Maximum-Reward Motion in a Stochastic Environment: The Nonequilibrium Statistical Mechanics Perspective

Citation: Ma, Fangchang, and Sertac Karaman. "Maximum-Reward Motion in a Stochastic Environment: The Nonequilibrium Statistical Mechanics Perspective." Algorithmic Foundations of Robotics XI. Ed. H. Levent Akin et al. Vol. 107. Cham: Springer International Publishing, 2015. 389–406.

As Published: http://dx.doi.org/10.1007/978-3-319-16595-0_23

Publisher: Springer International Publishing

Persistent URL: http://hdl.handle.net/1721.1/105883

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Maximum-reward Motion in a Stochastic Environment: The Nonequilibrium Statistical Mechanics Perspective

Fangchang Ma and Sertac Karaman

Massachusetts Institute of Technology, Cambridge MA 02139, USA,
{fcma,sertac}@mit.edu,

Abstract. We consider the problem of computing the maximum-reward motion in a reward field in an online setting. We assume that the robot has a limited perception range, and it discovers the reward field on the fly. We analyze the performance of a simple, practical lattice-based algorithm with respect to the perception range. Our main result is that, with very little perception range, the robot can collect as much reward as if it could see the whole reward field, under certain assumptions. Along the way, we establish novel connections between this class of problems and certain fundamental problems of nonequilibrium statistical mechanics. We demonstrate our results in simulation examples.

Keywords: Motion planning, stochastic environments, nonequilibrium statistical mechanics.

1 Introduction

Nonequilibrium statistical mechanics is a branch of physics that studies systems operating at out-of-equilibrium states [1–3]. Although the ideas originated in the physics literature, the theory has profound applications that lie well outside the domain of physics, such as biology [4, 3], stock markets [5], and highway traffic [6, 7]. Arguably, it is for this reason that developing a fundamental and comprehensive understanding of nonequilibrium statistical mechanics is considered to be one of the grand challenges in our time, both by the U.S. Department of Energy [8, 9] and the U.S. National Academy of Sciences [10].

In this paper, we point out novel connections between the fundamental problems of nonequilibrium statistical mechanics and a large class of robot motion planning and control problems. With the help of these connections, we design practical algorithms with provable performance guarantees for planning problems involving agile robots operating in stochastic environments. In what follows, we briefly introduce this class of problems and list our contributions.

We consider a large class of problems involving a robotic vehicle navigating in a stochastic reward field to collect maximal reward. Let us motivate these problems with an example. Consider an environmental monitoring system, where mobile robotic vehicles and stationary sensing devices work together to collect valuable information about the state of the environment, as in Figure 1. Imagine small sensing devices that house primitive sensors, for example, for seismic, acoustic, or magnetic measurements. Along with sensing, these devices include communication equipment and (primitive) computational platforms. Suppose these sensing devices are deployed throughout the environment for persistent monitoring purposes. Rather than attempting to form an ad-hoc network, we envision a mobile data-harvesting vehicle that traverses the environment, discovers the sensing devices on the fly, and approaches them to harvest their data.[1]

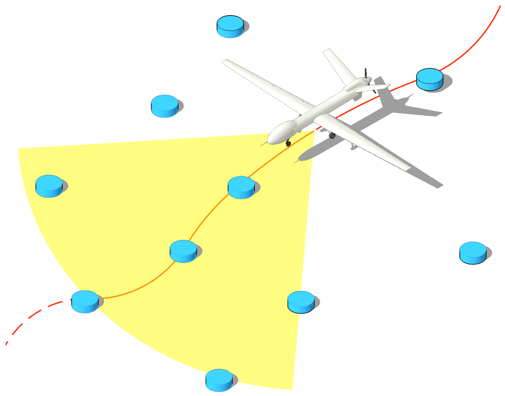


Fig. 1: An environmental monitoring system, where the blue cylinders are sensing devices and the vehicle tries to collect as much data from them as possible.

The robot may not know the precise positions of the sensing devices *a priori*. Instead, sensors are discovered on the fly, and the robot makes small corrections in its trajectory to collect as much information as possible. Clearly, the amount of information that can possibly be collected depends on the agility of the robot (its actuation capabilities) as well as its perception range (how soon it can discover the sensors). In this setting, we consider the following fundamental problems: *How quickly can the mobile robots harvest the data from the field, given their perception, actuation, and computation capabilities? What are the planning algorithms that achieve the optimal performance?*

Let us note that similar problems [11] arise in a large class of applications, including spy planes taking pictures of unexpected enemies, and rescue vessels

[1] Ad-hoc sensor networks may also be very valuable for environmental monitoring. In fact, such technologies have been developed over past years. We note that the presented approach is for motivational purposes. Yet, it may be beneficial over the ad-hoc network approach due to substantial energy savings at the stationary sensors (as communication requirement is much lower); hence, the sensor nodes require less maintenance. The main drawbacks are additional complexity of a mobile vehicle, and communication delay due to the mobile vehicles physically carrying the data.

saving lives after disasters, where target locations are discovered on the fly. To generalize, we consider a robot that is traversing a stochastic "reward field", where the precise value of the reward is discovered on the fly. Given the statistics of the reward, we aim to answer fundamental questions regarding the optimal performance that can be achieved.

We answer some of the aforementioned questions by establishing strong connections between this class of problems and nonequilibrium statistical mechanics. Roughly speaking, we view the robot as a particle traveling in a stochastic field. This perspective allows us to directly apply some of the recent results from mathematical physics to characterize various properties of agile robotics.

Designing planning algorithms for agile robots to avoid obstacles in cluttered environments has long been a focus of robotics [12–14]. In contrast, planning problem for collecting maximal reward in a stochastic environment has received relatively little attention, although similar problems were considered in the operations research literature (see, *e.g.*, [15]).

The analysis in this paper is fundamentally different from these references, as we utilize the mathematical foundations of nonequilibrium statistical mechanics. The results we utilize were reported in the mathematical physics literature fairly recently [16–22]. In fact, the connections we establish between mathematical physics and this class of maximum-reward motion planning problems and algorithms may be interesting on their own right, inspiring a novel class of analysis techniques and practical algorithms with formal performance guarantees.

