Footstep and Motion Planning in Semi-unstructured Environments Using Randomized Possibility Graphs

Michael X. Grey[1] Aaron D. Ames[2] C. Karen Liu[3]

Abstract— Traversing environments with arbitrary obstacles poses significant challenges for bipedal robots. In some cases, whole body motions may be necessary to maneuver around an obstacle, but most existing footstep planners can only select from a discrete set of predetermined footstep actions; they are unable to utilize the continuum of whole body motion that is truly available to the robot platform. Existing motion planners that can utilize whole body motion tend to struggle with the complexity of large-scale problems. We introduce a planning method, called the "Randomized Possibility Graph", which uses high-level approximations of constraint manifolds to rapidly explore the "possibility" of actions, thereby allowing lower-level motion planners to be utilized more efficiently. We demonstrate simulations of the method working in a variety of semi-unstructured environments. In this context, "semi-unstructured" means the walkable terrain is flat and even, but there are arbitrary 3D obstacles throughout the environment which may need to be stepped over or maneuvered around using whole body motions.

I. INTRODUCTION

As humanoid robotics technology continues to advance, there is growing interest in using legged platforms for disaster relief in hazardous environments where wheeled platforms may be unable to traverse. These hazardous environments may include structures like nuclear reactors or collapsing buildings. We refer to these environments as "semi-unstructured", because they were originally designed to be easily navigable but may have succumbed to conditions where intended routes are no longer clear of obstructions. Perhaps the passages have been structurally damaged or are being littered by various obstacles that have been left behind. For a legged robot to be useful under these conditions, it may need to use strategic foot placement and utilize the entire range of motion of its body, such as maneuvering between the bars in Fig. 1a.

Footstep planning is a cornerstone of humanoid robotics research due to its importance in navigating challenging terrain. Many algorithms have been designed for planning footsteps, but most tend to fall into one of four categories:

1) Planners that use bounding boxes for navigation
2) Planners that choose from a discrete set of actions
3) Planners that use optimization methods

[1]Michael X. Grey is a doctoral candidate in the School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA 30332 mxgrey@gatech.edu

[2]Aaron D. Ames is a Bren Professor in Mechanical and Civil Engineering & Control and Dynamical Systems, California Institute of Technology, Pasadena, CA 91125 ames@caltech.edu

[3]C. Karen Liu is an Associate Professor in the School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA 30332 karenliu@cc.gatech.edu
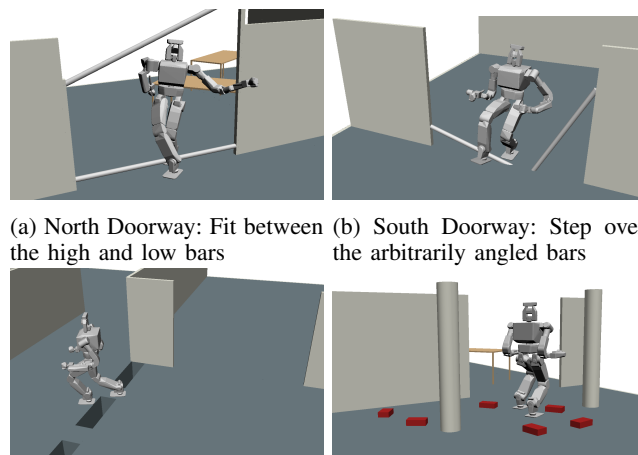
(a) North Doorway: Fit between the high and low bars

(b) South Doorway: Step over the arbitrarily angled bars

(c) West Doorway: Step across the gaps without falling in

(d) East Doorway: Step over the arbitrarily laid out bricks

Fig. 1: Doorways from the "Four Routes Scenario". Each passage has unique challenges that the robot must overcome.

4) Planners that find paths through and between discretized sets of contacts

Each of these categories has a limited scope of applicability. An example of the first category is [1] where the lower body is given a bounding box that encapsulates all the potential leg motion that the humanoid agent might exhibit. The upper body is left unbounded with the assumption that any upper body collisions will be resolved in a second stage. As a result, the humanoid is unable to realize when it could step over a small obstacle. By contrast, in [2], each foot is given an independent bounding box, and then the remainder of the robot's geometry is given one large bounding box which is elevated off the ground. All of the robot's motion takes place within these three bounding boxes, but having the bulk of the bounding geometry elevated allows the robot to navigate over small obstacles. However, the robot is unable to maneuver its upper body around overhanging obstacles.

Examples of work belonging to the second category include [3]–[7]. In all of these methods, there is a discrete set of predetermined actions that are available to the planner. They often use A* search where the predetermined actions are used for branching. These methods tend to be effective at stepping over and around short obstacles or gaps, but they generally are not able to alter their upper body configurations to negotiate overhanging obstacles or tight passages.

A variety of optimization methods have been applied to the problem of footstep planning and motion generation. In

[8] the footstep planning component was done with a human-in-the-loop, and then a motion trajectory was optimized over those footstep locations. The human operator is taken out of the loop in [9] and proofs for optimality and completeness are available, but this comes at the cost of requiring convex collision-free regions for the robot to pass through. Optimization can also be used to generate dynamically stable periodic walking gaits and controllers, such as methods proposed in [10], [11], although these do not address global search problems.

