

Interleaving Optimization with Sampling-Based Motion Planning (IOS-MP): Combining Local Optimization with Global Exploration

Alan Kuntz, Chris Bowen, and Ron Alterovitz

Abstract—Computing globally optimal motion plans for a robot is challenging in part because it requires analyzing a robot’s configuration space simultaneously from both a macroscopic viewpoint (i.e., considering paths in multiple homotopic classes) and a microscopic viewpoint (i.e., locally optimizing path quality). We introduce Interleaved Optimization with Sampling-based Motion Planning (IOS-MP), a new method that effectively combines global exploration and local optimization to quickly compute high quality motion plans. Our approach combines two paradigms: (1) asymptotically-optimal sampling-based motion planning, which is effective at global exploration but relatively slow at locally refining paths, and (2) optimization-based motion planning, which locally optimizes paths quickly but lacks a global view of the configuration space. IOS-MP iteratively alternates between global exploration and local optimization, sharing information between the two, to improve motion planning efficiency. We evaluate IOS-MP as it scales with respect to dimensionality and complexity, as well as demonstrate its effectiveness on a 7-DOF manipulator for tasks specified using goal configurations and workspace goal regions.

I. INTRODUCTION

Robots are increasingly entering domains such as transportation, surgery, and home assistance where safe interaction with people is critical. This interaction motivates robots which are capable of planning high quality motions under short time horizons. Unfortunately, the landscape of feasible motion plans in a robot’s configuration space can be extremely complex, consisting of many local minima, with paths spanning multiple homotopic classes. The complexity of this landscape means that an ideal motion planning algorithm must employ a macroscopic global view, considering paths in multiple homotopic classes, while also taking a microscopic local view, ensuring plans are as close to locally optimal as possible. Our work focusses on unifying these two perspectives by interleaving path refinement through local optimization, with global exploration through sampling-based motion planning. In this way, we compute high quality, locally optimized plans while continuing to explore the global landscape for as long as time allows.

Sampling-based motion planning methods typically take the macroscopic view, drawing samples from the robot’s entire configuration space to build a graph that progressively covers more and more of the space. The global nature

This research was supported in part by the National Science Foundation (NSF) under award IIS-1149965. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF.

Alan Kuntz, Chris Bowen, and Ron Alterovitz are with the Department of Computer Science, University of North Carolina at Chapel Hill, NC 27599-3175, USA {adkuntz, cbbowen, ron}@cs.unc.edu

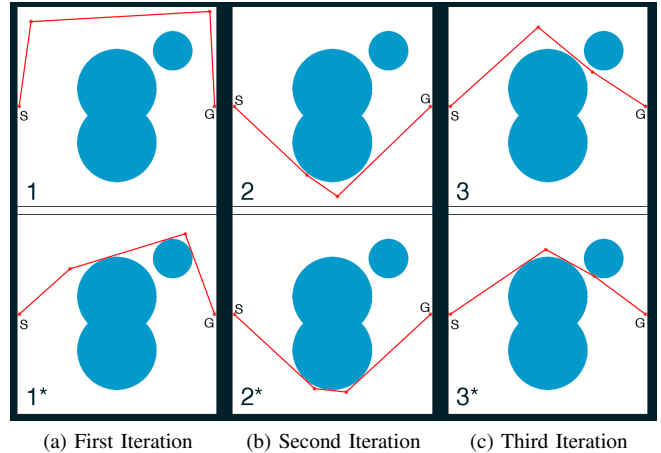


Fig. 1: Iterations of IOS-MP progress toward a globally optimal path. (a) The sampling-based motion planner finds the first path (top), which is then locally optimized (bottom). (b, c) In each subsequent iteration, the sampling-based motion planning step finds a path that is shorter than the previously found best path (top), and this new path is then locally optimized (bottom) before continuing to the next iteration.

of these algorithms allows them to explore every relevant homotopic class eventually. Many of these methods will also converge upon the globally optimal solution as the number of samples approaches infinity [10][6]. Unfortunately, the resulting paths may converge slowly, so in *finite* time, they frequently return paths that are far from locally optimal, especially in higher dimensional problems [8].

Optimization-based motion planning methods take a more microscopic approach. These methods quickly compute high quality paths by numerically optimizing an initial path, converging toward a local minimum [18][19][16][9]. However, the resulting path quality of such methods is directly tied to the path initialization. Due to the inherently non-convex nature of motion planning with obstacles, for some initializations the resulting paths may be far from globally optimal, or the optimization may not find a solution at all. This limitation can be partially circumvented through techniques such as restarting the optimization with multiple different initial paths (e.g., [19]), but such approaches are heuristically driven and typically provide no global guarantees. Additionally, by treating each restart as independent, they potentially ignore useful pre-computed information (such as collision-free configurations) that could be valuable if shared across restarts.

Our new method, Interleaved Optimization with Sampling-based Motion Planning (IOS-MP), alternates between local optimization and global exploration, sharing information between the two, effectively combining the best of both of these paradigms (see Fig. 1). IOS-MP starts with global exploration by building a graph using an asymptotically optimal planner, such as k -nearest Probabilistic Roadmap (PRM*) [10] or Batch Informed Trees (BIT*) [6], until it finds a collision free path (Fig. 1(a) top). IOS-MP then uses constrained local optimization based on an augmented Lagrangian method [20] to refine the path found by the sampling-based method (Fig. 1(a) bottom). The algorithm then iterates between (1) resuming the sampling-based motion planner until a new better path is found and (2) running constrained local optimization on this new path (see Fig. 1(b),(c)). The sampling-based motion planning phase of each iteration explores globally, discovering other homotopic classes in configuration space, as well as escaping local minima in the path cost landscape. The constrained local optimization phase in turn allows the method to provide a high quality local solution at the end of each iteration. For efficiency, information is shared by both of the phases at each iteration, with the sampling-based method seeding the local optimization with improving initial solutions, and the local optimization passing potentially valuable new vertices and edges to the sampling-based method.