This paper is organized as follows. In Section 2, we provide a more precise problem definition, and we discuss a set of algorithmic approaches that solve this problem in Section 3. We devote Section 4 to a mathematically rigorous analysis of the proposed algorithms. We lay out the connections with nonequilibrium statistical mechanics also in this section. In Section 5, we provide the results of several computational experiments that validate our theoretical results. Finally, we provide some concluding remarks in Section 6.

2 An Online Reward-collection Problem

We consider a mobile robotic vehicle that is tasked with visiting target locations. Due to differential constraints, it is impossible to visit all targets. The robot's mission is to visit these target locations as best as possible, measured by the amount of "reward" it collects per time unit during the course of the whole mission.

Targets are discovered on the fly, in the sense that the robot obtains the location and reward information associated with that target only when it gets sufficiently close to a target location. Hence, the robot does not know *a priori* all of the tasks and the reward associated with them. However, the statistics for the spatial distribution of the target locations and their reward is known, for example from past experience. To model this environment, we assume that the target locations and their rewards are generated by a stochastic process. The robot operates in this stochastic environment.

We formalize this online motion planning problem as follows. Consider a mobile robotic vehicle governed by the following ordinary differential equation:

$$\dot{x}(t) = f(x(t), u(t)) \tag{1}$$

where $x(t) \in X \subset \mathbb{R}^n$ is the state and $u(t) \in U \subset \mathbb{R}^m$ is the control input. A state trajectory $x : [0, T] \to Y$ is said to be a *dynamically-feasible trajectory*, if there exists $u : [0, T] \to U$ such that u, x satisfy Equation (1) for all $t \in [0, T]$.

Let $\mathcal{R}(\cdot)$ denote the reward function, which associates each dynamically-feasible trajectory, say $x : [0, T] \to Y$, with a reward denoted by $\mathcal{R}(x) \in \mathbb{R}$. The robot is tasked with finding a motion (*i.e.*, a dynamically-feasible trajectory) with maximal reward.

The reward function is not known *a priori*, but is revealed to the robot in an online manner. We formalize this aspect of the problem as follows. Let $\mathcal{P}(\cdot)$ denote the perception footprint of the robot that associates each state $z \in X$ of the robot with a footprint $\mathcal{P}(z) \subset X$. When the robot is in state $z \in X$, it is able to observe only the reward function associated with the partial trajectories within the set $\mathcal{P}(z)$. We assume that the reward function does *not* vary with time, and that the statistics of its distribution is known to the robot. The reward can only be collected once, so the robot keeps exploring new regions.

This general setting represents a large class of reward-collection problems. In this paper we study a special case that is more closely related to the motivational example presented in the previous section. Let $\mathcal{T} \subset X$ be a discrete set of target locations. Suppose each target $z \in \mathcal{T}$ is associated with a reward $r(z)$, and the robot collects the reward $r(z)$ if it visits the state z. That is, given a trajectory $x : [0, T] \to X$, its reward is $\mathcal{R}(x) := \sum_{z_i \in Z} r(z)$, where $Z := \{z \in \mathcal{T} : x(t) = z \text{ for some } t \in [0, T]\}$. The robot observes the locations and the rewards of all targets that fall within its perception footprint, and collects the associated reward if it visits a particular target location. It travels through this environment, discovering targets on the fly and adapting its trajectory to maximize the total reward it gathers by visiting these targets.

3 Lattice-based Motion Planning Algorithms

In such general setting, analytical solutions can be found only in some special cases. Yet, there are efficient computational approaches, based on the proper discretization of the set of all dynamically-feasible trajectories, that can achieve good performance. Below, we outline such an algorithm.

Lattice-based motion planning algorithms have long been successfully utilized in robotics applications [23, 24]. Roughly speaking, these algorithms form a directed lattice in the state space of the robot, and they select the best one among all paths through this lattice. This task is often computationally efficient, which makes the algorithm practical even in challenging problem instances. Below, we describe lattice-based planning algorithms in our notation.

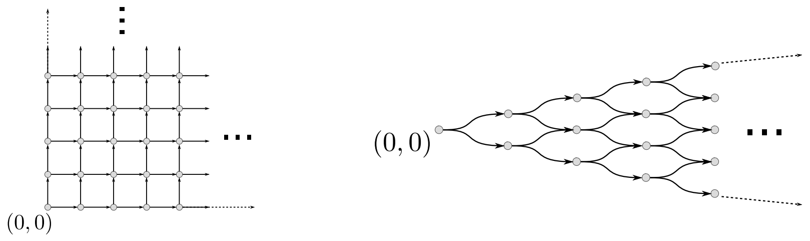
An infinite graph $G = (V, E)$, where V is a countable set of vertices and $E \subset V \times V$ is a set of edges, is said to be a *lattice*, if the following are satisfied:

(i) any vertex is a state of the dynamical system described by Equation (1), *i.e.*, $V \subset X$, and *(ii)* for any edge $e = (v_1, v_2) \in E$, there exists a dynamically feasible trajectory $x_e : [0, T_e] \to X$ such that $x(0) = v_1$ and $x(T_e) = v_2$.

A *lattice-based receding-horizon motion planning algorithm* works as follows. Initially, the robot is at a state $z_{\text{init}} \in V$. For each iteration, the best path $(e_1, e_2, \ldots, e_k)$ through the "visible" part of the lattice is computed, and the robot follows the dynamically-feasible trajectory $x_{e_1} : [0, T_{e_1}] \to X$ that is associated with the first edge on this path, denoted above by e_1. Once the robot reaches the state $v' = x_{e_1}(T_{e_1})$, the same procedure is repeated with the part of the lattice that is visible to the robot.