The fourth category is the only one where probabilistically complete whole body motion planning is available. For this category, a mode is defined by a set of contacts between the robot and the environment. These contacts determine the feasibility constraints for any configurations associated with that mode. Methods in this category are usually used to navigate extremely rough terrain or plan whole body manipulation. As an example, Multi-modal PRM [12], Random-MMP [13], and related methods [14], [15] first sample modes in the environment and then generate whole body motions to transition through and between those modes. A configuration that transitions between two modes must satisfy the feasibility constraints of both modes simultaneously. Random-MMP is theoretically capable of navigating the semi-unstructured environments of this paper, but it struggles to find narrow passages, which may be important for navigating through challenging environments.

In this paper, we present the "Randomized Possibility Graph" (RPG) for efficient whole body motion planning in semi-unstructured environments. By exploring the *possibility* of actions within a continuous search space, we can rapidly expand the search without being bottlenecked by expensive motion planning queries. We find that the RPG is effective at identifying passages which would be considered narrow by Random-MMP. We show that information from the RPG can be used to guide Random-MMP through narrow passages by focusing its sampling behavior. Together, these algorithms are able to traverse semi-unstructured environments on the order of seconds to minutes, depending on the complexity of the environment.

II. RANDOMIZED POSSIBILITY GRAPH

Motion planning methods ordinarily operate by constructing graphs or trees which consist of configurations that fully exist within the feasibility constraint manifold of the action they are performing. Remaining within this manifold is a reasonable requirement to place on the graph, because any vertices or edges which lie outside of the manifold are, by definition, invalid—which may mean it is physically impossible, or simply harmful to the robot or its surroundings. Unfortunately, for a humanoid robot to remain on the constraint manifold, expensive calls to a whole body inverse kinematics (IK) solver must be performed [16]–[18]. This results in a critical bottleneck if a broad area needs to be explored before finding a solution.

The key idea of this work is to explore the *possibility* of an action first, instead of immediately committing to
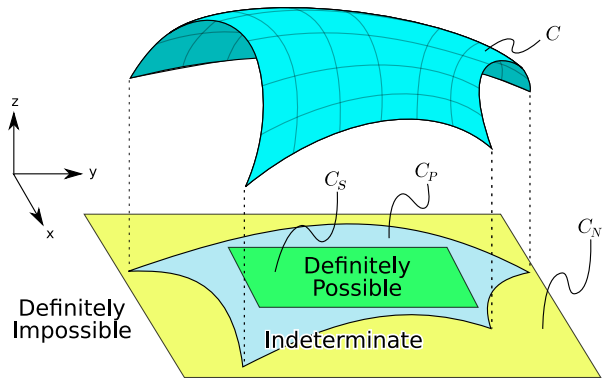

Fig. 2: Visual depiction of an abstract constraint manifold, C and its projection. The manifold is projected, C_P, from 3D space onto a plane. "Sufficient" C_S and "Necessary" C_N boundaries are fitted within and around the projection of the manifold. Elements inside the green box are definitely "Possible". Elements outside of the yellow box are definitely "Impossible". Elements inside the yellow box but outside of the green box are "Indeterminate" because they might or might not lie on the projection. Identifying whether a point lies inside or outside of C_S or C_N may be considerably faster than identifying whether it lies inside or outside of C_P.

costly whole body inverse kinematics queries. Therefore, the problem is broken down into two stages: The exploration of possibilities $\mathcal{P}$, and the generation of motions $\mathcal{M}$. The high-level graph generated during $\mathcal{P}$ will guide the efforts of the low-level planners used by $\mathcal{M}$.

The governing logical principles behind the RPG have a theoretical grounding in Possibility Theory [19], but the concepts are intuitive enough that a knowledge of Possibility Theory is not necessary to proceed. It is enough to understand that the *possibility* of any given action e can be labelled with "impossible", "possible", or "indeterminate" depending on whether it satisfies the necessary conditions (C_N) or the sufficient conditions (C_S) that are assigned to it:

$$e.\text{label} = \begin{cases} \text{"impossible"} & \text{if } C_N(e) \text{ is false} \\ \text{"possible"} & \text{if } C_S(e) \text{ is true} \\ \text{"indeterminate"} & \text{otherwise} \end{cases} \quad (1)$$

If we design necessary and sufficient conditions that can be checked much more quickly than querying the original constraint manifold, we can then construct a *Randomized Possibility Graph*, whose vertices are connected by either "possible" or "indeterminate" edges, and expand it very efficiently for whole body motion planning.

A. Simplifying the Manifold: Sufficient vs. Necessary

To construct the RPG, we must first design sufficient and necessary conditions for the feasibility constraint manifold of the action whose possibilities we are exploring. We should design the conditions to be tested quickly in order to reduce the computational cost of exploration. The conditions should also use as few parameters as possible, because randomized sampling methods are more effective in low-dimensional search spaces.