IOS-MP is an *anytime* motion planning algorithm, in that the algorithm can be stopped at any time (after iteration 1) and return a locally optimized path, and running the algorithm for more time will return locally optimized paths that asymptotically approach a globally optimal path. The framework we introduce is generalizable; while our implementation uses augmented Lagrangian optimization and PRM* or BIT*, other local optimization algorithms and other sampling-based planning algorithms can be substituted in as long as the formulations and constraints described in this paper can be applied. The contribution of this paper is in describing and evaluating a framework for constructing a motion planning algorithm which leverages both local path optimization and global exploration. This allows it to provide higher quality paths earlier than asymptotically optimal sampling-based motion planning algorithms alone can provide, while providing the guarantees—namely completeness and asymptotic optimality—which are not provided by most optimization-based motion planning algorithms.

In this paper we optimize trajectories with respect to path length. We demonstrate IOS-MP’s efficacy in simulated environments of varying dimensionality and complexity as well as for a 7 degree of freedom (DOF) manipulator performing tasks defined by both goal configurations and workspace goal regions.

II. RELATED WORK

In sampling-based motion planning, a graph data structure is constructed incrementally via random sampling providing a collision-free tree or roadmap in the robot’s configuration

space. Early versions of these algorithms provide probabilistic completeness, i.e., the probability of finding a path, if one exists, approaches one as the number of samples approaches infinity. Classic examples include the Rapidly-exploring Random Tree (RRT) [13] and the Probabilistic Roadmap (PRM) [11] methods. Adaptations of these algorithms can provide asymptotic optimality guarantees, wherein the path cost (e.g., path length) will approach the global optimum as the number of algorithm iterations increases. Examples include RRT* and PRM* [10] where the underlying motion planning graph is either rewired or has asymptotically changing connection strategies. Other asymptotically optimal algorithms include Batch Informed Trees (BIT*) [6] in which samples are processed in batches and Fast Marching Trees (FMT*) [8] which grows a tree in cost-to-arrive space. Other work such as the cross-entropy method [12] has investigated the distributions from which samples and trajectories are taken.

Optimization-based motion planning algorithms perform numerical optimization in a high dimensional trajectory space. Each trajectory is typically represented by a vector of parameters representing a list of robot configurations or controls. A cost can be computed for each trajectory, and the motion planner’s goal is to compute a trajectory that minimizes cost. In the presence of obstacles and other constraints, the problem can be formulated and solved as a numerical optimization problem. CHOMP [18] takes an initial trajectory and performs gradient descent. Traj-Opt [19] uses sequential quadratic programming with inequality constraints to locally optimize trajectories. ITOMP combines optimization with re-planning to account for dynamic obstacles [16]. These methods typically produce high quality paths but are frequently unable to escape local minima, and as such are subject to initialization concerns. To avoid local minima, some methods inject randomness into the system [9], [3].

Several methods aim to bridge the gap between sampling-based methods and optimization. Some methods use paths generated by global planners and refine them using shortcutting or smoothing methods in post processing or adaptively [11][15][7][14][17]. Recent work has begun considering ways to integrate ideas from optimization-based motion planning with sampling-based motion planning. Gradient-RRT moves vertices to lower cost regions using gradient descent during the construction of an RRT [2]. More recently, RABIT* uses CHOMP to locally optimize in-collision edges during BIT* planning to bring them out of collision and effectively find narrow passages [4]. In contrast, our method is using the local optimizer not to find narrow passages, but to improve the overall quality of the paths found by the sampling-based planner, while relying on the sampling-based planner’s completeness property to discover narrow passages.

IOS-MP aims to more thoroughly integrate optimization with sampling-based motion planning by interleaving them to achieve fast motion planning with asymptotic optimality.

III. PROBLEM DEFINITION

Let C be the configuration space of the robot. Let $\mathbf{q} \in C$ represent a single robot configuration and $\mathbf{p} =$

$\{\mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_n\}$ be a continuous path in configuration space represented in a piecewise manner by a sequence of configurations. In this paper, we minimize the path with respect to the length, where we define $\text{length}(\mathbf{p})$ to be the sum of Euclidean distances in configuration space along the path. We use the sum of Euclidean distances because it is a commonly used metric for path length and because it satisfies the triangle inequality property required by asymptotically optimal sampling-based motion planners (unlike other metrics such as sum of squared distances). While we use path length for our optimization objective, certain other cost functions that are continuous, differentiable, and do not violate the triangle inequality could be used instead.

In the robot's workspace are obstacles that must be avoided. Each obstacle may be composed of a set of obstacle primitives, and let the set of primitives composing all obstacles be O . Let $C_{\text{free}} \subseteq C$ be the collision free subspace and $C_{\text{obs}} \subseteq C$ be the in-collision subspace of C based on obstacle primitives O . In the context of this work, we are considering static environments, such that C_{obs} is unchanging as a function of time. A path is collision-free if each edge $(\mathbf{q}_i, \mathbf{q}_{i+1}), i = 0, \dots, |\mathbf{p}| - 1$, does not intersect an obstacle primitive $o \in O$. Formally, we define $\text{clearance}(\mathbf{q}_i, \mathbf{q}_{i+1}, o)$ as the signed squared distance between the path edge $(\mathbf{q}_i, \mathbf{q}_{i+1})$ and obstacle primitive o (where negative values correspond to obstacle penetration distance); a collision-free path is a path for which each edge has non-negative clearance. This requires that the obstacle and robot representations allow for the computation, either analytically or numerically, of the signed distance between the robot and the obstacle primitive. This property holds for robots and obstacles represented using collections of bounding spheres or bounding capsules (volumes defined by a sphere swept along a straight line segment) as well as point clouds and other representations, which are common representations used in the literature and useful in practice.