We formalize this algorithm below by first introducing some notation and a couple of sensing and actuation procedures that this algorithm utilizes. Let $G = (V, E)$ be a lattice for the robot governed by Equation (1). Two edges $e_1 = (v_1, v_1'), e_2 = (v_2, v_2') \in E$ are said to be *connected* if $v_1' = v_2$. A *path* through G is a sequence of edges, denoted by $\pi = (e_1, e_2, \ldots, e_k)$ such that $e_i \in E$ and e_i and e_{i+1} are connected for all $i \in \{1, 2, \ldots, k-1\}$. The ith edge on path π is denoted by $\pi(i)$. The set of all paths through G is denoted by $\texttt{Paths}(G)$. Given a path $p = (e_1, e_2, \ldots, e_k)$, let $x_{e_i} : [0, T_{e_i}] \to X$ denote the dynamically feasible trajectory attached to the edge e_i in the lattice $G = (V, E)$, and let $\texttt{Trajectory}(p)$ denote the dynamically-feasible trajectory formed by concatenating x_{e_i}'s, that is, $\texttt{Trajectory}(\texttt{p})$ is a dynamically-feasible trajectory $x_p : [0, T_p] \to X$, where $T_p = \sum_{i=1}^{k} T_{e_i}$ and $x_p(t) = x_{e_i}(t - \sum_{j=1}^{i} T_{e_j})$ for all $t \in \left[\sum_{j=1}^{i} T_{e_j}, \sum_{j=1}^{i+1} T_{e_j}\right]$ and all $i \in \{1, 2, \ldots, k-1\}$. Recall that X is the state space of the robot. Given a subset $P \subset X$ and a (potentially infinite) graph $G = (V, E)$, the projection of G on P is a new graph defined and denoted as follows: $\texttt{Projection}(G, P) := (V_P, E_P)$, where $V_P = V \cap P$ and $E_P = E \cap (V_P \times V_P)$.

Now, we define two procedures that allow the algorithm's perception and actuation. Let $\texttt{CurrentState}()$ be a procedure that returns the current state of the robot. Given a dynamically-feasible trajectory $x : [0, T] \to X$, let $\texttt{Execute}(x)$ denote the command that makes the robot follow the trajectory x.



(a) The two-dimensional lattice.

(b) A lattice with Dubins paths.

Fig. 2: The two-dimensional directed regular lattice, $\mathbb{N}^2$, is illustrated in Figure (a). An example lattice for a curvature constrained vehicle, also called the Dubins vehicle, is shown in Figure (b). The latter lattice can be embedded in $\mathbb{N}^2$.

Finally, we provide a formal description of the lattice-based receding-horizon motion planning procedure in Algorithm 1. The algorithm first retrieves the robot's current state (Line 2). Subsequently, it computes the portion of the lattice that falls within its sensor footprint (Line 3), and it then computes the optimal path through this sub-lattice (Line 4). Finally, the robot takes the trajectory corresponding to the first edge along the best path (Line 5), and it follows this path until it reaches the state that corresponds to the next vertex on the lattice (Line 6). This procedure continues for N iterations (Lines 1-7).

```
1 for $t = 1, \ldots, N$ do
2     $z \leftarrow$ CurrentState();
3     $G_P \leftarrow$ Projection($G, \mathcal{P}(z)$);
4     $\pi \leftarrow \arg\max\{\mathcal{R}($Trajectory$(\pi) : \pi \in$ Paths$(G_P))\}$;
5     $x_{\text{first}} \leftarrow$ Trajectory($\pi(1)$);
6     Execute($x_{\text{first}}$);
7 end
```

Algorithm 1: Lattice-based receding-horizon online motion planning

This algorithm computes the maximum-reward path on a graph. Note that this problem is NP-hard on a general graph [25]. However, for example, on acyclic graphs, this problem can be solved efficiently [25]. In this paper, we focus on the analysis of Algorithm 1 for acyclic lattices. This implies that, roughly speaking, the robot does not return to a place it has been before, hence it constantly explores new regions in the environment. An important acyclic graph is the d-dimensional directed regular lattice $L_d = (V, E)$, where $V = \mathbb{N}^d$ and $(v, v') \in E$ if $v = (v_1, v_2, \ldots, v_d)$ and $v' = (v_1, v_2, \ldots, v_k{+}1, \ldots, v_n)$ for some $k \in \{1, 2, \ldots, k\}$. The two-dimensional directed lattice is illustrated in Figure 2a. We say that a lattice $G = (V, E)$ is embedded in $\mathbb{N}^d$ if it is isomorphic to $\mathbb{N}^d$. In Section 4.2, we pay special attention to the two-dimensional lattice $\mathbb{N}^2$. In Figure 2b, we provide an example 2 dimensional lattice for a non-holonomic vehicle.

4 Analysis of Online Motion Planning Algorithms

This section is devoted to the analysis of the lattice-based online planning algorithm. This analysis sheds light on the relationship between the perception capabilities of the robot and its performance. Specifically, we consider the following questions: How much reward can the robot collect given a certain perception range? How does this reward compare with the fundamental limit when the robot can observe the entire reward field *a priori* and compute the best path?

4.1 On Perception Range versus Performance

Before presenting the main theoretical result of this section, let us provide some notation. Let $G = (V, E)$ be an acyclic graph with infinitely many vertices. Recall that the set of all paths in $G = (V, E)$ is denoted by Paths(G). Given a path $\pi \in$ Paths(G), let $|\pi|$ denote the length of π measured by the number of vertices that π visits. Let $\mathit{\Pi}(v_{\text{init}}, n)$ denote the set of all paths that start from the vertex

v_{init} and cross at most n vertices. Suppose each vertex $v \in V$ is associated with a reward denoted by $\rho(v)$. Let $R(v_{\text{init}}, n)$ denote the total reward collected by following some path that starts from v_{init} and crosses at most n vertices, *i.e.*,

$$R(v_{\text{init}}, n) = \max_{\pi \in \Pi(v_{\text{init}}, n)} \sum_{v \in \pi} \rho(v).$$

Note that $R(v_{\text{init}}, n)$ is the maximum reward that the robot can collect in n steps if it could see the whole environment, not the reward collected with limited perception range.

The perception range limitation allows the robot to observe the reward associated with only a subset of the vertices. Let m be a number such that any vertex that can be reached with a path of length m is within the perception range independently of the starting vertex, *i.e.*, m is such that, for all $v_{\text{init}} \in V$, any state in $\{v \in \pi : \pi \in \Pi(v_{\text{init}}, m)\}$ is within the perception range of the robot when the robot is at state v_{init}.