Suppose we have a 2D constraint manifold, C, which exists in a 3D state space (Fig. 2). Let the xy-plane be a low-dimensional feature space which contains the essential information for navigating C. We denote the projection of C by C_P. Even with a flattened-out projection, identifying which points are inside or outside of the manifold may still be costly or difficult, because the boundary of C_P may consist of functions that are expensive to compute or hard to fully define. However, suppose a box, circle, or some other simple shape can be fit within C_P such that it is *guaranteed* that every point within the simple shape also lies within the manifold projection. Such a shape would be a suitable representation of the sufficient condition manifold, C_S. Any point lying inside of C_S also lies inside of C_P and should be labelled with "possible". Similarly, if C_P can be bounded by a simple shape, C_N, such that $C_P \subseteq C_N$, then C_N would qualify as the necessary condition manifold. Any point lying outside of C_N should be labelled with "impossible". Finally, any point inside of C_N but outside of C_S should be labelled with "indeterminate".

B. RPG Construction

In the previous section, we introduced the concepts of C_N and C_S, the necessary and sufficient (respectively) condition manifolds which occupy a lower dimensional space than the state space. We define $\mathscr{E} = P(X)$ to be the "possibility exploration space" where X is the state space of the robot, and in this paper $P : X \to \mathrm{SE}(3)$ maps from the robot's state to the SE(3) transformation of the robot's root link. In general, P is chosen to be a projection operator such that $C_S \subseteq C_N \subseteq P(X)$. We will use $\mathscr{E}$ as the search domain for the possibility exploration stage $\mathcal{P}$.

Definition 1: A Randomized Possibility Graph is a tuple

$$RPG = (\Gamma_{\mathcal{P}}, \Phi_{\mathcal{M}}, \Omega_{\mathcal{M}}, \Gamma_{\mathcal{M}})$$

where,

- $\Gamma_{\mathcal{P}} = (V_{\mathcal{P}}, E_{\mathcal{P}})$ is a graph where $V_{\mathcal{P}}$ is a set of vertices which are elements of $\mathscr{E}$, and $E_{\mathcal{P}}$ is a set of directed edges, each with a "possible" or "indeterminate" label,
- $\Phi_{\mathcal{M}} : E_{\mathcal{P}} \times E_{\mathcal{P}} \to X^k$ is an operator which takes two edges and produces a set of $k > 0$ states,
- $\Omega_{\mathcal{M}} : X \times E_{\mathcal{P}} \times X \to X^f$ is an operator which maps two states with a possibility edge in between them into a discretized trajectory of $f \geqslant 0$ states, where $f = 0$ implies failure to find a trajectory,
- $\Gamma_{\mathcal{M}} = (V_{\mathcal{M}}, E_{\mathcal{M}})$ is a graph where $V_{\mathcal{M}}$ is a set of vertices which are elements of X, and $E_{\mathcal{M}}$ is a set of directed edges which indicate feasible paths between the vertices of $V_{\mathcal{M}}$.

The graph $\Gamma_{\mathcal{P}}$ is used to solve the first stage, $\mathcal{P}$, which explores possibilities. $\Gamma_{\mathcal{P}}$ can be constructed using a sample-based motion planner, such as PRM [20] or RRT [21]. If C_N or C_S have a small volume within $\mathscr{E}$, then a projection-based sampler may be needed, such as CBiRRT [22] where C_N and C_S are treated as task constraints. Alternatively, C_N and C_S can be treated as "hard" and "soft" task constraints respectively (meaning that C_N is required but C_S is preferred)
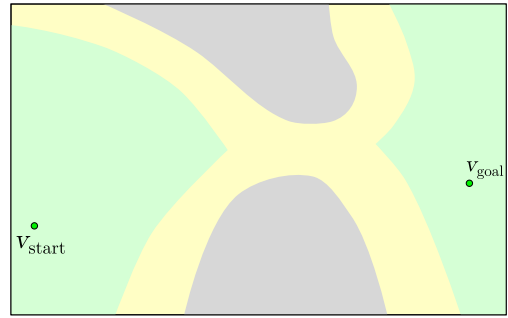

