Simplicial Label Correcting Algorithms for Continuous Stochastic Shortest Path Problems

Dmitry S. Yershov
Department of Computer Science
University of Illinois at Urbana-Champaign,
Urbana, IL 61801,
Email: yershov2@uiuc.edu

Steven M. LaValle
Department of Computer Science
University of Illinois at Urbana-Champaign,
Urbana, IL 61801,
Email: lavalle@uiuc.edu

Abstract—The problem of optimal feedback planning under prediction uncertainties among static obstacles is considered. A discrete-time stochastic state transition model is defined over a continuous state space. A relation to a "nearby" deterministic model is shown for small time steps; the cost-to-go function of the stochastic model is approximated with that of the deterministic model, and the approximation error is found to be proportional to the time step. This motivates using numerical methods, which are vastly available for solving deterministic problems, to approximate the original stochastic problem. We demonstrate this application on a Simplicial Label Correcting Algorithm, which, assuming a piecewise linear discretization, computes the shortest-path plan on the simplicial complex. Additionally, a theoretical error bound between the approximate solution and the exact solution is derived and confirmed in numerical experiments. This paper provides a rigorous analysis as well as algorithmic and implementation details of the proposed model for the stochastic shortest path problem in continuous spaces with obstacles.

I. INTRODUCTION

Recently, interest in the theory of stochastic control with applications in robotics has increased significantly. Many aspects contribute to the success of this theory: 1) it predicts the system performance more accurately compared to deterministic control theory, 2) techniques and results from the deterministic models are readily applicable to stochastic models, and 3) unknown environments and incomplete, noisy sensing are naturally incorporated into the stochastic framework, whereas deterministic models always require perfect observations and the knowledge of the environment. Thus, stochastic control models, as a generalization of their deterministic counterparts, are the state of the art in the field robotics community.

Stochastic control models are of four types: Type 1 – discrete space and discrete time; Type 2 – discrete space and continuous time; Type 3 – continuous space discrete time; and Type 4 – continuous space and time. The general form of Type 1, 2, and 3 models is

$$x_{i+1} = f(x_i, u_i, \omega_i) , \qquad (1)$$

in which x_i is a state of a system, u_i is a control signal, and ω_i is a random disturbance parameter, all of which are taken at time t_i. In the case of Type 1 and 3 models, $\{t_i\}$ is a fixed time sequence. For models of Type 2, however, $\{t_i\}$ is a Poisson process, and, hence, each t_i is a random

variable. Finally, models of Type 4 is usually described using the Stochastic Differential Equation (SDE) of the form:

$$\mathrm{d}x(t) = f(x(t), u(t)) \, \mathrm{d}t + \mathrm{d}\omega(t) , \qquad (2)$$

in which $\omega(t)$ is a Wiener process.

Models of Types 1 and 2 are extensively studied in the existing literature, including textbooks [7], [20]. Optimal feedback control is derived using Bellman's dynamic programming principle, the result of which is the Hamilton-Jacobi-Bellman (HJB) equation for the optimal cost-to-go function [4]. The value iteration and the policy iteration algorithms, which solve the resulting HJB equation, were introduced in [4] and later analyzed for the convergence and the accuracy in [22], [31]. For the Shortest Path Problem (SPP), a family of Label Correcting Algorithms (LCA), including the special case of Dijkstra's algorithm, were introduced in [12] and discussed further in [6], [13]. In the case of consistently improving policies, these algorithms are guaranteed to terminate in finite time. Moreover, the total running time of LCA can be tweaked using different queuing strategies. Finally, Foster-Lyapunov analysis is developed in [21] to bound the performance of the stochastic systems given a stabilizing policy. Despite significant developments in the theory and the implementation for models of Types 1 and 2, they do not apply directly to continuous-space systems. Hence, for the problem of optimal navigation these models are useful after the discretization step only [5], [11], [14], [18].

Models of Type 4, on the other hand, are continuous in space and time and are appealing for applications in robotics. Originally, these models were developed in the area of physics studying non-equilibrium statistical mechanics. In that field the Brownian motion of a microscopic particle under the influence of a great number of other particles with comparable momenta is of particular interest. Later, the theory of SDE was extended to other scientific disciplines to study general systems under influence of a significantly large amount of random uncorrelated force sources (noise): radio telecommunication channels influenced by the cosmic radiation originating from hundreds of billions of stars, electrical grids influenced by turbulent solar winds, stock markets influenced by thousands of agents, and many others. Applications of SDE are also found in robotics,

including the optimal navigation under diffusive models of motion [16], [17] and the dead-reckoning problem [33].