Let $Q(v_{\text{init}}, n; m)$ denote the reward that is achieved as follows. Let R_1 denote the reward that can be collected by a path that starts from v_{init} and has length m, *i.e.*, $R_1 := R(v_{\text{init}}, m)$. Let v_1 denote the vertex that the maximum-reward path (achieving reward R_1) ends at. Similarly, define $R_k := R(v_{k-1}, m)$, and let v_k be the vertex that path achieving reward R_k ends at. Finally, define

$$Q(v_{\text{init}}, n; m) := \sum_{i=1}^{n/m} R_i$$

Compare this quantity with the reward that Algorithm 1 can achieve. Notice that Algorithm 1 considers a larger set of paths each time it computes a maximum-reward path through the observable part of the lattice. Moreover, Algorithm 1 computes a new path right after the first edge along the path is executed. In contrast, the computation of $Q(v_{\text{init}}, n; m)$ considers only those paths that are of distance m, and moreover it only computes a new path after the current one is fully executed. Given these observations, we expect the reward achieved by Algorithm 1 to be at least $Q(v_{\text{init}}, n; m)$. In other words, $Q(v_{\text{init}}, n; m)$ is a lower bound for the reward that Algorithm 1 can collect, when measured in terms of suitable statistics, such as the expectation. Although this statement can be properly formalized, we omit this formalism due to space limitations.

Now we focus on the analysis of $Q(v_{\text{init}}, n; m)$. In particular we compare $Q(v_{\text{init}}, n; m)$ and $R(v_{\text{init}}, n)$. The former is the reward that the robot can collect with limited perception range m. The latter is the reward that the robot can collect if it had infinite perception range. In what follows, the initial vertex v_{init} is fixed, and it is the same for all results reported below. For simplicity, we drop v_{init} from our notation, and we write $Q(n; m)$ and $R(n)$ in the sequel.

Our first result allows us to define the mean reward.

Proposition 1. *The following holds:*

$$\lim_{n \to \infty} \frac{\mathbb{E}[R(n)]}{n} = \sup_{n \in \mathbb{N}} \frac{\mathbb{E}[R(n)]}{n}.$$

Proof. The result follows directly from Fekete's lemma [26], noting that the sequence $\mathbb{E}[R(n)]$ is superadditive, hence $-\mathbb{E}[R(n)]/n$ is subadditive. $\square$

Let's define the mean reward per step as $R^* := \lim_{n\to\infty} \mathbb{E}[R(n)]/n$, which is well defined by Proposition 1. We compare R^* with $Q(n;m)/n$ for suitable values of m.

Theorem 1. *Suppose R^* is finite. Suppose that the rewards $\rho(v)$ are independent (but not necessarily identically distributed) and that they are uniformly almost-surely bounded random variables,* i.e., *there exists some L such that $\mathbb{P}(|\rho(v)| \leq L) = 1$, for all $v \in V$. Then, for any $\delta > 0$, there exists a constant c such that*

$$\lim_{n\to\infty} \mathbb{P}\Big(\Big| \frac{Q(n, c\log n)}{n} - R^* \Big| \geq \delta \Big) \; = \; 0.$$

Roughly speaking, Theorem 1 implies that the robot can navigate to any vertex that is at most n steps away almost optimally (as if it had infinite perception range), if its perception range is at order $\log n$. In other words, as the perception range increases, the amount of distance that the robot can travel optimally increases exponentially fast, as stated below.

Corollary 1. *Suppose the assumptions of Theorem 1 hold. Then, for any $\delta > 0$, there exists some constant c such that*

$$\lim_{m\to\infty} \mathbb{P}\Big(\Big| \frac{Q(L(m), m)}{L(m)} - R^* \Big| \geq \delta \Big) \; = \; 0,$$

where $L(m) = e^{c\,m}$ for some constant c that is independent of m (but depends on δ).

This corollary follows from Theorem 1 with a change of variables.

Before proving Theorem 1, we state an intermediate result that enables our proof. This intermediate result is a concentration inequality, which plays a key role in deriving many results in nonequilibrium statistical mechanics [20].

Lemma 1 (See [20]). *Let $\{Y_i, i \in \mathcal{I}\}$ be a finite collection of independent random variables that are bounded almost surely,* i.e., *$\mathbb{P}(|Y_i| \leq L) = 1$ for all $i \in \mathcal{I}$. Let $\mathcal{C}$ be a collection of subsets of $\mathcal{I}$ with maximum cardinality R,* i.e., *$\max_{C\in\mathcal{C}} |C| \leq R$ and let $Z = \max_{C\in\mathcal{C}} \sum_{i\in C} Y_i$. Then for any $u > 0$,*

$$\mathbb{P}(|Z - \mathbb{E}Z| \geq u) \leq \exp\bigg(-\frac{u^2}{64RL^2} + 64\bigg).$$

Finally, we present the proof for Theorem 1.

Proof (Theorem 1). Let $\mathcal{I}$ be the collection of nodes in the lattice. Define $\mathcal{C} = \{\mathcal{N}(\pi), \pi \in \mathit{\Pi}\}$, where $\mathcal{N}(\pi) = \{\mathbf{v} \in \pi\}$ is the set of nodes in the path π. Then, for the maximum-reward path with at most n steps, the maximum cardinality is $\max_{C\in\mathcal{C}} |C| \leq n$. Then, by substituting $R(n)$ for Z in Lemma 1,

$$P(|R(n) - \mathbb{E}R(n)| \geq u) \;\leq\; \exp\Big(-\frac{u^2}{64nL^2} + 64\Big).$$