The optimal motion planning problem then becomes finding a collision-free path from $\mathbf{q}_{\text{start}}$ to $\mathbf{q}_{\text{goal}} \in C_{\text{goal}}$ that minimizes cost, where C_{goal} is the (possibly singleton) set of goal configurations. This optimal motion planning problem can be formulated as a nonlinear, constrained optimization problem:

$$\begin{aligned} \mathbf{p}^* &= \underset{\mathbf{p}}{\text{argmin}} \text{length}(\mathbf{p}) \\ \text{Subject to:} \\ \text{clearance}(\mathbf{q}_i, \mathbf{q}_{i+1}, o) &\geq 0, & 0 \leq i < |\mathbf{p}|, \forall o \in O \\ \mathbf{g}(\mathbf{p}) &\geq 0, & \forall \mathbf{g} \in J \\ \mathbf{q}_0 &= \mathbf{q}_{\text{start}} \\ \mathbf{q}_{|\mathbf{p}|} &= \mathbf{q}_{\text{goal}} & \mathbf{q}_{\text{goal}} \in C_{\text{goal}} \end{aligned} \quad (1)$$

where J is a set of generalized inequality constraints that are specific to the problem and robot (e.g., to represent joint angle limits for a manipulator), and where \mathbf{p}^* is the optimal motion plan. Let K be the set of inequality constraints implied by O for obstacle avoidance. The set of all inequality constraints then becomes $I = J \cup K$. IOS-MP is an iterative algorithm for efficiently solving this problem such that the

solution asymptotically approaches \mathbf{p}^* .

IV. METHOD

IOS-MP integrates ideas from both sampling-based and optimization-based motion planning. The method interleaves path optimization steps with graph expansion steps to achieve fast convergence.

A. Overview

The top level algorithm for IOS-MP, Alg. 1, runs in an anytime manner, iterating as time allows and storing the best path found up to any time.

Algorithm 1: IOS-MP

Input: start configuration $\mathbf{q}_{\text{start}}$, obstacle set O , time limit t	
Output: motion plan \mathbf{p}^*	
1	$G \leftarrow (\{\mathbf{q}_{\text{start}}\}, \emptyset)$
2	Best cost $c \leftarrow \infty$
3	$\mathbf{p}^* \leftarrow \emptyset$
4	while time elapsed $\leq t$ do
5	$\mathbf{p} \leftarrow \text{graphExpansionStep}(O, G, c, t)$
6	$c \leftarrow \text{cost}(\mathbf{p})$
7	$\hat{\mathbf{p}} \leftarrow \text{optimizationStep}(\mathbf{p})$
8	$c \leftarrow \text{cost}(\hat{\mathbf{p}})$
9	$G \leftarrow G \cup \hat{\mathbf{p}}$
10	$\mathbf{p}^* \leftarrow \hat{\mathbf{p}}$
11	end

In the first step of each iteration, IOS-MP executes a sampling-based motion planner to expand the graph G until a new path is found that has cost lower than c . The sampling-based motion planner returns the new path \mathbf{p} and updates c . In the second step of each iteration, IOS-MP executes a local optimization method to locally optimize \mathbf{p} . The method saves the optimized path $\hat{\mathbf{p}}$ as the best new motion plan found. It also adds the configurations and segments defining $\hat{\mathbf{p}}$ as new vertices in G . The algorithm then iterates, returning to global exploration with the sampling-based motion planner, but seeded with vertices and edges generated by both the prior random sampling and the local optimization.

B. Global Exploration using Sampling-based Motion Planning

The global exploration step uses a sampling-based motion planner to expand a graph until a new path—spanning from the start configuration to any goal configuration—is found that is of lower cost than any previously found path. The sampling-based motion planner maintains a graph $G = (V, E)$, where V is a set of vertices which represent collision-free configurations of the robot and E is a set of edges, where an edge represents a collision-free transition between two robot configurations.

To expand the graph G , our method is designed to use an asymptotically optimal sampling based motion planner such as k -nearest Probabilistic Roadmap (PRM*) [10]. PRM* samples random configurations in the robot's configuration

space, locates their k nearest neighbors (where k changes as a function of the number of vertices in the graph), and attempts to connect the configurations to each of its neighbors (connection step). In each graph expansion step, PRM* executes until a path better than the current best is found, at which point the optimization step begins. We require that the algorithm randomly samples and attempts to connect at least one vertex in between consecutive optimization steps, to ensure that the global exploration continues. We refer to IOS-MP with k -nearest PRM* for its sampling-based planning step as IOS-MP (PRM*).

IOS-MP can alternatively be used with other asymptotically optimal motion planners, such as Batch Informed Trees (BIT*) [6]. BIT* operates by processing samples in batches. From a batch of samples it builds a tree using a graph search heuristic during tree construction until a solution is found or the tree can no longer be expanded. For the next batch, it limits its sampling to the subspace in which a solution of higher quality could be found. For IOS-MP with BIT*, the usual BIT* algorithm executes for a short amount of time without interruption (≤ 0.2 seconds) to take advantage of its batching properties. After the short execution, the best path found by BIT* is evaluated against the previous best, and if it has improved, it is optimized. We refer to IOS-MP with BIT* for its sampling-based planning step as IOS-MP (BIT*). In Sec. V, we show results for both when PRM* and when BIT* are used within IOS-MP.