In many applications of robotics the macroscopic influence of a great number of random and uncorrelated force sources is negligible, and, thus, the behavior of macroscopic mechanical systems is far from diffusive. The following simple experiment demonstrates this: We consider (2) and assume $f(x, u) = u$ for simplicity. Let $\tilde{u}(t) = \delta u$, for some vector u and a real, positive parameter δ, be an input signal applied to the system during the time period $[0, \delta^{-1}]$. Starting at the origin and executing $\tilde{u}(t)$, the ideal noiseless system arrives at u. Moreover, in this case, the motion is scale invariant, that is, no matter how small δ is, the final point remains the same. If noise is present, however, then the smaller the δ the larger the spread of the final robot position. This is due to the additive nature of noise, which does not scale with the velocity. At the limit of very small velocities, the robot "diffuses" indefinitely in space. Nevertheless, in the real world scenario, a toy car in a backyard with an open loop control on uneven terrain under strong wind conditions arrives near the desired destination with a bounded spread, which is independent (within reasonable physical limits) of its velocity. This experiment demonstrates that the level of noise depends on the parameters of the system and, therefore, cannot be additive. Additionally, stochastic diffusion models suffer from another drawback: the probability of collision with obstacles is nonzero after infinitesimally short time regardless the control input. Thus, to formulate the shortest path problem either the probability of success is restricted to a given level of tolerance or "soft" obstacles are introduced, otherwise the cost function is always infinite.

To resolve the problem of an unbounded spread, random disturbances must be introduced into the motion model in a parametric form. Poisson arrival process is a good candidate to model the change of the disturbance parameters. Using this treatment, we model rough terrains using controlled jump process models [9]. In this paper, we simplify the Poisson arrival process using a fixed time step process and demonstrate that the resulting system is of Type 3, rather than Type 4. In this setting, the spread remains bounded in finite time, and it is possible to formulate the SPP with 100% goal arrival probability and infinite penalty for collision. Thus, borrowed from physics Type 4 models, although popular in robotics, are most applicable for stochastic diffusion phenomena, and not the motion of macroscopic mechanical systems. On the other hand, a model of Type 3 is good candidate to describe the motion of robots under random disturbances.

In the existing literature, only special cases of Type 3 models are studied, in which the optimal control problem can be solved exactly. For example, the linear systems quadratic cost model with additive Gaussian uncertainty (LQG) is solved using the algebraic Riccati equation [2]. The LQG model is widely used in robotics for system stabilization either at the origin or along a precomputed trajectory provided by a motion planning algorithm [1], [10], [25], [30]. However, this model excludes obstacles and the precomputed path must be optimized beforehand to minimize the probability of collisions.

Moreover, the exact solution may not exist for nonlinear systems, nonquadratic cost functionals, or if obstacles are present. Hence, models of Type 3 remains mostly unstudied.

In this paper, we use a Type 3 model to formulate the stochastic SPP in a continuous state space with obstacles. A closed-form solution may not exist for this model; nevertheless, Bellman's principle is still applicable, resulting in the HJB equation. This equation is then solved using numerical approximation. A family of LCA is generalized to simplicial complexes, resulting is a Simplicial Label Correcting Algorithm (SLCA) for discretized equations. Our approximation method follows closely ideas arising in disjoint scientific areas: level-set methods [3], [15], [23] and the discretization of the continuous Markov Decision Process derived from the discrete Partially Observable Markov Decision Process [19], [27], [28]. Nevertheless, being seemingly unrelated, these methods are solving similar problems and are applicable for optimal robot navigation.

The rest of the paper is organized as follows. In Section II we formulate the stochastic SPP in the environment with obstacles and present the HJB equation associated with this problem. In Section III we derive a "nearby" deterministic control problem, which approximates the stochastic SPP. We also prove the bound on the difference between a solution to the stochastic problem and that of the nearby deterministic problem, which is found to be proportional to the time step of the arrival process. A piecewise linear numerical discretization for the deterministic problem is introduced in Section IV, along with proofs of the approximation error bounds. The Simplicial Label Correcting Algorithm (SLCA) is presented in Section V. In Section VI, we discuss experimental results obtained by Monte Carlo simulation. Finally, we conclude our findings and summarize future work in Section VII.

