High-Level Path Planning in Hostile Dynamic Environments

Extended Abstract

Jacopo Banfi
Cornell University
Ithaca, New York
jb2639@cornell.edu

Mark Campbell
Cornell University
Ithaca, New York
mc288@cornell.edu

ABSTRACT

In this paper, we introduce and study a graph-based variant of the path planning problem arising in hostile environments. Here, the robot must reach a given destination while avoiding being intercepted by probabilistic entities which exist in the graph with a given probability and move according to a probabilistic motion pattern. Given a deadline to reach its goal, the robot must compute a path that maximizes its chances of survival. To solve this problem, which is proven to be NP-hard, we present a convex Mixed-Integer Nonlinear Program to compute optimal solutions and a more scalable heuristic algorithm.

KEYWORDS

Robot control; Path planning; Hostile environments

ACM Reference Format:
Jacopo Banfi and Mark Campbell. 2019. High-Level Path Planning in Hostile Dynamic Environments. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 3 pages.

1 INTRODUCTION

Motivated by the recent interest of the AI community in the navigation of hostile environments [1], in this paper we introduce and study a graph-based variant of the path planning problem arising in those kinds of settings. Here, the robot must reach a given destination while avoiding being intercepted by some entities –that we generically call "obstacles"– existing with a given probability, moving according to a probabilistic motion pattern, and capable of "intercepting" the robot along its path (e.g., within a given range). Given a deadline to reach its goal, the robot must compute a path that maximizes its chances of survival (see Figure 1 for an example problem instance). Possible applications include, for instance, those related to intrusion scenarios, where obstacles may represent enemy guards deployed to protect the entrances of a building [6]. Another example is given by situations where the robot's sensors have been compromised by an attacker, who is trying to make the robot crash by injecting false dynamic obstacles into the robot's perception pipeline [5]. To solve this problem, which is proven to be NP-hard, we present a convex Mixed-Integer Nonlinear Program (MINLP) to compute optimal solutions and a more scalable heuristic algorithm.

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

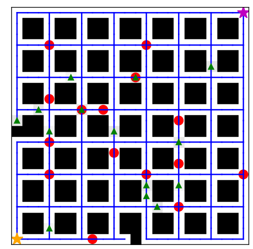

Figure 1: A problem instance. The environment is a 22x22 grid graph with holes (the black squares). The robot must move from the bottom-left corner to the upper-right. Static probabilistic obstacles (green triangles) and dynamic probabilistic obstacles (red circles) can intercept the robot. The robot must reach the goal within a given deadline by traveling along the path that minimizes its interception probability.

2 PROBLEM SETTING

Let $G = (V, E)$ be a connected, undirected, simple graph with unitary edge lengths representing the environment, and let $V = \{1, \ldots, n\}$. A robot must plan a path on G in the form of an ordered sequence of vertices $\pi = (v_s, v_1, v_2, \ldots, v_g)$ from a start vertex v_s to a goal vertex v_g. The robot moves deterministically on G: time evolves in discrete steps and, at each step, the robot can either stay still at the current vertex, or move along a graph edge. We use $\pi[i]$ to denote the position of the robot at the i-th time step when executing π, $|\pi|$ to denote the number of time steps required to reach the goal in π, and $\mathcal{P}$ to denote the set of all the possible paths according to our definition. We assume that the environment may contain *probabilistic obstacles* able to *intercept* the robot along its path, which can be either *static* or *dynamic*. The existence and the interception events related to a given obstacle in G are assumed to be *independent* of those of all the others.

The set of static probabilistic obstacles is denoted by S. Each $s \in S$ is completely described by a probability of existence $p_s < 1$ and by a set of vertices $V(s) \subseteq V \setminus \{v_s, v_g\}$ inducing a connected subgraph on G. The semantic associated with a static probabilistic obstacle s is simple: if the robot executes a path π traversing *any* of the vertices in $V(s)$, it is *intercepted* by s with probability p_s and it *survives* s with probability $1 - p_s$. We use the function $\sigma : \mathcal{P} \times S \longrightarrow \{0, 1\}$ to associate each $\langle \pi, s \rangle$ pair with the presence ($\sigma(\pi, s) = 1$) or absence ($\sigma(\pi, s) = 0$) of at least one passage of π through a vertex of s.