C. Local Optimization using an Augmented Lagrangian Method

For the local optimization step, we use a nonlinear constrained optimization method called the augmented Lagrangian (AL) method to locally optimize the path. AL is similar to standard Lagrangian methods, in that it utilizes Lagrangian multipliers, but differs in that it adds additional quadratic constraint terms. AL is also similar to penalty-based optimization methods, but by introducing explicit Lagrange multiplier estimates at each step of the optimization, in practice it is able to reduce ill conditioning [20].

We apply AL to the nonlinear constrained optimization problem in (1), where the initial value for \mathbf{p} is the most recent path found by the sampling-based method. The AL method iteratively minimizes the augmented Lagrangian function, \mathcal{L}_A defined in (2) and (3), based on each constraint g_i in the set of constraints I , and an iterative approximation of the Lagrangian multipliers λ .

$$\mathcal{L}_A(\mathbf{p}, \boldsymbol{\lambda}^k, \mu_k) = f(\mathbf{p}) + \sum_{i \in I} \psi(g_i(\mathbf{p}), \lambda_i^k, \mu_k) . \quad (2)$$

$$\psi(\gamma, \sigma, \mu) = \begin{cases} -\sigma\gamma + \frac{1}{2\mu}\gamma^2 & \gamma - \mu\sigma \leq 0 \\ -\frac{\mu}{2}\sigma^2 & \text{otherwise} \end{cases} . \quad (3)$$

At the k^{th} step of the AL algorithm, outlined in Alg. 2, \mathcal{L}_A is minimized with respect to the trajectory \mathbf{p}_k . In our implementation we use gradient descent with line search for this minimization.

Algorithm 2: The Augmented Lagrangian method

<p>Input: $\mathbf{p}_0, \mu_0 > 0, \tau > 0, \boldsymbol{\lambda}_0$ Output: optimized path \mathbf{p}</p> <pre> 1 $k \leftarrow 0$ 2 while $\ \nabla_{\mathbf{p}} \mathcal{L}_A\ \geq \tau$ do 3 $\mathbf{p}_{k+1} := \text{minimize w.r.t. } \mathbf{p}_k, \mathcal{L}_A(\mathbf{p}_k, \boldsymbol{\lambda}, \mu)$; 4 Update $\boldsymbol{\lambda}$; 5 $\mu := \mu \cdot \mu_{up}$, where $\mu_{up} \in (0, 1)$; 6 $k \leftarrow k + 1$; 7 end 8 Return \mathbf{p}_{k-1};</pre>

After the minimization, for each inequality constraint $i \in I$, its associated Lagrangian multiplier λ_i is then updated using the following formula.

$$\lambda_i^{k+1} = \max\left(\lambda_i^k - \frac{g_i(\mathbf{p}_k)}{\mu_k}, 0\right) \quad (4)$$

Penalty multiplier μ is then updated by multiplying it by a parameter $\mu_{up} : 0 < \mu_{up} < 1$. The algorithm then updates the augmented Lagrangian function and iteration continues until convergence of \mathbf{p} .

For IOS-MP, $f(\mathbf{p})$ is path length, and the set of inequality constraints I in (1) is the union of the problem-specific inequality constraints J and the obstacle avoidance constraints. For obstacle avoidance, the set of inequality constraints K consists of a constraint per obstacle primitive per pair of adjacent configurations in \mathbf{p} . For a path \mathbf{p} consisting of k configurations, there will be $(k-1) \cdot |O|$ inequality constraints related to obstacle avoidance, as we require the edge between configurations to be collision free. So for constraint g_i , associated with obstacle primitive o and configurations \mathbf{q}_k and \mathbf{q}_{k+1} , $g_i(\mathbf{p}) = \text{clearance}(\mathbf{q}_k, \mathbf{q}_{k+1}, o)$.

In the case of motion planning from one configuration to another configuration, the AL method does not update these two configurations but rather updates only the configurations between the start and goal. In this way, the equality constraints $\mathbf{q}_0 = \mathbf{q}_{\text{start}}$ and $\mathbf{q}_{|\mathbf{p}|} = \mathbf{q}_{\text{goal}}$ are trivially enforced. In the case of goal regions, an additional inequality constraint is added to guarantee that the goal configuration remains within the goal region. Also in practice, to guarantee timely exit, we enforce maximum iteration counts for both the internal minimization and the outside AL loop.

D. Analysis

We show that IOS-MP (PRM*), a variant of IOS-MP that uses k -nearest PRM* for its sampling-based planning step, is asymptotically optimal. To ensure that paths computed asymptotically approach the globally optimal path, we must ensure that adding vertices to the graph based on the optimization step does not affect the asymptotic optimality properties of the sampling-based motion planner. We do this by making a distinction between vertices added by the sampling-based method and vertices added by the optimization. We only count vertices added by the sampling-based

method toward the k -nearest neighbor count when adding edges. Because of this distinction, we can show that IOS-MP (PRM*) produces a valid supergraph of what PRM* would have produced, were the optimization steps of IOS-MP (PRM*) not performed. The behavior, as time progresses, of IOS-MP (PRM*) can be conceptualized as the behavior of PRM* with *finite* time interruptions occurring periodically during which the optimization steps are performed. To ensure that the asymptotic optimality of PRM* is retained by IOS-MP (PRM*), the following assumptions must hold:

Assumption 1: The optimization step is guaranteed to exist in finite time.

