         $\mathbf{x}_{\text{parent}} \leftarrow \text{Parent}(\mathbf{x}_{\text{near}})$;
28         $E \leftarrow E \setminus \{(\mathbf{x}_{\text{parent}}, \mathbf{x}_{\text{near}})\}$;
29         $E \leftarrow E \cup \{(\mathbf{x}_{\text{new}}, \mathbf{x}_{\text{near}})\}$;
30   **if** $\text{InGoalRegion}(\mathbf{x}_{\text{new}})$ **then**
31       $X_{\text{soln}} \leftarrow X_{\text{soln}} \cup \{\mathbf{x}_{\text{new}}\}$;
32   **return** $\mathcal{T}$;

Algorithm 2: Sample $(\mathbf{x}_{\text{start}}, \mathbf{x}_{\text{goal}}, c_{\text{max}})$

1   **if** $c_{\text{max}} < \infty$ **then**
2     $c_{\text{min}} \leftarrow \left\| \mathbf{x}_{\text{goal}} - \mathbf{x}_{\text{start}} \right\|_2$;
3     $\mathbf{x}_{\text{centre}} \leftarrow \left(\mathbf{x}_{\text{start}} + \mathbf{x}_{\text{goal}} \right) / 2$;
4     $\mathbf{C} \leftarrow \text{RotationToWorldFrame}(\mathbf{x}_{\text{start}}, \mathbf{x}_{\text{goal}})$;
5     $r_1 \leftarrow c_{\text{max}}/2$;
6     $\{r_i\}_{i=2,\ldots,n} \leftarrow \left(\sqrt{c_{\text{max}}^2 - c_{\text{min}}^2} \right) / 2$;
7     $\mathbf{L} \leftarrow \text{diag}\{r_1, r_2, \ldots, r_n\}$;
8     $\mathbf{x}_{\text{ball}} \leftarrow \text{SampleUnitNBall}$;
9     $\mathbf{x}_{\text{rand}} \leftarrow (\mathbf{CL}\mathbf{x}_{\text{ball}} + \mathbf{x}_{\text{centre}}) \cap X$;
10   **else**
11     $\mathbf{x}_{\text{rand}} \sim \mathcal{U}(X)$;
12   **return** $\mathbf{x}_{\text{rand}}$;

---

most planning problems this rotation matrix only needs to be calculated at the beginning of the problem.

SampleUnitNBall: The function, SampleUnitNBall returns a uniform sample from the volume of an n-ball of unit radius centred at the origin, i.e. $\mathbf{x}_{\text{ball}} \sim \mathcal{U}(X_{\text{ball}})$.

A. Calculating the Rewiring Radius

At each iteration, the rewiring radius, r_{RRT^*}, must be large enough to guarantee almost-sure asymptotic convergence while being small enough to only generate a tractable number of rewiring candidates. Karaman and Frazzoli [7] present a lower-bound for this rewiring radius in terms of the measure of the problem space and the number of vertices in the graph. Their expression assumes a uniform distribution of samples of a unit square. As Informed RRT* uniformly samples the *subset* of the planning problem that can improve the solution, a rewiring radius can be calculated from the measure of this informed subset and the related vertices inside it. This updated radius reduces the amount of rewiring necessary and further improves the performance of Informed RRT*. Ongoing work is focused on finding the exact form of this expression, but the radius provided by [7] appears appropriate. There also exists a k-nearest neighbour version of this expression.

VI. SIMULATIONS

Informed RRT* was compared to RRT* on a variety of simple planning problems (Figs. 5 to 7) and randomly generated worlds (e.g., Figs. 1, 2). Simple problems were used to test specific challenges, while the random worlds were used to provide more challenging problems in a variety of state dimensions.