The set of dynamic probabilistic obstacles is denoted by D. Each $d \in D$ is associated with a probability of existence p_d and, at each time step t, is described by a belief vector

$$b^d(t) = [b^d_0(t), b^d_1(t), \ldots, b^d_n(t)]. \quad (1)$$

The first element of the vector, $b_0^d(t)$, represents the probability that the robot has already been *intercepted* by d by time t if it executes a given path π, conditioned on the obstacle existence in the environment. All the subsequent vector elements $b_v^d(t), v \in \{1, \dots, n\}$ represent the probability that at time t the obstacle is in vertex v, again assuming that the obstacle is actually present. Belief vectors evolve according to (a) the obstacles' probabilistic motion models, and (b) the interception events associated with π.

The probabilistic motion model is assumed to be Markovian and not changing between two subsequent steps. We can therefore represent it by means of a single stochastic matrix M^d whose generic entry M_{uv}^d represents the probability that d will move from u to v between two subsequent steps t and $t+1$.

Interception events related to a dynamic obstacle d are described by means of n *interception matrices* $N^{d;v}, v \in V$ having size $(n+1) \times (n+1)$. The effect of the application of an interception matrix on the belief vector is to "move" some probabilities related to the obstacle being in vertices allowing it to intercept the robot when located in v to the robot's interception state $b_0^d(t)$ (see [4] for an example in the context of a search problem).

For a robot executing path π, the belief update equation describing its interactions with dynamic obstacle d is

$$b^d(t+1) = b^d(t) \begin{bmatrix} 1 & 0 \\ 0 & M^d \end{bmatrix} N^{d;\pi[t+1]}, \tag{2}$$

where the 0s denote vectors of appropriate size. The semantic associated with a dynamic probabilistic obstacle d is as follows: if the robot executes path π, it is *intercepted* by d with probability $p_d \cdot b_0^d(|\pi|)$ and it *survives* d with probability $1 - p_d \cdot b_0^d(|\pi|)$. We consider the following optimization problem:

Problem 1. *Given $\langle G, S, D \rangle$ and a deadline $T \geq d(v_s, v_g)$, compute the path π^* defined as*

$$\pi^* = \arg\max_{\pi \in \mathcal{P}} \prod_{s \in S} (1-p_s)^{\sigma(\pi,s)} \prod_{d \in D} (1 - p_d \cdot b_0^d(|\pi|)) \text{ s.t.} \tag{3}$$

$$\text{Belief Update Equation (2)} \quad \forall d \in D,\ t \in \{1, \dots, |\pi|\} \tag{4}$$

$$|\pi| \leq T \tag{5}$$

We have proven that the decision version of Problem 1 is NP-hard even on seemingly simple problem instances ($T \leq |V|$, rectangular grid graphs where dynamic obstacles have a very small "interception range", move at 1 cell/step, and never cross static obstacles nor each others' paths at the same step).

3 ALGORITHMS

To compute optimal solutions to Problem 1, we first propose a convex MINLP. In our formulation, legal paths are modeled on a time-stamped version of G with vertex set $V^t = \{\langle v, t \rangle \in V \times \{0, 1, \dots, T\} | d(v_s, v) \leq t \wedge t + d(v, v_g) \leq T\} \cup \{v_g'\}$, with v_g' acting as dummy goal vertex (in general, the robot could reach the goal between steps $d(v_s, v_g)$ and T), and (directed) arc set $A^t = \{(\langle u, t \rangle, \langle v, t+1 \rangle) | \langle u \neq v_g, t \rangle \in V^t \wedge \langle v, t+1 \rangle \in V^t \wedge [(u,v) \in E \vee u = v]\} \cup \{(\langle v_g, t \rangle, v_g') | \langle v_g, t \rangle \in V^t\}$. This allows to use *binary path variables* and linear constraints to express the robot's movements in G between two subsequent steps (see [2]). Interception events related to static obstacles are modeled with the help of *binary variables* z_s, which are enforced to take value 1 by means of linear constraints iff the robot traverses any of the vertices of static obstacle s at any time step along its path. Interception events related to dynamic obstacles are modeled by defining *continuous belief variables* $\beta_{i;t}^d$, representing the entries $b_i^d(t)$ of the dynamic obstacles' belief vectors up to step T. Again, we use only linear constraints to bind the path variables to an evolution of the belief variables coherent with Eq. (2) (if a path reaches the goal at $\hat{t} < T$, we make sure that $\beta_0^d(t)$ does not change after $\hat{t}$). The objective function can be stated as

$$\text{maximize} \sum_{s \in S} z_s \log(1-p_s) + \sum_{d \in D} \log(1 - p_d \cdot \beta_{0;T}^d), \tag{6}$$

hence obtaining a convex MINLP.