Assumption 2: The order and configurations randomly sampled by the graph construction algorithm are unchanged from what they would be were there no optimization steps occurring.

Theorem 1: For any finite running time τ , by which an unmodified k -nearest PRM* motion planning algorithm would have produced graph G_u , there exists a finite running time $\hat{\tau}$, by which IOS-MP (PRM*) produces a graph G_s which is a valid supergraph of G_u .

Proof: Consider the ordered list $V_u = \{v_0, v_1, \dots, v_k\}$ where V_u represents the order in which vertices are sampled during the construction of graph $G_u = (V_u, E_u)$ up to time τ . Also consider the ordered list V_u^i as the sub list of V_u at iteration i . Additionally, consider the set of edges E_u^i as the subset of E_u at iteration i . As we are only considering static environments, E_u^i is dependent only on the order and presence of each vertex $v \in V_u^i$ at the time of connection. That is to say that the edges are not affected by the specific *time* at which a vertex was added, but rather only on the existence of the set of vertices in the graph when the connection is attempted.

Now consider IOS-MP (PRM*)'s construction of graph $G_s = (V_s, E_s)$, and specifically the ordered construction of V_s^i and E_s^i , as above. By assumption 2, each vertex v_s^i will be identical in order and value, up to its counterpart vertex v_u^i in V_u^i . Also, as a result of the connection strategy detailed above that treats optimization-added vertices as distinct when computing k -nearest neighbors, E_s^i will contain all of the edges present in E_u^i .

Next, consider the sequence of steps taken by the sampling-based algorithm alone during construction up to iteration i . An example of such a sequence could be: sample vertex 0, sample vertex 1, connection step 0, sample vertex 2, sample vertex 3, connection step 1, \dots , sample vertex i . Note that the frequency of a connection step is dependent on the specifics of the implementation but happens no more than once per sample.

The corresponding sequence of steps taken by IOS-MP (PRM*) will look very similar, for example: sample vertex 0, sample vertex 1, connection step 0, optimization step 0, sample vertex 2, sample vertex 3, connection step 1, optimization step 1, \dots , sample vertex i . Note also that we require there to be at most one optimization step present between consecutive connection steps (see Sect. IV-B). Also, by the connection strategies detailed above, the set of vertices

and edges after an optimization step is a superset of the set of vertices and edges prior to the optimization step, and because we only add the vertices and edges back if they compose a valid path, it is a valid graph.

Let the time required to sample vertex k be τ_s^k , and the time required for connection step j be τ_c^j . The time required to construct the graph up to i sampled vertices, τ_g^i is then:

$$\tau_g^i = \sum_{n=0}^i \tau_s^n + \sum_{m=0}^j \tau_c^m \text{ where } j \leq i . \quad (5)$$

Now consider the time required to complete the k^{th} optimization step, $\hat{\tau}_o^k$. This time will be finite by assumption 1. The aggregate time then, to perform IOS-MP (PRM*) up to sampled vertex i then becomes:

$$\hat{\tau}_g^i = \sum_{n=0}^i \tau_s^n + \sum_{m=0}^j \tau_c^m + \sum_{p=0}^k \hat{\tau}_o^p \text{ where } k \leq j \leq i . \quad (6)$$

As such, the aggregate time required of IOS-MP (PRM*) up to sampling iteration i will be a finite sum of finite times, and subsequently finite itself.

Therefore, there exists a finite time $\hat{\tau}$, where IOS-MP (PRM*)'s underlying sampling of vertices during construction of V_s as a superset of V_u where the vertices which are present in V_u were added to V_s in the same order as in V_u , and the connections E_s were constructed as a superset of the edges E_u . This results in a graph G_s which is a valid supergraph of G_u produced in finite time. ■

For any finite time, IOS-MP (PRM*) will produce, in a longer finite time, a supergraph of the planner which would have been produced by PRM* alone, by Thm. 1. As such, any path that would have been present in PRM*'s graph will also eventually be present in IOS-MP (PRM*)'s graph in finite time. IOS-MP (PRM*) will therefore produce, in finite time, a path that is equal or better than any path that would be produced by the PRM* alone in finite time. It then follows naturally that as PRM* is both probabilistically complete and asymptotically optimal [10], IOS-MP (PRM*) will also be probabilistically complete and asymptotically optimal.

It should be noted that the supergraph assumption does not necessarily hold for IOS-MP (BIT*) in our implementation, and as such we cannot claim that IOS-MP (BIT*) is asymptotically optimal as a result of this proof, but we note that in most cases it works comparably or better than IOS-MP (PRM*) as shown in the next section.

V. RESULTS

We evaluate the performance of IOS-MP with respect to configuration space dimensionality and varying environmental complexity, as well as demonstrate performance on simulated 7-DOF manipulation tasks based on the Fetch robot [1]. We evaluated two versions of IOS-MP: (1) IOS-MP (PRM*), which uses PRM* as the underlying sampling-based planner, and (2) IOS-MP (BIT*), which uses BIT* as the underlying sampling-based planner. We used implementations of PRM* and BIT* from the Open Motion Planning Library (OMPL) [5] for the sampling-based motion planning