Therefore, for any $\delta = \frac{u}{n} > 0$,

$$\begin{aligned}
&\mathbb{P}(\big|\frac{Q(n, c\,\log n)}{n} - \frac{\mathbb{E}R(n)}{n}\big| \geq \delta) \\
&=\mathbb{P}(\big|\frac{\sum_{i=1}^{\frac{n}{m}} R_i(m)}{n} - \frac{\mathbb{E}R(n)}{n}\big| \geq \delta) \\
&=\mathbb{P}(\big|\frac{\sum_{i=1}^{\frac{n}{m}} R_i(m)}{n} - \frac{\mathbb{E}R(m)}{m} + \frac{\mathbb{E}R(m)}{m} - \frac{\mathbb{E}R(n)}{n}\big| \geq \delta) \qquad (2) \\
&\leq\mathbb{P}(\{\big|\frac{\sum_{i=1}^{\frac{n}{m}} R_i(m)}{n} - \frac{\mathbb{E}R(m)}{m}\big| \geq \frac{\delta}{2})\}\bigcup\{\frac{\mathbb{E}R(m)}{m} - \frac{\mathbb{E}R(n)}{n}\big| \geq \frac{\delta}{2}\}), \qquad (3) \\
&\leq\mathbb{P}(\big|\frac{\sum_{i=1}^{\frac{n}{m}} R_i(m)}{n} - \frac{\mathbb{E}R(m)}{m}\big| \geq \frac{\delta}{2}) + \mathbb{P}(\frac{\mathbb{E}R(m)}{m} - \frac{\mathbb{E}R(n)}{n}\big| \geq \frac{\delta}{2}) \qquad (4) \\
&=\mathbb{P}(\sum_{i=1}^{\frac{n}{m}} \big|R_i(m) - \mathbb{E}R(m)\big| \geq \frac{n\delta}{2}) + \mathbb{P}(\frac{\mathbb{E}R(m)}{m} - \frac{\mathbb{E}R(n)}{n}\big| \geq \frac{\delta}{2}). \qquad (5)
\end{aligned}$$

The inequality between line (2) and line (3) can be seen if we take the complements on both sides, where $\{\big|(\frac{\sum_{i=1}^{\frac{n}{m}} R_i(m)}{n} - \frac{\mathbb{E}R(m)}{m}) + (\frac{\mathbb{E}R(m)}{m} - \frac{\mathbb{E}R(n)}{n})\big| < \delta\} \supset \{\big|\frac{\sum_{i=1}^{\frac{n}{m}} R_i(m)}{n} - \frac{\mathbb{E}R(m)}{m}\big| < \frac{\delta}{2}\}\bigcap\{\big|\frac{\mathbb{E}R(m)}{m} - \frac{\mathbb{E}R(n)}{n}\big| < \frac{\delta}{2}\}$. Union bound is applied between between line (3) and line (4). Now we set $m = c\log n$. Taking limit on both sides, we get

$$\begin{aligned}
&\lim_{n\to\infty} \mathbb{P}(\big|\frac{Q(n, c\,\log n)}{n} - \frac{\mathbb{E}R(n)}{n}\big| \geq \delta) \\
&\leq \lim_{n\to\infty} \mathbb{P}(\sum_{i=1}^{\frac{n}{m}} \big|R_i(m) - \mathbb{E}R(m)\big| \geq \frac{n\delta}{2}) + \lim_{n\to\infty} \mathbb{P}(\frac{\mathbb{E}R(m)}{m} - \frac{\mathbb{E}R(n)}{n}\big| \geq \frac{\delta}{2}) \qquad (6) \\
&\leq \lim_{n\to\infty} \mathbb{P}(\sum_{i=1}^{\frac{n}{m}} \big|R_i(m) - \mathbb{E}R(m)\big| \geq \frac{n\delta}{2}) + 0 \qquad (7) \\
&\leq \lim_{n\to\infty} \sum_{i=1}^{\frac{n}{m}} \mathbb{P}(|R_i(m) - \mathbb{E}R(m)| \geq \frac{m\delta}{2}) \qquad (8) \\
&\leq \lim_{n\to\infty} \frac{n}{m} \cdot \exp(-\frac{(\frac{m\delta}{2})^2}{64mL^2} + 64) \qquad (9) \\
&= \lim_{n\to\infty} \frac{1}{c\,\log n} \cdot \exp\big((1 - \frac{\delta^2}{256L^2} \cdot c)\log n + 64\big) \qquad (10)
\end{aligned}$$

The first inequality comes from line (5). The inequality between line (6) and line (7) is due to Proposition 1. As n increases, $m \to \infty$, and thus both $\frac{\mathbb{E}R(m)}{m}$ and $\frac{\mathbb{E}R(n)}{n}$ converge to the same constant R^*. Union bound is again applied between between line (7) and line (8). Lemma 1 is applied in line (9). Line(10) converges to 0 when the constant c is sufficiently large, *i.e.*, for any constant $c \geq \frac{256L^2}{\delta^2}$.

□

Although the conclusion in Theorem 1 is exciting, it may be too restricted for real-world applications, as it requires both independence among $\rho(v)$ and the random variables $\rho(v)$ to be bounded almost surely. We propose a conjecture that generalizes Theorem 1 by relaxing these assumptions. We believe that the reward collected by a robot is arbitrarily close to optimal (as formalized by Theorem 1), even if the reward is locally dependent (instead of independent) and the distribution of reward has light tails (instead of being bounded). Local dependence refers to the case when $\rho(v_1)$ is conditionally independent of all other rewards $\rho(v_2)$ given the local neighborhood of v_1, if v_2 is not a neighbor.

Conjecture 1. Assume that the non-negative reward $\rho(\mathbf{z})$ at each state $\mathbf{z}$ is distributed with local dependence and that their distributions satisfy:

$$\int_0^\infty (1 - F(x))^{1/d} dx < \infty.$$

Then, for any $\delta > 0$, there exists a constant c such that

$$\lim_{n\to\infty} \mathbb{P}\Big(\Big| \frac{Q(n, c \log n)}{n} - R^* \Big| \geq \delta \Big) \ = \ 0.$$

Notice that the assumptions of this conjecture are much weaker. We leave the proof of this conjecture as a future work. The rationale behind this conjecture is two-fold. Firstly, the relaxation on the boundedness of F is supported in a recent text [16], where the sketch of the proof is given and the details are left as an exercise. Secondly, the relaxation of the independence requirement is inspired by [17], where the Hoeffding inequality for independent random variables can be extended to the local dependent case with only slight modifications.

4.2 Special Case: Planning on the Directed Regular Lattice

The result in Theorem 1 includes two constants, namely the mean reward R^* and the constant c, the precise values of which are not known. In this section, we prove various results to characterize these constants when the lattice $G = (V, E)$ can be embedded in $\mathbb{N}^d$, particularly in the case when $d = 2$. Throughout this section, we tacitly assume that $G = (V, E)$ is embedded in $\mathbb{N}^d$. Unless stated otherwise, our results hold for all values of d satisfying $d \geq 2$.