Fig. 5(a) was used to examine the effects of the problem range and the ability to find paths within a specified tolerance of the true optimum, with the width of the obstacle, w, selected randomly. Fig. 5(b) was used to demonstrate Informed RRT*'s ability to find topologically distinct solutions, with the position of the narrow passage, y_g, selected randomly. For these toy problems, experiments were ended when the planner found a solution cost within the target tolerance of the optimum. Random worlds, as in Fig. 2, were used to test Informed RRT* on more complicated problems and in higher state dimensions by giving the algorithms 60 seconds to improve their initial solutions. For each variation of every experiment, 100 different runs of both RRT* and Informed RRT* were performed with a common pseudo-random seed and map.
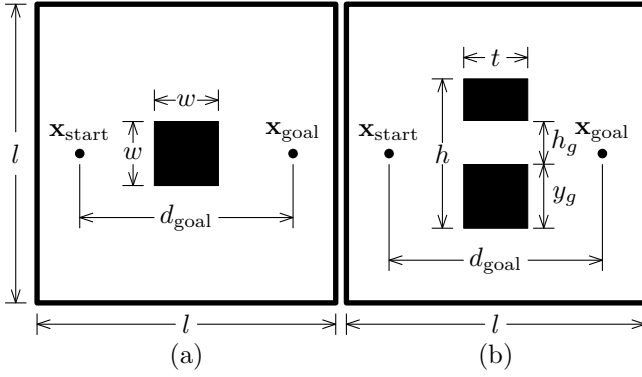
---

problem that can improve the solution. It does this through direct sampling of the ellipsoidal heuristic. As solutions are found (line 30), Informed RRT* adds them to a list of possible solutions (line 31). It uses the minimum of this list (line 6) to calculate and sample $X_{\widehat{f}}$ directly (line 7). As is conventional, we take the minimum of an empty set to be infinity. The new subfunctions are described below, while descriptions of subfunctions common to RRT* can be found in [7]:

Sample: Given two poses, $\mathbf{x}_{\text{from}}, \mathbf{x}_{\text{to}} \in X_{\text{free}}$ and a maximum heuristic value, $c_{\text{max}} \in \mathbb{R}$, the function $\text{Sample}(\mathbf{x}_{\text{from}}, \mathbf{x}_{\text{to}}, c_{\text{max}})$ returns independent and identically distributed (i.i.d.) samples from the state space, $\mathbf{x}_{\text{new}} \in X$, such that the cost of an optimal path between $\mathbf{x}_{\text{from}}$ and $\mathbf{x}_{\text{to}}$ that is constrained to go through $\mathbf{x}_{\text{new}}$ is less than c_{max} as described in Section III and Alg. 2. In most planning problems, $\mathbf{x}_{\text{from}} \equiv \mathbf{x}_{\text{start}}$, $\mathbf{x}_{\text{to}} \equiv \mathbf{x}_{\text{goal}}$, and lines 2 to 4 of Alg. 2 can be calculated once at the start of the problem.

InGoalRegion: Given a pose, $\mathbf{x} \in X_{\text{free}}$, the function $\text{InGoalRegion}(\mathbf{x})$ returns True if and only if the state is in the goal region, X_{goal}, as defined by the planning problem, otherwise it returns False. One common goal region is a ball of radius r_{goal} centred about the goal, i.e.,

$$X_{\text{goal}} = \left\{ \mathbf{x} \in X_{\text{free}} \mid \|\mathbf{x} - \mathbf{x}_{\text{goal}}\|_2 \leq r_{\text{goal}} \right\}.$$

RotationToWorldFrame: Given two poses as the focal points of a hyperellipsoid, $\mathbf{x}_{\text{from}}, \mathbf{x}_{\text{to}} \in X$, the function $\text{RotationToWorldFrame}(\mathbf{x}_{\text{from}}, \mathbf{x}_{\text{to}})$ returns the rotation matrix, $\mathbf{C} \in SO(n)$, from the hyperellipsoid-aligned frame to the world frame as per (6). As previously discussed, in

Fig. 5. The two planning problems used in Section VI. The width of the obstacle, w, and the location of the gap, y_g, were selected randomly for each experimental run.

The algorithms share the same unoptimized code, allowing for the comparison of relative computational time[2]. While further optimization would reduce the effect of graph size on the computational cost and reduce the difference between the two planners, as they have approximately the same cost per iteration it will not effect the order. To minimize the effects of the steer parameter on our results, we set it equal to the RRT* rewiring radius at each iteration calculated from $\gamma_{\mathrm{RRT}} = 1.1\gamma^*_{\mathrm{RRT}}$, a choice we found improved the performance of RRT*. As discussed in Section V-A, for Informed RRT* we calculated the rewiring radius for the subproblem defined by the current solution using the expression in [7].