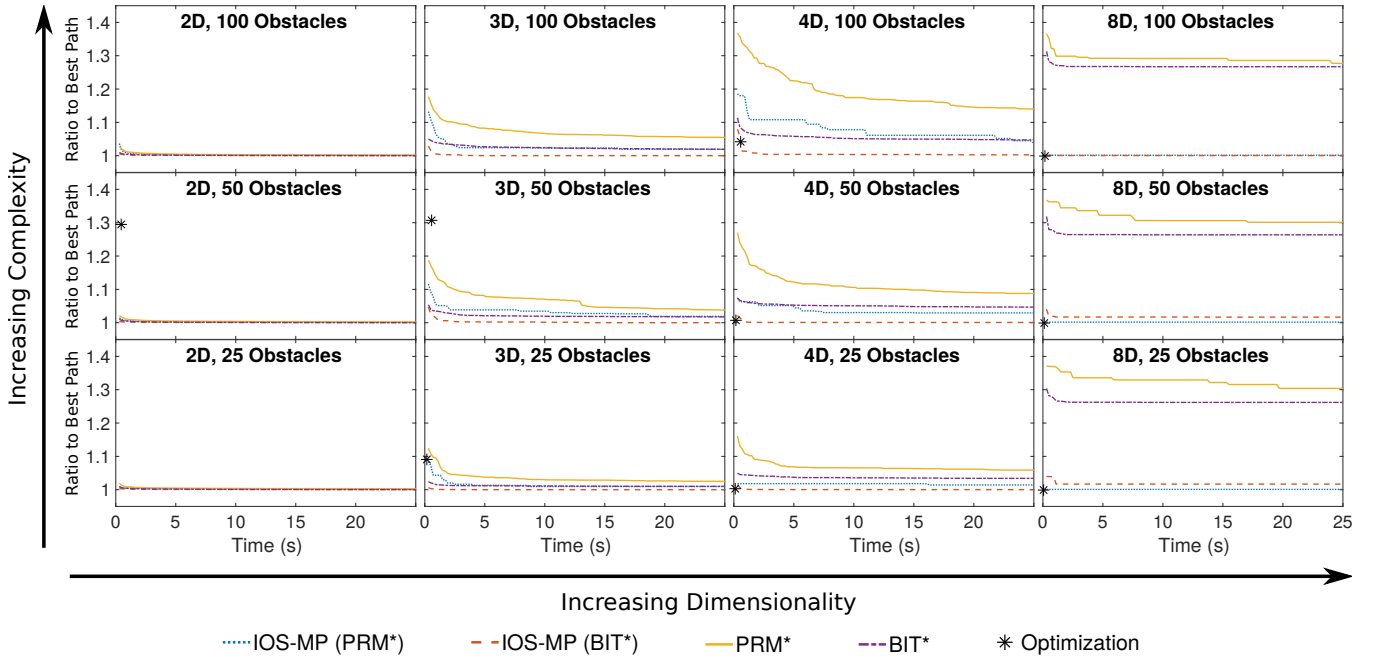


Fig. 2: Comparison of methods in randomized environments of varying complexity and dimensionality. In the cases where the asterisk denoting the results from the local optimization is not shown, the value is above the upper end of the plot, between 1.45 and 1.55

step of IOS-MP. We compared our two variants of IOS-MP with methods based solely on sampling-based motion planning (PRM* and BIT* as implemented in OMPL) or based solely on optimization (an augmented Lagrangian constrained optimization). All experiments were performed on a system with an 8 core 3.2GHz Intel Xeon E5-2667 processor.

A. Dimensionality and Complexity Scaling

To evaluate how the method scales with respect to both configuration space dimensionality and environmental complexity, we evaluate IOS-MP for a point robot in a unit box of dimension d . We varied d from 2 to 8 and varied the number of obstacle primitives from as few as 25 to as many as 100. In each environment of dimension d we randomly placed obstacles in the form of hyperspheres of dimension d , i.e., circles for 2D, spheres for 3D, and hyperspheres for 4D and 8D. The radii of each obstacle was a random number sampled from a uniform distribution in the range $[0.05, 0.2]$. We defined the start and goal configurations for the point robot as points at the centers of opposite faces of the box. In the optimization formulation for (1), we used inequality constraints for clearance from the hypersphere obstacles. An example environment for 2D is shown in Fig. 1.

We show the relative performance of IOS-MP (PRM*), IOS-MP (BIT*), PRM*, BIT*, and Optimization (based on a local augmented Lagrangian method) for scenarios of different dimensionality and environment complexity in Fig. 2. In each plot, we ran the IOS-MP variants and the sampling-based methods in an anytime manner for 25 seconds, considering the best solution found over time and averaging over

runs in 15 environments. For Optimization, we initialized the algorithm with a straight line path in configuration space from start to goal and a fixed number of trajectory points (20) and ran the algorithm until convergence for each of the 15 environments. Optimization does not always succeed in finding a solution (it was successful in 86% of the point robot scenarios represented here); because of this, we computed averages only across runs for which a collision-free solution was found and plot a * for this average. To display meaningful averages across the 15 environments for each graph, we defined the “best solution” for a particular environment as the shortest path found by any method at any time for that environment, and then averaged over ratios with respect to the best solution in each environment.