First, let us define some useful notation. The vertices of $\mathbb{N}^d$ are denoted by $\mathbf{w} = (w_1, w_2, \ldots, w_d)$, where $w_k \in \mathbb{N}$. We define $\lfloor k\, \mathbf{w} \rfloor := (\lfloor k\, w_1 \rfloor, \lfloor k\, w_2 \rfloor, \ldots, \lfloor k\, w_d \rfloor)$. Given a vertex $\mathbf{w} \in \mathbb{N}^d$, let $T(\mathbf{w})$ denote the reward of the maximum reward path that starts from the origin and reaches the vertex $\mathbf{w}$. With a slight abuse of notation let $\Pi(\mathbf{w})$ denote the set of all paths that start from the origin and end at the vertex v. Then,

$$T(\mathbf{w}) := \max_{\pi \in \Pi(\mathbf{w})} \sum_{\mathbf{w}' \in \pi} \rho(\mathbf{w}').$$

On Almost-sure Convergence of the Reward: Let us point out an existing result that shows the convergence result in Proposition 1 can be improved.

Proposition 2 (See Proposition 2.1 in [21]). *Assume* $\mathbb{E}[\rho(\mathbf{w})] < \infty$. *Define*

$$g(\mathbf{w}) := \sup_{k \in \mathbb{N}} \frac{\mathbb{E}[T(\lfloor k\,\mathbf{w} \rfloor)]}{k}.$$

Then, $\frac{T(\lfloor n\,\mathbf{w} \rfloor)}{n}$ *converges to* $g(\mathbf{w})$ *almost surely as* n *diverges to infinity,* i.e.,

$$\mathbb{P}\Big(\lim_{n \to \infty} \frac{T(\lfloor n\,\mathbf{w} \rfloor)}{n} = g(\mathbf{w}) \Big) = 1$$

This result implies that R^* is finite and that $R(n)/n$ converges to R^* as $n \to \infty$, almost surely.

On Mean Reward: Suppose the dimensionality of the lattice is two, *i.e.*, $d = 2$. Let F denote the distribution for i.i.d. random variables $\rho(\mathbf{w})$. Then, the results in [21] imply that there are two cases for which R^* can be computed exactly, namely when F is an exponential distribution or a geometric distribution. More specifically, if F is the exponential distribution with parameter $\lambda = 1$, then

$$g\big((x, y)\big) = (\sqrt{x} + \sqrt{y})^2, \qquad \text{for all } (x, y) \in \mathbb{N}^2. \tag{11}$$

On the other hand, if F is the geometric distribution with parameter p, then

$$g\big((x, y)\big) = \frac{x + 2\sqrt{xy(1-p)} + y}{p}, \qquad \text{for all } (x, y) \in \mathbb{N}^2.$$

According to a recent survey paper [22], Timo Sepplinen conjectured that the function for general distributions F with mean μ and variance σ^2 is

$$g\big((x, y)\big) = \mu(x + y) + 2\sqrt{\sigma^2 xy}, \qquad \text{for all } (x, y) \in \mathbb{N}^2.$$

It is noted that a rigorous proof is beyond the reach of the mathematical statistical physics community at this stage. However, if this conjecture holds, it has an important implication for the problem considered in this paper. By the definition of $R(n)$ and Proposition 2, the reward per step, $R(n)/n$ converges to $\mu + \sigma$ as the distance that the vehicle travels increases to infinity, almost surely, *i.e.*.

$$\mathbb{P}\Big(\lim_{n \to \infty} \frac{R(n)}{n} = \mu + \sigma \Big) = 1.$$

under mild technical assumptions. Hence, in this case, $R^* = \mu + \sigma$. Simulation results supporting this conjecture are shown in the next section.

On the Fluctuations of the Reward: Now, we shift our attention to the constant c in the statement of Theorem 1. Recall that this constant is independent of n; however, it depends on δ. We characterize how this constant depends on δ by employing results from the nonequilibrium statistical mechanics literature. This investigation is possible by utilizing more accurate characterizations of the function $T(\cdot)$. It is shown in [19] that, for the aforementioned two cases,

$$\frac{T\big(\,(\lfloor xn\rfloor,\lfloor yn\rfloor)\,\big)-n\,g\big(\,(x,y)\,\big)}{n^{\frac{1}{3}}}\to F_2$$

as n goes to infinity, where F_2 is the Tracy-Widom distribution.

In this case, we find that $c=\kappa\,\delta^{3/2}$ for some constant κ that is independent of δ and n, as stated below.

Theorem 2. *Suppose the lattice $G=(V,E)$ is embedded in $\mathbb{N}^2$ and $\rho(\mathbf{w})$ are independent identically distributed random variables. Suppose their common distribution is either the exponential distribution or geometric distribution. Then,*

$$\lim_{m\to\infty}\mathbb{P}\Big(\Big|\frac{Q(L(m);m)}{m}-R^*\Big|\geq\delta\Big)=0,$$

where $L(m)=\exp(\kappa\,\delta^{3/2}\,m)$, for some constant $\kappa>0$ independent of m and δ.

Before proving the theorem, let us compare it with Corollary 1. While Corollary 1 characterizes the reward with respect to perception range m, Theorem 2 also identifies its dependence on the error term δ. A natural conjecture is that the result of Theorem 2 holds for any distribution with finite variance. In the next section, we present simulation results that support this conjecture.

The proof of Theorem 2 is similar to that of Theorem 1. We omit the full proof; but we outline the main differences.