Experiments varying the width of the problem range, l, while keeping a fixed distance between the start and goal show that Informed RRT* finds a suitable solution in approximately the same time regardless of the relative size of the problem (Fig. 8). As a result of considering only the informed subset once an initial solution is found, the size of the search space is independent of the planning range (Fig. 6). In contrast, the time needed by RRT* to find a similar solution increases as the problem range grows as proportionately more time is spent searching states that cannot improve the solution (Fig. 8).

Experiments varying the target solution cost show that Informed RRT* is capable of finding near-optimal solutions in significantly fewer iterations than RRT* (Fig. 9). The direct sampling of the informed subset increases density around the optimal solution faster than global sampling and therefore increases the probability of improving the solution and further focusing the search. In contrast, RRT* has uniform density across the entire planning domain and improving the solution actually *decreases* the probability of finding further improvements (Fig. 6).

Experiments varying the height of h_g in Fig. 5(b) demonstrate that Informed RRT* finds difficult passages that improve the current solution, regardless of their homotopy class, quicker than RRT* (Fig. 10). Once again, the result of considering only the informed subset is an increased state density in the region of the planning problem that includes the optimal solution. Compared to global sampling, this increases the probability of sampling within difficult passages, such as narrow gaps between obstacles, decreasing the time necessary

[2]Experiments were run in Ubuntu 12.04 on an Intel i5-2500K CPU with 8GB of RAM.
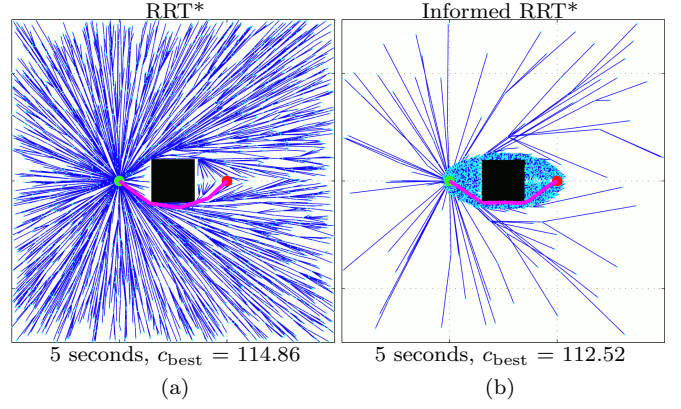

Fig. 6. An example of Fig. 5(a) after 5 seconds for a problem with an optimal solution cost of 112.01. Note that the presence of an obstacle provides a lower bound on the size of the ellipsoidal subset but that Informed RRT* still searches a significantly reduced domain than RRT*, increasing both the convergence rate and quality of final solution.
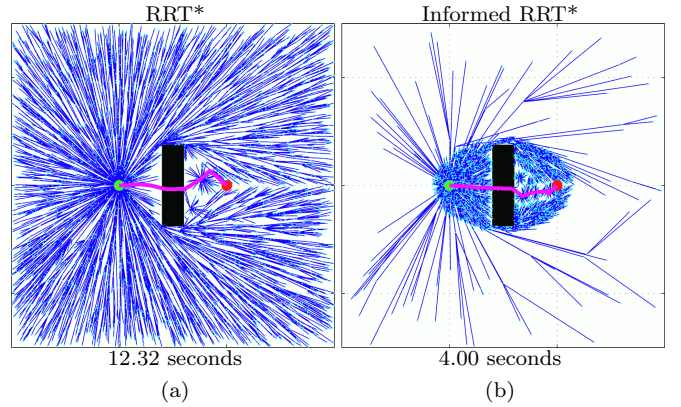

Fig. 7. An example of Fig. 5(b) for a 3% off-centre gap. By focusing the search space on the subset of states that may improve an initial solution flanking the obstacle, Informed RRT* is able to find a path through the narrow opening in 4.00 seconds while RRT* requires 12.32 seconds.

to find such solutions (Fig. 7).

Finally, experiments on random worlds demonstrate that the improvements of Informed RRT* apply to a wide range of planning problems and state dimensions (Fig. 11).