In Fig. 2, both IOS-MP (BIT*) and IOS-MP (PRM*) outperform their sampling only counterparts in every case, and outperform optimization-based motion planning in most cases, with IOS-MP (BIT*) being the best all around method. The results show that the sampling-based methods PRM* and BIT* perform relatively well in the lower dimensionality cases, especially where the environment is complex (i.e. many obstacle primitives), while local optimization performs increasingly well as the dimensionality increases. By interleaving sampling-based methods and local optimization, IOS-MP gains the speed of both of these methods on these different types of problems. When looking at the trend from left to right of Fig. 2, dimensionality increases, and the percent of configuration space that is free also rises because the number and average radius of the obstacles is held constant. Because local optimization is relatively more efficient at reducing the length of a path in higher

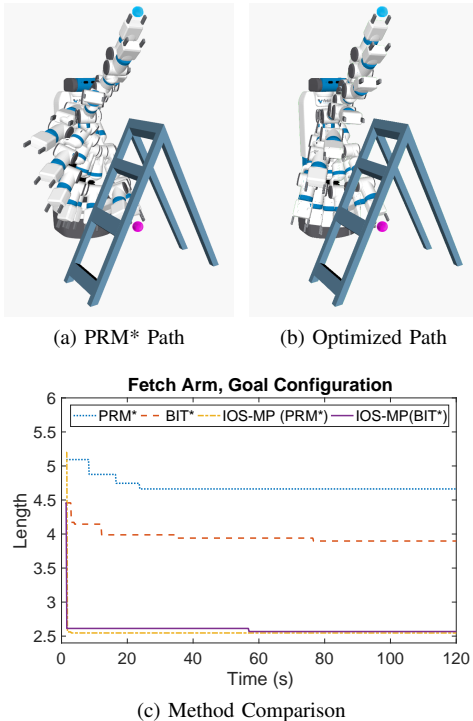


Fig. 3: (a-b) Visualization of the first Fetch robot arm manipulator task, start configuration in pink and goal configuration in blue, (a) with a path found by PRM* (b) and its optimized equivalent. (c) A comparison of the performance of the 4 methods for the Fetch task.

dimensional configuration spaces that are more sparse, IOS-MP is particularly effective for these higher dimensional problems compared to the motion planners that only use sampling-based methods.

B. 7-DOF Manipulation Tasks

We demonstrate IOS-MP’s efficacy in various 7-DOF manipulation tasks with the arm of a Fetch robot.

1) *Motion Planning to a Goal Configuration:* In the first scenario (Fig. 3), the Fetch robot must plan a motion from its start configuration, where its arm is beneath a ladder to grasp a tool, to a goal configuration, where its arm is above the ladder, as if handing off the tool to a person above. The ladder and robot links were represented by 8 capsule primitives. In the optimization formulation for (1), we used inequality constraints for obstacles avoidance, to represent joint limits, and to guarantee the robot does not self-collide.

Fig. 3c shows the results of applying PRM*, BIT*, IOS-MP (PRM*), and IOS-MP (BIT*) to this motion planning problem. The methods were run for 120 seconds, recording the path length based on Euclidean distance in configuration space of the best path found up to that time. Additionally, we evaluated augmented Lagrangian local optimization, initialized with a 20-point straight line path in configuration space, but this method did not converge to a valid collision-free solution.

In this scenario, both IOS-MP (PRM*) and IOS-MP (BIT*) drastically outperform their sampling-only counterparts. In just a couple of iterations, the loop of sampling-based methods and local optimization produces better plans than other methods produce in 120 seconds. The high dimensionality of the problem means that the optimization step of IOS-MP can provide a large benefit in a short time. Fig. 3 illustrates (a) a path found during the graph expansion step of IOS-MP (PRM*), and (b) the path after the optimization step of the same iteration of IOS-MP. The optimization step of IOS-MP significantly reduces extraneous motion, and iterating with both a sampling-based step and optimization step allows for fast improvement toward a globally optimal solution.

2) *Motion Planning to a Workspace Goal Region:* In the second scenario, the Fetch robot is placed in a more cluttered scene (see Fig.4) and must plan a path starting from a start configuration, in which it is reaching up as if to grab a tool from a person on the ladder, and moving such that its end-effector enters a goal *region* in the workspace under the ladder. We randomly perturb the location of the workspace goal region under the ladder and average the results of 80 runs. The ladder, robot, and other obstacle geometries are being represented in this scene using 16 capsule obstacle primitives. In this scenario, the goal is not a specific configuration, but a potentially infinite number of configurations, all configurations in which the end effector is within a workspace region. In fact, in configuration space the goal region may be both non-convex and disconnected. To solve this motion planning problem, the robot must consider many possible paths across multiple homotopic classes that may pass between different rungs or around the side of the ladder.

During execution, IOS-MP (PRM*) incrementally discovers and explores the goal region as the PRM* graph is constructed. As configurations which lay within the goal region are sampled, they are added to the set of goal configurations to which the planner is attempting to connect to from the start configuration. In the optimization formulation for this scenario, we add an additional inequality constraint, the end effector’s proximity to the workspace goal region, which requires the end effector to remain within the workspace goal region during optimization, however the goal configuration is allowed to vary during the optimization as long as the constraint is respected at convergence. Due to goal region restrictions in the OMPL 1.2.1 BIT* implementation, we only evaluate IOS-MP (PRM*) and PRM* for this scenario. To evaluate the efficacy of sharing information from the optimizer with the roadmap we show results for which the optimized paths from IOS-MP are added back into the roadmap and results for which they are not. To average multiple runs we show results beginning at the instant the first path was found.