(a) The RPG is initialized with just a start vertex and goal vertex. The pale green regions are areas where C_S is satisfied, pale yellow is where C_N is satisfied, and gray is where C_N is violated. These colored regions represent information which is not directly known to the algorithm, so it must be searched via randomized sampling.
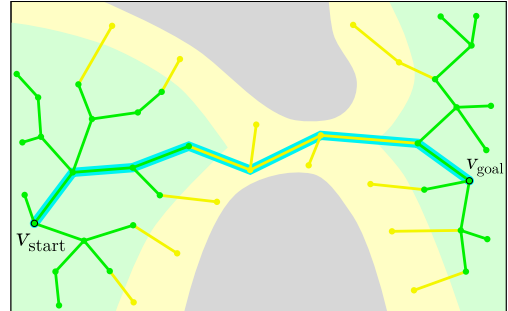
(b) A randomized sampling motion planner has constructed $\Gamma_{\mathcal{P}}$ to find a path from the start to the goal through $\mathscr{E}$. This provides a guide route, which is highlighted in cyan.
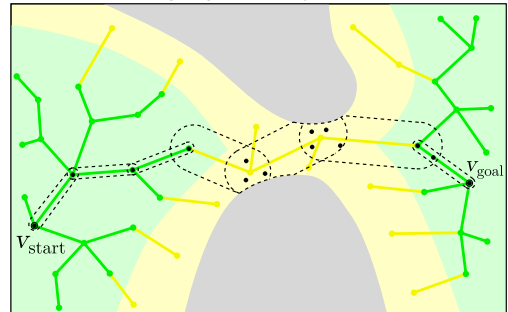
(c) The dotted regions represent the ranges of π_S or π_N for each edge along the guide route. The π_S regions are thin because we are guaranteed to find a solution directly along the routes where C_S is satisfied. Conversely, π_N may need to search a broader area to find a solution. $\Phi_{\mathcal{M}}$ is applied to the edges of the guide route, generating sets of states (black dots) where the ranges of the planners overlap.
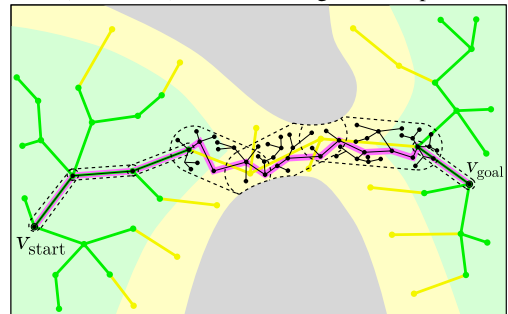
(d) States and whole body paths (black dots and edges) are generated by $\Omega_{\mathcal{M}}$ using the guide route and the states generated by $\Phi_{\mathcal{M}}$. These elements are put into $\Gamma_{\mathcal{M}}$ until a complete whole body path is found from the start to the goal.

Fig. 3: Illustration of the RPG procedure

which would make the method in [23] more applicable. For this paper, we use the "hard" and "soft" constraint approach, and we refer to this high-level planner as $\Pi_{\mathcal{P}}$.

We initiate $\Gamma_{\mathcal{M}}$ with one or more "start" states, $V_{\mathcal{M}}^{\text{start}}$, and one or more "goal" states, $V_{\mathcal{M}}^{\text{goal}}$. $\Gamma_{\mathcal{P}}$ is initiated with the projections of these states, $P(V_{\mathcal{M}}^{\text{start}})$ and $P(V_{\mathcal{M}}^{\text{goal}})$. The high-level motion planner of choice, $\Pi_{\mathcal{P}}$, is used to find a route $(e_1, ..., e_n \in E_{\mathcal{P}})$ which connects a start projection to a goal projection while remaining within C_N. The solved path generated by $\Pi_{\mathcal{P}}$ represents a guide route, similar to the "guide trajectories" of [24]. An illustration of this process can be seen in Figs. 3a-3b. The guide route is used to focus the efforts of the next stage, $\mathcal{M}$.

Provided the guide route found by $\mathcal{P}$, we want to use $\Omega_{\mathcal{M}}$ to examine each segment of the route to see if it can be realized in X. We define $\Omega_{\mathcal{M}}$ as:

$$\Omega_{\mathcal{M}}(x_i^0, e_i, x_i^f) = \begin{cases} \pi_S(x_i^0, e_i, x_i^f) & \text{if } C_S(e_i) \\ \pi_N(x_i^0, e_i, x_i^f) & \text{if } C_N(e_i) \wedge \neg C_S(e_i) \end{cases}$$

where π_S and π_N are sub-planners which produce full state space trajectories given a sub-start state (x_i^0), a sub-goal state (x_i^f), and an edge (e_i) of $E_{\mathcal{P}}$ which is used as a "guide". The "guide" edge may be used to compute a heuristic, determine footstep locations, or focus randomized samples, depending on the nature of the planner. We choose π_S such that it is guaranteed to quickly find a solution along routes where the sufficient conditions C_S are satisfied. π_N is a whole body motion planner, ideally with a probabilistic completeness guarantee. π_N can be applied to edges which satisfy the necessary conditions C_N even if the sufficient conditions are not satisfied, but it is not guaranteed to return a solution.

To find a feasible continuous motion through state space, we need to choose x_i^f and x_{i+1}^0 to be equal. Define $X_0(e_i)$ to be the set of states that can be used as sub-start states, x_i^0, for $\Omega_{\mathcal{M}}(x_i^0, e_i, x_i^f)$, and $X_f(e_i)$ to be the set that can be used as sub-goal states, x_i^f. We can then define $\Phi_{\mathcal{M}}(e_i, e_{i+1}) = X_f(e_i) \cap X_0(e_{i+1})$. Using $\Phi_{\mathcal{M}}$ to determine the endpoints (sub-starts and sub-goals) used by $\Omega_{\mathcal{M}}$ ensures that the RPG is able to create continuous state trajectories. Note that $\Phi_{\mathcal{M}}(e_0, e_1) = V_{\mathcal{M}}^{\text{start}} \cap X_0(e_1)$, and $\Phi_{\mathcal{M}}(e_n, e_{n+1}) = X_f(e_n) \cap V_{\mathcal{M}}^{\text{goal}}$. $\Phi_{\mathcal{M}}$ is illustrated by the black dots in the overlapping dotted regions of Fig. 3c. Determining $X_0(e)$ and $X_f(e)$ will depend on the implementation of π_S and π_N, but most planners have either a discrete set of permissible endpoints or a continuous set that can be sampled from. In practice, many planners allow multiple start and goal states to be specified per query.