VII. DISCUSSION & CONCLUSION

In this paper, we discuss that a necessary condition for RRT* algorithms to improve a solution is the addition of a state from a subset of the planning problem, $X_f \subseteq X$. For problems seeking to minimize path length in $\mathbb{R}^n$, this subset can be estimated, $X_{\hat{f}} \supseteq X_f$, by a prolate hyperspheroid (a special type of hyperellipsoid) with the initial and goal states as focal points. It is shown that the probability of adding a new state from this subset through rejection sampling of a larger set becomes arbitrarily small as the dimension of the problem increases, the size of the sampled set increases, or the solution approaches the theoretical minimum. A simple method to sample $X_{\hat{f}}$ directly is presented that allows for the creation of informed-sampling planners, such as Informed RRT*. It is shown that Informed RRT* outperforms RRT* in the ability to find near-optimal solutions in finite time regardless of state dimension without requiring any assumptions about the optimal homotopy class.
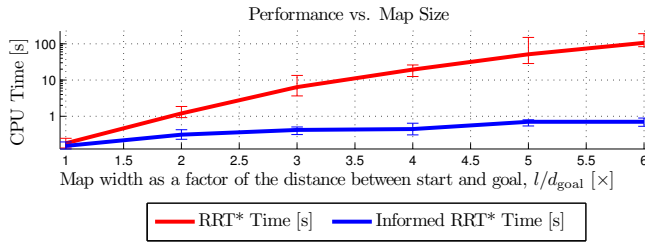
Informed RRT* uses heuristics to shrink the planning

Fig. 8. The median computational time needed by RRT* and Informed RRT* to find a path within 2% of the optimal cost in $\mathbb{R}^2$ for various map widths, l, for the problem in Fig. 5(a). Error bars denote a nonparametric 95% confidence interval for the median number of iterations calculated from 100 independent runs.
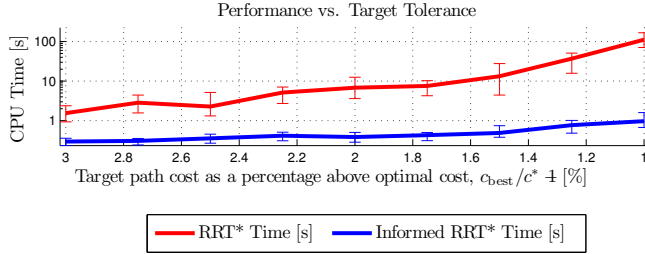

Fig. 9. The median computational time needed by RRT* and Informed RRT* to find a path within the specified tolerance of the optimal cost, c^*, in $\mathbb{R}^2$ for the problem in Fig. 5(a). Error bars denote a nonparametric 95% confidence interval for the median number of iterations calculated from 100 independent runs.

problem to subsets of the original domain. This makes it inherently dependent on the current solution cost, as it cannot focus the search when the associated prolate hyperspheroid is larger than the planning problem itself. Similarly, it can only shrink the subset down to the lower bound defined by the optimal solution. We are currently investigating techniques to focus the search without requiring an initial solution. These techniques, such as Batch Informed Trees (BIT*) [25], incrementally *increase* the search subset. By doing so, they prioritize the initial search of low-cost solutions.

An open motion planning library (OMPL) implementation of Informed RRT* is described at http://asrl.utias.utoronto.ca/code.
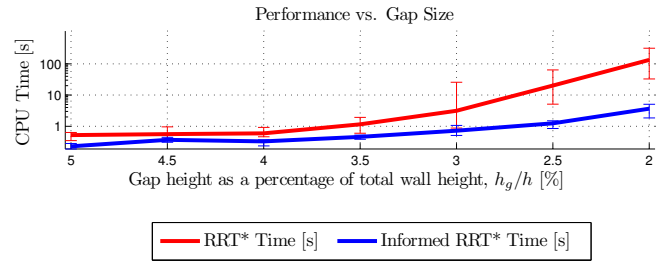
Fig. 10. The median computational time needed by RRT* and Informed RRT* to find a path cheaper than flanking the obstacle for various gap ratios, h_g/h for the problem defined in Fig. 5(b). Error bars denote a nonparametric 95% confidence interval for the median number of iterations calculated from 100 independent runs.
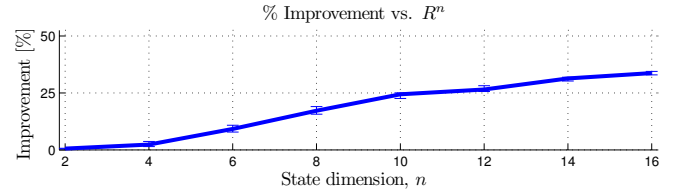

Fig. 11. The median performance of RRT* and Informed RRT* 60 seconds after finding an initial solution for random worlds (e.g., Figs. 1, 2) in $\mathbb{R}^n$. Plotted as the relative difference in cost, $(c_{\text{best}}^{\text{RRT*}} - c_{\text{best}}^{\text{Informed RRT*}})/(c_{\text{best}}^{\text{RRT*}})$. Error bars denote a nonparametric 95% confidence interval for the median number of iterations calculated from 100 independent runs.