We also present an algorithm allowing to obtain heuristically good solutions in reasonable time ($O(n|S| + n^2|D|T)$ worst-case runtime) for large scale problems. The idea is to compute the shortest $(\langle v_s, 0 \rangle, v_g')$ path on the directed graph (V^t, A^t) introduced above, where each arc $(\langle u, t \rangle, \langle v, t+1 \rangle) \in A^t$ is associated with a *fixed weight* computed as the negative log probability that the robot will not be intercepted by any probabilistic obstacle when located in v at time $t+1$, assuming no other prior interaction of the robot with them. This algorithm should in general be able to make the robot avoid space-temporal regions associated with a high probability of interception. Besides, it computes the optimal solution when the robot can survive with probability 1.

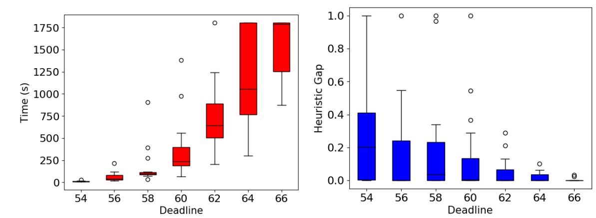

Figure 2: Results obtained on the 28×28 grids. Left: MINLP solution times. Right: heuristic gaps computed as (optimal_solution - heuristic_solution)/optimal_solution.

4 RESULTS

We assess the performance of the proposed algorithms on grid instances similar to that shown in Figure 1, simulating an intrusion scenario: each dynamic obstacle can intercept the robot within 1 cell, and is moving around a black square at 1 cell/step with a 0.05 probability of remaining still. We consider grids of size $m \times m$, $m \in \{16, 22, 28\}$, and use the SCIP solver [3] with a 30 minutes timeout to solve the MINLPs. Our results show that the MINLP approach is generally able to compute the optimal solution in reasonable time in instances of moderate size ($m = 16, 22$), and that the heuristic behaves better as we increase the deadline T (as the survivability approaches 1). Figure 2 shows the results obtained in the most challenging 28×28 instances, having $p_s \in \{0.05, 0.1\}$ for each $s \in S$ and $p_d = 1$ for each $d \in D$.

ACKNOWLEDGMENTS

This work is supported by the NASA grant NNX16AB09G.

REFERENCES

[1] N. Agmon. 2018. Robotic strategic behavior in adversarial environments. In *Proc. IJCAI*. 5106–5110.

[2] J. Banfi, J. Guzzi, F. Amigoni, E. Feo Flushing, A. Giusti, L. Gambardella, and G. Di Caro. 2018. An integer linear programming model for fair multitarget tracking in cooperative multirobot systems. *Auton Robot* (2018). https://link.springer.com/article/10.1007/s10514-018-9735-4 To appear.

[3] A. Gleixner, M. Bastubbe, L. Eifler, T. Gally, G. Gamrath, R. Gottwald, G. Hendel, C. Hojny, T. Koch, M. Lübbecke, S. Maher, M. Miltenberger, B. Müller, M. Pfetsch, C. Puchert, D. Rehfeldt, F. Schlösser, C. Schubert, F. Serrano, Y. Shinano, J. Viernickel, M. Walter, F. Wegscheider, J. Witt, and J. Witzig. 2018. *The SCIP Optimization Suite 6.0.* Technical Report. Optimization Online. http://www.optimization-online.org/DB_HTML/2018/07/6692.html

[4] G. Hollinger, S. Singh, J. Djugash, and A. Kehagias. 2009. Efficient multi-robot search for a moving target. *Int J Robot Res* 28, 2 (2009), 201–219.

[5] J. Liu, J. Corbett-Davies, A. Ferraiuolo, A. Ivanov, M. Luo, G. E. Suh, A. C. Myers, and M. Campbell. 2018. Secure autonomous cyber-physical systems through verifiable information flow control. In *Proc. CPS-SPC*.

[6] D. Portugal and R. Rocha. 2011. A survey on multi-robot patrolling algorithms. In *Technological Innovation for Sustainability*, L. M. Camarinha-Matos (Ed.). Springer Berlin Heidelberg, 139–146.