The overall procedure for planning with the RPG is to generate a guide route $(e_1, ..., e_n \in E_{\mathcal{P}})$ by constructing $\Gamma_{\mathcal{P}}$ with $\Pi_{\mathcal{P}}$ and feeding that route through $\Phi_{\mathcal{M}}$ and then through $\Omega_{\mathcal{M}}$. Whenever $\Omega_{\mathcal{M}}$ identifies feasible state trajectories, the states and edges of those trajectories are added to $\Gamma_{\mathcal{M}}$. A solution is found when $\Gamma_{\mathcal{M}}$ contains a path from a start to a goal state. The procedure is illustrated in Fig. 3.

Depending on the implementation of π_N, it may take an indeterminable amount of time to produce a solution.

Moreover, it might not be able to produce a solution for some e_i if the edge is not a truly feasible guide route. Rather than waiting for $\Omega_{\mathcal{M}}$ to return a result of success or failure, the stage $\mathcal{P}$ can search for alternative guide routes by deleting any of the indeterminate edges of the guide route from $\Gamma_{\mathcal{P}}$ and then continuing to grow $\Gamma_{\mathcal{P}}$ in parallel to $\mathcal{M}$. This parallelism allows the RPG to avoid being bottlenecked by challenging routes when alternatives exist. When elements are added to $\Gamma_{\mathcal{M}}$, their projections can be added to $\Gamma_{\mathcal{P}}$ as "possible" elements to assist the ongoing high-level search.

III. WALKING IN SEMI-UNSTRUCTURED ENVIRONMENTS
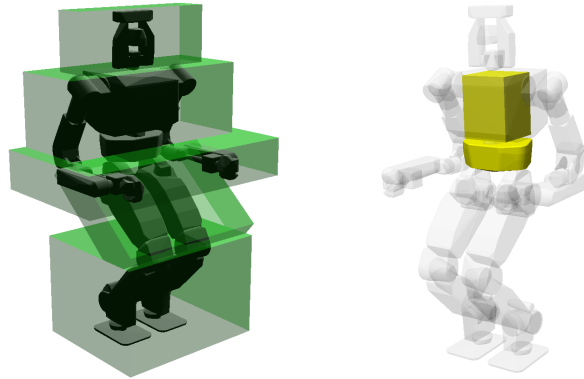
(a) Bounding geometry used for sufficient conditions (Green)

(b) Minimal geometry used for necessary conditions (Yellow)

Fig. 4: Collision geometries used for the (a) sufficient and (b) necessary conditions. Since (a) occupies more space, the constraint manifold for the sufficient conditions is smaller—and therefore more restrictive—than the manifold for the necessary conditions.

We seek to find feasible motion plans for a bipedal robot to traverse a "semi-unstructured" environment. In the context of this paper, we define "semi-unstructured" to mean that the terrain is structured—flat and even—but there are arbitrary unstructured obstacles throughout the environment. The robot may need to step over, duck under, or maneuver around these obstacles using whole body motions. The planner is provided with a kinematic model of the robot, a geometric model of all obstacles—including walls—and a layout of the floor. No other contextual information is provided to the planner (such as explicitly labelling doorways or passages).

Planning a path through a semi-unstructured environment entails finding a physically feasible sequence of footsteps combined with whole body motions that are constrained to those footsteps which move the robot from the start state to the goal state. To apply the RPG to the problem of walking in semi-unstructured environments, we must first define the possibility exploration space for route exploration, as well as the sufficient and the necessary conditions that pertain to the robot's ability to take steps and move its whole body.

A. Possibility Exploration Space

The vertices of the RPG exist in the possibility exploration space, $\mathcal{E}$. For the problem of walking in semi-unstructured environments, we define $\mathcal{E}$ as SE(3), which offers enough

parameters to design effective sufficient and necessary conditions. Each point in $\mathcal{E}$ represents a transformation of the robot's root (pelvis) link. As we generate edges in $\mathcal{E}$, we are creating routes to guide the root link towards its destination. Figure 5 illustrates what a section of an RPG may look like.

B. Sufficient Conditions

We define the sufficient conditions by first constructing a bounding box which encapsulates all the motions that the robot might exhibit while walking or turning in any direction using some basic gait generator. The bounding geometry can be seen in Fig. 4a. If nothing in the environment is colliding with this geometry, then the robot is guaranteed to not collide with anything while performing its normal gait. For this condition, validating an edge $e(v_a, v_b)$ in SE(3) involves applying a dense sampling of SE(3) transformations from v_a to v_b to the bounding geometry and checking for collisions with the environment. In addition, we require the corners of the support polygon of each foot to be supported by solid ground while the robot stands in the nominal configuration shown in Fig. 4. We can then employ a simple gait generator that is guaranteed to find a solution when these sufficient conditions are satisfied, providing us with π_S.