[7] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *IJRR*, 30(7): 846–894, 2011.
[8] M. Otte and N. Correll, "C-FOREST: Parallel shortest path planning with superlinear speedup," *TRO*, 29(3): 798–806, Jun. 2013
[9] Y. Gabriely and E. Rimon, "CBUG: A quadratically competitive mobile robot navigation algorithm," *TRO*, 24(6): 1451–1457, Dec. 2008.
[10] N. Gasilov, M. Dogan, and V. Arici, "Two-stage shortest path algorithm for solving optimal obstacle avoidance problem," *IETE Jour. of Research*, 57(3): 278–285, May 2011.
[11] S. Kiesel, E. Burns, and W. Ruml, "Abstraction-guided sampling for motion planning," *SoCS*, 2012.
[12] R. Alterovitz, S. Patil, and A. Derbakova, "Rapidly-exploring roadmaps: Weighing exploration vs. refinement in optimal motion planning," *ICRA*, 3706–3712, 2011.
[13] B. Akgun and M. Stilman, "Sampling heuristics for optimal motion planning in high dimensions," *IROS*, 2640–2645, 2011.
[14] J. Nasir, F. Islam, U. Malik, Y. Ayaz, O. Hasan, M. Khan, and M. S. Muhammad, "RRT*-SMART: A rapid convergence implementation of RRT*," *Int. Jour. of Adv. Robotic Systems*, 10, 2013.
[15] D. Kim, J. Lee, and S. Yoon, "Cloud RRT*: Sampling cloud based RRT*," *ICRA*, 2014.
[16] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the RRT*," *ICRA*, 1478–1483, 2011.
[17] M. Jordan and A. Perez, "Optimal bidirectional rapidly-exploring random trees," CSAIL, MIT, MIT-CSAIL-TR-2013-021, 2013.
[18] O. Arslan and P. Tsiotras, "Use of relaxation methods in sampling-based algorithms for optimal motion planning," *ICRA*, 2013.
[19] S. Koenig, M. Likhachev, and D. Furcy, "Lifelong planning A*," *Artificial Intelligence*, 155(1–2): 93–146, 2004.
[20] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "On recursive random prolate hyperspheroids," Autonomous Space Robotics Lab, University of Toronto, TR-2014-JDG002, 2014. arXiv:1403.7664 [math.ST]
[21] H. Sun and M. Farooq, "Note on the generation of random points uniformly distributed in hyper-ellipsoids," in *Fifth Int. Conf. on Information Fusion*, 1: 489–496, 2002.
[22] J. D. Gammell, and T. D. Barfoot, "The probability density function of a transformation-based hyperellipsoid sampling technique," Autonomous Space Robotics Lab, University of Toronto, TR-2014-JDG004, 2014. arXiv:1404.1347 [math.ST]
[23] G. Wahba, "A least squares estimate of satellite attitude," *SIAM Review*, 7: 409, 1965.
[24] A. H. J. de Ruiter and J. R. Forbes, "On the solution of Wahba's problem on SO(n)," *Jour. of the Astronautical Sciences*, 2014, to appear.
[25] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "BIT*: Batch informed trees for optimal sampling-based planning via dynamic programming on implicit random geometric graphs," Autonomous Space Robotics Lab, University of Toronto, TR-2014-JDG006, 2014. arXiv:1405.5848 [cs.RO]

REFERENCES

[1] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *TSSC*, 4(2): 100–107, Jul. 1968
[2] S. M. LaValle and J. J. Kuffner Jr., "Randomized kinodynamic planning," *IJRR*, 20(5): 378–400, 2001.
[3] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *TRA*, 12(4): 566–580, 1996.
[4] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *IJRR*, 21(3): 233–255, 2002.
[5] C. Urmson and R. Simmons, "Approaches for heuristically biasing RRT growth," *IROS*, 2: 1178–1183, 2003.
[6] D. Ferguson and A. Stentz, "Anytime RRTs," *IROS*, 5369–5375, 2006.