Fig. 4(a-b) depicts paths which exist in two separate homotopic classes, demonstrating the effectiveness of the global nature of the sampling-based planning while still benefiting from the optimization of the paths found. Fig. 4(c) also

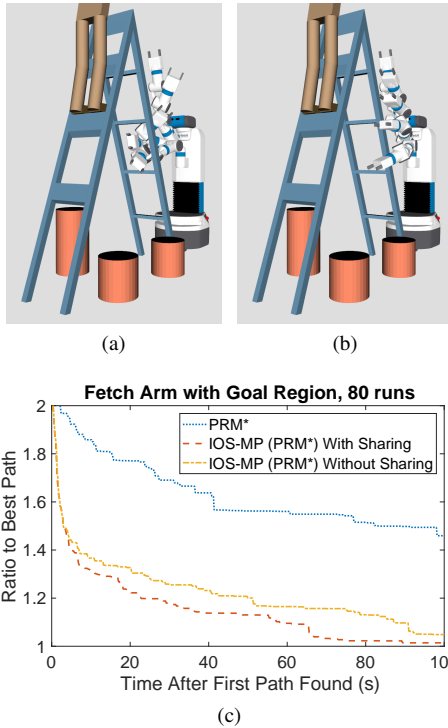


Fig. 4: (a-b) A task in which the Fetch robot must move its end effector from above its head to a region underneath the ladder. (a) A path found during an early iteration of IOS-MP in a non-ideal homotopic class between ladder rungs. (b) A path in a better homotopic class found in a later iteration of IOS-MP. (c) A comparison of the performance of IOS-MP (PRM*), with and without optimized path sharing, and PRM*. We average results over 80 runs in which the workspace goal region was randomly perturbed under the ladder.

shows that IOS-MP (PRM*) performs very well compared to the sampling-based planner alone, and demonstrates the substantial benefit of adding the optimized path back into the roadmap.

VI. CONCLUSION

In this paper, we present a method designed to achieve the best of both local optimization and global sampling-based motion planning. Through interleaving local optimization with global exploration, and sharing valuable information between the two, our method works in an anytime fashion, providing locally optimized solutions as of the most recent optimization step, while still providing global asymptotic optimality. We demonstrate that our method, integrated with both PRM* and BIT*, performs well compared to local optimization or sampling-based methods alone in experiments of varying environmental complexity and dimensionality. In the future, we plan to investigate integrating IOS-MP with other optimization methods, such as sequential quadratic programming, using cost functions other than path length, and using different obstacle primitives.

REFERENCES

- [1] <http://fetchrobotics.com/research/>.
- [2] Dmitry Berenson, Thierry Simeon, and Siddhartha S Srinivasa. Addressing cost-space chasms in manipulation planning. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 4561–4568, May 2011.
- [3] W.F. Carriker, P.K. Khosla, and B.H. Krogh. The use of simulated annealing to solve the mobile manipulator path planning problem. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 204–209, 1990.
- [4] Sanjiban Choudhury, Jonathan D Gammell, Timothy D Barfoot, Siddhartha S Srinivasa, and Sebastian Scherer. Regionally Accelerated Batch Informed Trees (RABIT*): A Framework to Integrate Local Information into Optimal Path Planning. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 4207–4214, Stockholm, Sweden, May 2016.
- [5] Ioan A. Şucan, Mark Moll, and Lydia E. Kavraki. The Open Motion Planning Library. *IEEE Robotics and Automation Magazine*, 19(4):72–82, December 2012.
- [6] Jonathan D Gammell, Siddhartha S Srinivasa, and Timothy D Barfoot. Batch informed trees (BIT*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 3067–3074. IEEE, 2015.
- [7] Kris Hauser and Victor Ng-Thow-Hing. Fast smoothing of manipulator trajectories using optimal bounded-acceleration shortcuts. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 2493–2498, May 2010.
- [8] Lucas Janson, Edward Schmerling, Ashley Clark, and Marco Pavone. Fast Marching Trees: a Fast Marching Sampling-Based Method for Optimal Motion Planning in Many Dimensions. *International Journal of Robotics Research*, 2015.
- [9] Mrinal Kalakrishnan, Sachin Chitta, Evangelos Theodorou, Peter Pastor, and Stefan Schaal. STOMP: Stochastic trajectory optimization for motion planning. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 4569–4574, May 2011.
- [10] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *Int. J. Robotics Research*, 30(7):846–894, June 2011.
- [11] L E Kavraki, P Svestka, J.-C. Latombe, and M Overmars. Probabilistic roadmaps for path planning in high dimensional configuration spaces. *IEEE Trans. Robotics and Automation*, 12(4):566–580, August 1996.
- [12] Marin Kobilarov. Cross-Entropy Motion Planning. *Int. J. Robotics Research*, 31(7):855–871, jun 2012.
- [13] Steven M LaValle and James J. Kuffner. Randomized kinodynamic planning. *Int. J. Robotics Research*, 20(5):378–400, May 2001.
- [14] Ryan Luna, Ioan A. Şucan, Mark Moll, and Lydia E Kavraki. Anytime solution optimization for sampling-based motion planning. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 5068–5074, May 2013.
- [15] Jia Pan, Liangjun Zhang, and Dinesh Manocha. Collision-free and smooth trajectory computation in cluttered environments. *Int. J. Robotics Research*, 31(10):1155–1175, September 2012.
- [16] Chonhyon Park, Jia Pan, and Dinesh Manocha. ITOMP: Incremental trajectory optimization for real-time replanning in dynamic environments. In *Int. Conf. Automated Planning and Scheduling (ICAPS)*, 2012.
- [17] Sean Quinlan and Oussama Khatib. Elastic Bands: Connecting Path Planning and Control. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 802–807, 1993.
- [18] Nathan D. Ratliff, Matt Zucker, J. Andrew Bagnell, and Siddhartha Srinivasa. CHOMP: Gradient optimization techniques for efficient motion planning. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 489–494, May 2009.
- [19] John Schulman, Jonathan Ho, Alex Lee, Ibrahim Awwal, Henry Bradlow, and Pieter Abbeel. Finding locally optimal, collision-free trajectories with sequential convex optimization. In *Robotics: Science and Systems (RSS)*, June 2013.
- [20] Stephen Wright and Jorge Nocedal. *Numerical Optimization*. Springer Science, 1999.