Proof. Let TW be a random variable with the Tracy-Widom distribution. Then, the results in [19] imply the following: For all $u\geq 0$,

$$\begin{aligned}\mathbb{P}(TW\geq u)&=\lim_{n\to\infty}\mathbb{P}\Big(\frac{R(n)-nR^*}{n^{1/3}}\;\geq\;u\Big)\\&=\lim_{n\to\infty}\mathbb{P}\Big(n^{2/3}\,(R(n)/n-R^*)\;\geq\;u\Big)\\&=\lim_{n\to\infty}\mathbb{P}\Big(\Big(\frac{R(n)}{n}-R^*\Big)\;\geq\;u\,n^{-2/3}\Big)\end{aligned}$$

Define $\delta:=u\,n^{-2/3}$. Hence, $u=\delta\,n^{2/3}$. It was showed very recently [27] that the right tail of the Tracy-Widom distribution F_2 can be characterized as follows:

$$\lim_{u\to\infty}\mathbb{P}(TW\geq u)=\alpha\exp\left(-\frac{4}{3}\,u^{3/2}\right).$$

Combining this with the previous equality, we obtain:

$$\lim_{n\to\infty}\mathbb{P}\left(\frac{R(n)}{n}-R^*\geq\delta\right)=\alpha\exp\left(-\frac{4}{3}\,\left(\delta\,n^{2/3}\right)^{3/2}\right)=\alpha\exp\left(-\frac{4}{3}\,\delta^{3/2}\,n\right).$$

The rest of the proof follows the proof of Theorem 1. □

5 Computational Experiments

In this section we provide simulations to support our analysis. There are two major results. The first simulation shows how the speed of reward collection converges as the distance n that the vehicle travels increases. The second simulation visualizes how the expected distance that the robot can travel without losing too much reward changes, as we increase the perception range.

5.1 Mean of Reward per Step

According to Equation (11), when F is an exponential distribution with $\lambda = 1$, the limit of reward per step is known. In this experiment, we create a 2-dimensional matrix where each element $\mathbf{w} = (x, y)$ inside the matrix corresponds to the reward $\rho(\mathbf{w})$. The i.i.d random variables $\rho(\mathbf{w})$ follow an exponential distribution with mean 1. We find the maximal reward $T(\mathbf{z})$, from the origin to a set of locations $\{\mathbf{z} = (x, y) | x + y = n\}$, using dynamic programming. This process is repeated 1000 times to compute the empirical average. The result is shown in Figure 3. A similar result can also be found for the geometric distribution.

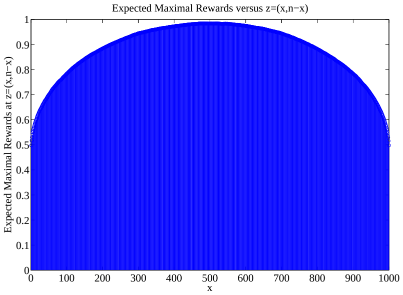


Fig. 3: This plot shows the relationship between the reward per step, $T(x, n - x)/n$, and the x-coordinate of the final destination of the robot. It verifies that the lemma $g(x, y) = (\sqrt{x} + \sqrt{y})^2$ for exponential distribution is correct.

Recall the conjecture that $g(x, y) = \mu(x + y) + 2\sqrt{\sigma^2 xy}$, which implies that $\frac{R(n)}{n} \to \mu + \sigma$ almost surely as $n \to \infty$. As mentioned earlier, a rigorous proof is not known yet, but we show some simulation examples that support this conjecture. We set up an experiment which is very similar to the previous one, except that the distribution F of the random variables $\rho(\mathbf{w})$ is a Poisson distribution with $\lambda = 0.05$. The Poisson distribution is interesting because it is closely related with the example we provide in Section 1, when the sensing devices are dispersed according to a Poisson distribution.

In Figure 4, we plot the relationship between $\frac{R(n)}{n}$ and n. In this scenario, $\mu + \sigma = \lambda + \sqrt{\lambda} = 0.05 + \sqrt{0.05} = 0.2736$. From the graph we can see that $\frac{R(n)}{n}$ is converging towards this value.

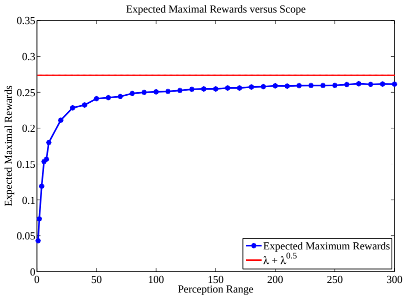


Fig. 4: This plot shows the relationship between the reward per step, $\frac{R(n)}{n}$, and the travel distance n, for Poisson distribution with $\lambda = 0.05$. It supports the conjecture that $\frac{R(n)}{n} \to \mu + \sigma$ as $n \to \infty$ for general distributions.

5.2 Receding Horizon

In Section 4.2 we have shown that with a perception range of $m = O(\log n)$, the robot will be able to collect almost as many rewards as the optimal case (with full information). Based on Theorem 2, we claim that for any fixed $\delta > 0$, if we know that the optimal reward per step is R^*, and we if would like to keep a reward per step of no less than $R^* - \delta$ with a fixed perception range m, then the expected maximal travel distance of the robot is of order $L(m) = \exp(\kappa \cdot \delta^{1.5} \cdot m)$ for some constant $\kappa > 0$ that depends only on the distribution F.

Consider the following computational experiment. Suppose the robot is running Algorithm 1 with a lattice that is embedded in $\mathbb{N}^2$. We run this simulation until the reward per step falls below $R^* - \delta$, and we measure the distance that the robot has travelled before the simulation stops. We repeat this experiment 1000 times on different realizations of the random variables. We average the distance that the robot travels to compute an empirical average. In Figures 5, we show the relationship between the distance that the robot travels and the perception range m for geometric distribution with $p = 0.5$ and and different values of δ.

The simulation results show that this distance increases exponentially, in fact obeying the order $\exp(\kappa\, \delta^{1.5}\, m)$, where κ is around 0.4 in this case. These simulation results support the result of Theorem 2.

6 Conclusions

We analyze the maximum-reward paths computed by a simple, practical receding-horizon online motion planning algorithm. In particular, we show that the distance that the robot can travel almost optimally increases exponentially fast with increasing perception range. We also characterize the exponent in terms of the error term. Along the way, we establish novel connections between a class of path planning problems and certain fundamental problems of non-equilibrium statistical mechanics, which may be interesting on their own right.