C. Necessary Conditions

The necessary conditions are much less restrictive than the sufficient conditions. The collision geometry used for the necessary conditions can be seen in Fig. 4b. This minimal collision geometry is a subset of the actual robot's collision geometry. This subset is chosen such that it is completely unaffected by any joint motion (assuming the root link remains fixed in place). Therefore, if this geometry is in collision with any obstacles, then the robot is guaranteed to be in collision, no matter what its joint positions are. If this minimal geometry cannot sweep along an edge, then that edge is considered "impossible" and is left out of the RPG. Secondly, there must be at least one foothold within reach of the root link. These conditions are similar to the necessary conditions used in [24], although we leave the minimal collision geometry as it is instead of inflating it.

D. Guided Multi-modal Planning

Once the sufficient and necessary conditions are defined, the only remaining decision in our algorithm is the choice of whole body motion planner for π_N. In this work, we use Random-MMP by Hauser et. al. [13] because it solves motion planning problems with frequent changes in contacts.

Random-MMP by itself can solve the semi-unstructured problems posed in this paper with the theoretical property of probabilistic completeness. However, Random-MMP performs best when it is able to utilize informed sampling. For example, when used for manipulation, Random-MMP should be provided with information on how to sample a state that is within the reachable region of the manipulatable target. In this paper, we aim to solve problems wihtout providing extra contextual information, meaning Random-MMP on its own could take prohibitively long to find a solution.
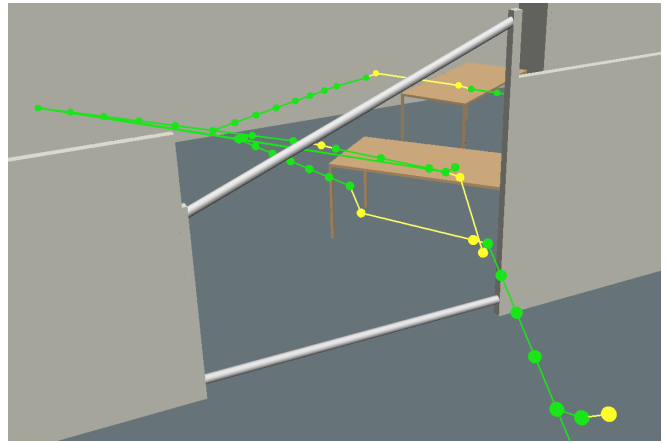

Fig. 5: Snapshot of an RPG. Green edges are "possible"; yellow edges are "indeterminate". Portions of the graph that are in wide open hallways are Possible, whereas elements that squeeze between obstacles are Indeterminate.

The key advantage of the RPG is that indeterminate edges can be used to guide low-level planners. Therefore, even though the original problem does not provide us with information about passages in the environment, indeterminate edges can be seen as potential passages that are worthy of further inspection. This allows us to turn an uninformed search into an informed search. We will refer to this as Guided-MMP to distinguish it from uninformed Random-MMP. Figure 6 illustrates the difference in behavior between these methods. The uninformed search takes tens of minutes whereas the guided search takes tens of seconds. The only difference between the uninformed vs. guided approaches is that the guided approach focuses its randomized mode and configuration samples to be near the indeterminate edges, according to a Normal Distribution. Meanwhile, the uninformed search broadly samples the entire domain.

IV. EXPERIMENTS

We present two virtual experimental scenarios. The first scenario, shown in Fig. 6, is referred to as the "Limbo Scenario". The robot must traverse from one side of the platform to the other while ducking underneath a limbo bar. It demonstrates the performance difference between an uninformed Random-MMP search versus Guided-MMP which is supplemented by the Randomized Possibility Graph. Even in such a simple scenario, the RPG improves performance by two orders of magnitude.

The second scenario is referred to as the "Four Routes Scenario". It demonstrates some of the larger scale capabilities of the RPG. The robot must navigate from the southwest corner to the northeast corner of the floor. Each room has two entrances, making a total of four possible routes the robot may choose from. Each doorway has its own challenges associated with passing through it, seen in Fig. 1.

The route which is easiest for the planner to find passes through the west doorway, down the hallway, and then through the east doorway as shown in Fig 8a. Those doorways are the easiest to plan for because they have the most

TABLE I: Time performance results tested on an Intel® Xeon® Processor E3-1290 v2 (8M Cache, 3.70 GHz) with 16GB of RAM. This table shows the time spent (in seconds) and the rate of success for 30 trials of each scenario and each variation. The standard deviation is given in parentheses. Numerous trials are run because randomized planners have non-deterministic run times and non-deterministic success rates. All trials were given a one-hour timeout, at which point we consider the trial to have failed.

	Uninformed Random-MMP		Randomzied Possibility Graph	
Scenario	Time	Success	Time	Success
Limbo Scenario (Fig. 6)	2355.5 (1000.3)	86.67%	20.56 (26.51)	100%
Four Routes Scenario (Fig. 7)	3600 (0)	0%	68.44 (33.98)	96.67%
West Door Blocked (Fig. 8b)	3600 (0)	0%	295.54 (205.847)	96.67%
East Door Blocked (Fig. 8c)	3600 (0)	0%	349.60 (195.38)	100%
Bars Removed (Fig. 8d)	3600 (0)	0%	7.31 (5.73)	100%
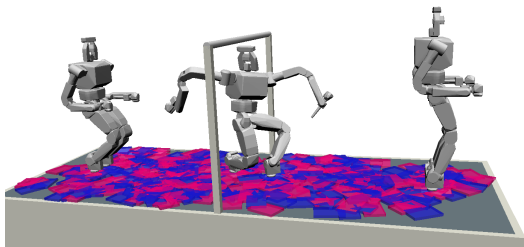
(a) Uninformed Random-MMP: Exhaustive search
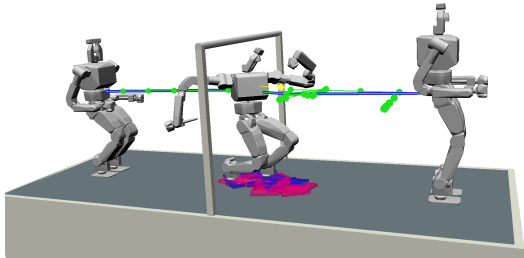
(b) Guided-MMP: Focuses the search around the indeterminate (yellow) edge of the RPG.

Fig. 6: Limbo scenario. The difference in search behaviors for uninformed (a) versus guided (b) versions of Random-MMP. Blue and magenta rectangles are the foot placements of the left and right (respectively) feet for each sampled mode.

Fig. 7: Basic "Four Routes" scenario.

"expansiveness" as defined in [25]. Since the planner chooses the first successful plan that it finds, this will be the most commonly chosen route (although the randomized nature of the planner does not guarantee that this will always be the chosen route). We can tweak the environment to force the robot to find a path through the south doorway by blocking off the west doorway (as in Fig. 8b). Similarly, the robot can be forced to find a path through the north doorway by blocking off the east (as in Fig. 8c). Each of these modifications result in considerably longer run times as shown in Table I. This shows that when the planner is not required to find routes through challenging regions, it will naturally tend to favor regions where it can find paths easily.

In some cases, there may be a clear path from the start to the goal. To simulate this, in Fig. 8d we remove the bars that are obstructing the north and south doorways. The result is a lean graph and a short plan time as seen in Table I.

V. CONCLUSIONS

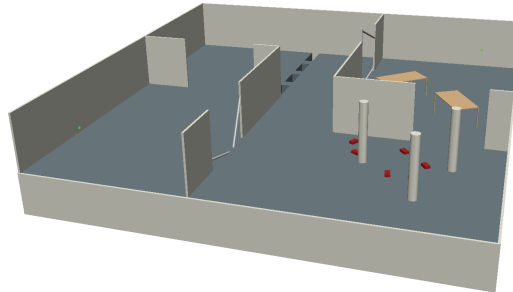This work presented a new high-level algorithm for identifying routes in semi-unstructured environments, and a way to leverage this information in lower-level motion planners. The method presented improves performance of existing algorithms by orders of magnitude, making large-scale problems tractable when they would not have been previously.

The use of necessary conditions and guide routes shares some similarities to the reachability-based planner of Tonneau, et. al. [24]. The key novelty of this work is that we also leverage sufficient conditions, which are not used by Tonneau. Furthermore, we use probabilistically complete whole body planners when evaluating the indeterminate portions of the guide routes, which we believe allows the robot to more fully utilize its maneuverability, at the cost of longer run times. In future work, we will apply the RPG to uneven terrain, at which point a direct comparison between the RPG and the reachability-based planner will be possible.

Future work will also investigate other ways that the RPG can be leveraged to generate plans. For example, the graph could consider the possibilities of dynamic actions rather than just quasi-static actions. To improve the evaluation of indeterminate edges, it would be natural to use a variety of low-level planners in parallel that could compete to confirm indeterminate edges rather than using a single catch-all probabilistically complete low-level planner which might not perform particularly well in all scenarios. We may also be able to define some distinct variations of sufficient conditions where each variation corresponds to a different flavor of gait generator, allowing a broader range of edges to be covered by sufficient conditions.
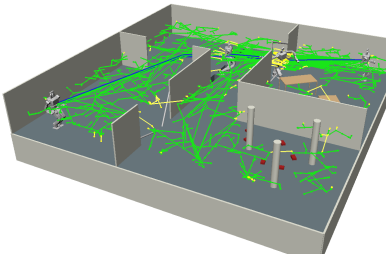
The completeness of the RPG is currently being investigated. Future work will examine under what conditions the RPG may be considered probabilistically complete. This will likely depend on the completeness of the low-level planners that it can utilize.
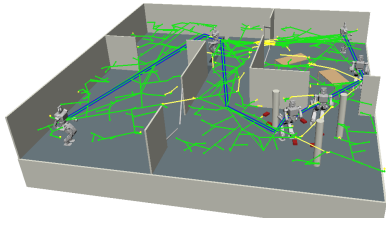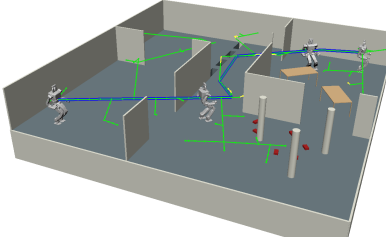
(a) Solution for the standard variation of the scenario
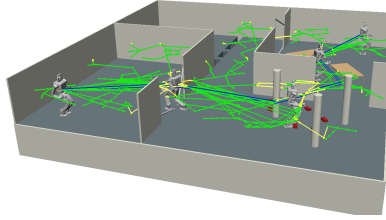
(b) Solution when the west doorway is blocked off

(c) Solution when the east doorway is blocked off

(d) Solution with bars removed from the north and south

Fig. 8: Variations of the "Four Routes Scenario".