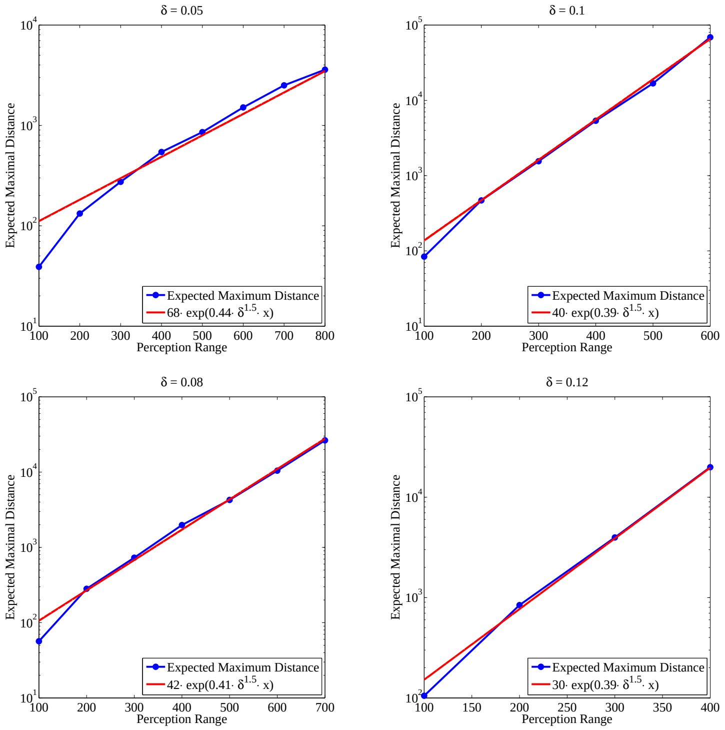


Fig. 5: This plot shows the relationship between the distance a robot can travel (with reward per step no less than $R^* - \delta$ for geometric distribution with $p = 0.5$) and different values of δ. Note that the y-axis is semi-log, which indicates that as perception range increases, the distance a robot can travel without too much loss in reward increases exponentially. The red line is an approximation of the real data. Notice that the exponent is around $0.4 \cdot \delta^{1.5}$, which supports Theorem 2.

Future work includes the construction of a rigorous proof for the main conjecture of the paper (given in Conjecture 1). We will study how the maximum reward scales with other perception capabilities, such as perception uncertainty, as well as with the actuation and on-board computation capabilities of the robot.

References

1. C V Heer. *Statistical Mechanics, Kinetic theory, and Stochastic Processes.* Academic Press, 2012.
2. G. F. Mazonko. *Nonequilibrium Statistical Mechanics.* Wiley, 2008.
3. T Chou, K Mallick, and RKP Zia. Non-equilibrium statistical mechanics: from a paradigmatic model to biological transport. *Reports on Progress in Physics*, 2011.

4. Leah B Shaw, R K P Zia, and Kelvin H Lee. Totally asymmetric exclusion process with extended objects: A model for protein synthesis. *Physical Review E*, 68(2):021910, August 2003.
5. L Ingber. Statistical Mechanics of Nonlinear Nonequilibrium Financial Markets. *Mathematical Modeling*, 5:343–361, January 1984.
6. T Antal and G M Schutz. Asymmetric exclusion process with next-nearest-neighbor interaction: Some comments on traffic flow and a nonequilibrium reentrance transition. *Physical Review E*, 62(1):83–93, 2000.
7. T Nagatani. Bunching of cars in asymmetric exclusion models for freeway traffic. *Physical Review E*, 51(2):922–928, 1995.
8. G R Fleming and M A Ratner. Grand challenges in basic energy sciences. *Physics Today*, 61(7):28, 2008.
9. Basic Energy Sciences Advisory Committee. Directing Matter and Energy: Five Challanges for Science and Imagination. pages 1–144, November 2007.
10. National Research Council Committee on CMMP 2010. *Condensed-Matter and Materials Physics: The Science of World Around Us.* January 2010.
11. Michael Otte, Nikolaus Correll, and Emilio Frazzoli. Navigation with foraging. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3150–3157. IEEE, 2013.
12. S. Karaman and E. Frazzoli. High-speed Flight in an Ergodic Forest. In *IEEE International Conference on Robotics and Automation*, 2011.
13. Antoine Beyeler, Jean-Christophe Zufferey, and Dario Floreano. Vision-based control of near-obstacle flight. *Autonomous Robots*, 27(3):201–219, August 2009.
14. S Scherer, S Singh, L Chamberlain, and M Elgersma. Flying Fast and Low Among Obstacles: Methodology and Experiments. *The International Journal of Robotics Research*, 27(5):549–574, May 2008.
15. D J Bertsimas. A Stochastic and Dynamic Vehicle Routing Problem in the Euclidean Plane. *Operations Research*, 39(4):601–615, February 2008.
16. Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence.* Oxford University Press, 2013.
17. Devdatt P Dubhashi and Alessandro Panconesi. *Concentration of measure for the analysis of randomized algorithms.* Cambridge University Press, 2009.
18. Laure Dumaz and Bálint Virág. The right tail exponent of the tracy-widom-beta distribution. *arXiv preprint arXiv:1102.4818*, 2011.
19. Kurt Johansson. Shape fluctuations and random matrices. *Communications in mathematical physics*, 209(2):437–476, 2000.
20. James B Martin. Linear growth for greedy lattice animals. *Stochastic Processes and their Applications*, 98(1):43–66, 2002.
21. James B Martin. Last-passage percolation with general weight distribution. *Markov Process. Related Fields*, 12(2):273–299, 2006.
22. Xingyuan Zeng, Zhenting Hou, Chunxiao Guo, and Yanfeng Guo. Directed last-passage percolation and random matrices. *Advances in Mathematics*, 42(3):3, 2013.
23. C. Urmson et al. Autonomous driving in urban environments: Boss and the Urban Challenge. *Journal of Field Robotics*, 25(8):425–466, 2008.
24. S Koenig, M Likhachev, and D Furcy. Lifelong planning A∗. *Artificial Intelligence*, 2004.
25. A. Schrijver. *Combinatorial optimization.* Springer Verlag, 2003.
26. M J Steele. *Probability Theory and Combinatorial Optimization.* SIAM, 1996.
27. Laure Dumaz and Bálint Virág. The right tail exponent of the Tracy–Widom β distribution. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 49(4):915–933, November 2013.