REFERENCES

[1] J. Pettré, J.-P. Laumond, and T. Siméon, "A 2-stages locomotion planner for digital actors," in *Proceedings of the 2003 ACM SIG-GRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2003, pp. 258–264.

[2] N. Perrin, O. Stasse, F. Lamiraux, Y. J. Kim, and D. Manocha, "Real-time footstep planning for humanoid robots among 3d obstacles using a hybrid bounding box," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 977–982.

[3] J. J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, "Footstep planning among obstacles for biped robots," in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 1. IEEE, 2001, pp. 500–505.

[4] J. Kuffner, S. Kagami, K. Nishiwaki, M. Inaba, and H. Inoue, "Online footstep planning for humanoid robots," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 1. IEEE, 2003, pp. 932–937.

[5] J. Chestnutt, J. Kuffner, K. Nishiwaki, and S. Kagami, "Planning biped navigation strategies in complex environments," in *IEEE Int. Conf. Hum. Rob., Munich, Germany*, 2003.

[6] J. Garimort, A. Hornung, and M. Bennewitz, "Humanoid navigation with dynamic footstep plans," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3982–3987.

[7] N. Perrin, O. Stasse, L. Baudouin, F. Lamiraux, and E. Yoshida, "Fast humanoid robot collision-free footstep planning using swept volume approximations," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 427–439, 2012.

[8] M. Fallon, S. Kuindersma, S. Karumanchi, M. Antone, T. Schneider, H. Dai, C. P. D'Arpino, R. Deits, M. DiCicco, D. Fourie, *et al.*, "An architecture for online affordance-based perception and whole-body planning," *Journal of Field Robotics*, vol. 32, no. 2, pp. 229–254, 2015.

[9] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014, pp. 279–286.

[10] A. Hereid, E. A. Cousineau, C. M. Hubicki, and A. D. Ames, "3d dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics," in *IEEE International Conference on Robotics and Automation (ICRA). IEEE*, 2016.

[11] M. J. Powell, H. Zhao, and A. D. Ames, "Motion primitives for human-inspired bipedal robotic locomotion: walking and stair climbing," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 543–549.

[12] K. Hauser and J.-C. Latombe, "Multi-modal motion planning in non-expansive spaces," *The International Journal of Robotics Research*, 2009.

[13] K. Hauser and V. Ng-Thow-Hing, "Randomized multi-modal motion planning for a humanoid robot manipulation task," *The International Journal of Robotics Research*, vol. 30, no. 6, pp. 678–698, 2011.

[14] K. Hauser, T. Bretl, J.-C. Latombe, K. Harada, and B. Wilcox, "Motion planning for legged robots on varied terrain," *The International Journal of Robotics Research*, vol. 27, no. 11-12, pp. 1325–1349, 2008.

[15] T. Bretl, S. Lall, J.-C. Latombe, and S. Rock, "Multi-step motion planning for free-climbing robots," in *Algorithmic Foundations of Robotics VI*. Springer, 2004, pp. 59–74.

[16] L. Sentis and O. Khatib, "A whole-body control framework for humanoids operating in human environments," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*. IEEE, 2006, pp. 2641–2648.

[17] T. Sugihara and Y. Nakamura, "Whole-body cooperative balancing of humanoid robot using cog jacobian," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 3. IEEE, 2002, pp. 2575–2580.

[18] M. Gienger, H. Janssen, and C. Goerick, "Task-oriented whole body motion for humanoid robots," in *5th IEEE-RAS International Conference on Humanoid Robots, 2005*. IEEE, 2005, pp. 238–244.

[19] D. Dubois and H. Prade, *Possibility theory: an approach to computerized processing of uncertainty*. Springer Science & Business Media, 2012.

[20] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[21] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 2. IEEE, 2000, pp. 995–1001.

[22] D. Berenson, S. Srinivasa, D. Ferguson, and J. Kuffner, "Manipulation planning on constraint manifolds," in *IEEE International Conference on Robotics and Automation (ICRA '09)*, May 2009, pp. 625–632.

[23] T. Kunz and M. Stilman, "Manipulation planning with soft task constraints," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 1937–1942.

[24] S. Tonneau, N. Mansard, C. Park, D. Manocha, F. Multon, and J. Pettré, "A reachability-based planner for sequences of acyclic contacts in cluttered environments," in *International Symposium on Robotics Research (ISSR 2015)*, 2015.

[25] D. Hsu, J.-C. Latombe, and R. Motwani, "Path planning in expansive configuration spaces," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 3. IEEE, 1997, pp. 2719–2726.