II. PROBLEM FORMULATION

A. Controlled Markov Chains

Consider $X \subset \mathbb{R}^d$ to be either a configuration space of a robot or a workspace for a point robot. Typically, if X is the workspace, then $d \leq 3$; however, higher values of d are expected if X is a configuration space. Let the *obstacle region*, $X_{\text{obs}} \subset X$, be an open set, such that for all $x \in X_{\text{obs}}$ the corresponding configuration is in collision with an obstacle. The *free space* consists of all collision free configurations and is denoted as $X_{\text{free}} = X \setminus X_{\text{obs}}$ and is compact. Assume robot's initial position, $x_{\text{init}} \in X_{\text{free}}$, is given, and let the *goal region*, $X_{\text{goal}} \subset X_{\text{free}}$, be a closed set with a nonempty interior.

To motivate a discrete time continuous space state transition model, we firstly consider an ordinary differential equation driven by a bounded, Lipschitz continuous, controllable vector field g with parametric noise:

$$\dot{x} = g(x, u, \omega), \text{ with } x(0) = x_{\text{init}}, \quad (3)$$

in which x is a state of the system in X, u is a control signal taken from a compact, convex, and symmetric around the origin input set $U \subset \mathbb{R}^d$, and ω is a noise parameter.

We assume a fixed time step arrival process as the model for ω. Let $\Delta t > 0$, $t_i = i\,\Delta t$, and ω_i be mutually independent identically distributed random samples from a compact probability space $(\Omega, \mathcal{F}, P)$. The disturbance parameter ω as a function of time is given as $\tilde{\omega}(t) = \omega_i$ for $t \in [t_i, t_{i+1})$ and $i = 0, \ldots, N$. Note that $\tilde{\omega}(t)$ is a random process, which is not Markov. Nevertheless, assuming that control sample rate is at most that of the arrival process, we recover Markov property by transition to discrete-time system.

Integrating (3) over time step Δt, we derive a stochastic discrete-time state transition model over X:

$$x_{i+1} = x_i + \Delta t\, f_{\Delta t}(x_i, u_i, \omega_i)\,, \ \text{with} \ x_0 = x_{\text{init}}\,. \quad (4)$$

In the above, $f_{\Delta t}(x_i, u_i, \omega_i) = \Delta t^{-1} \int_{t_i}^{t_{i+1}} g(x(t), u_i, \omega_i)\,\mathrm{d}t$, x_i is robot's state at time t_i, and u_i is a constant input signal during the time interval $[t_i, t_{i+1})$.

Using (4), we relate a given control history, $\tilde{u} = \{u_i\}_{i=0}^N$, a history of the disturbance process, $\tilde{\omega} = \{\omega_i\}_{i=0}^N$, and the initial condition, x_{init}, with a unique trajectory, $\tilde{x}(x_{\text{init}}, \tilde{u}, \tilde{\omega})$. Nodes of $\tilde{x}(x_{\text{init}}, \tilde{u}, \tilde{\omega})$ are at points $\{x_i\}_{i=0}^{N+1}$, which, in turn, satisfy (4) for all $i = 0, \ldots, N$. If $\tilde{\omega}$ is unknown, then we say that each control history, $\tilde{u}$, and initial condition, x_{init}, produce a random trajectory $\tilde{x}(x_{\text{init}}, \tilde{u})$ from the set of all trajectories, $\{\tilde{x}(x_{\text{init}}, \tilde{u}, \tilde{\omega}) \mid \tilde{\omega} \in \Omega^N\}$, with the statistics induced by P.

Considering control law given by a feedback function $\pi : X_{\text{free}} \to U$, that is, $u_i = \pi(x_i)$, the state transition equation becomes

$$x_{i+1} = x_i + \Delta t f_{\Delta t}(x_i, \pi(x_i), \omega_i)\,, \ \text{with} \ x_0 = x_{\text{init}}\,, \quad (5)$$

and the system is a Markov Chain (MC) over X. Similarly, we define a discrete-time trajectory, $\tilde{x}(x_{\text{init}}, \pi, \tilde{\omega})$, using nodes $\{x_i\}_{i=0}^{N+1}$, which satisfy (5) for all $i = 0, \ldots, N$. In the case of unknown disturbance, $\tilde{x}(x_{\text{init}}, \pi)$ is considered to be a random trajectory from $\{\tilde{x}(x_{\text{init}}, \pi, \tilde{\omega}) \mid \tilde{\omega} \in \Omega^N\}$. Since a feedback control is necessary to stabilize the system, we consider only feedback control laws for the rest of this paper.

B. Cost Functional

The performance of a control history is measured using a cost functional defined on the space of trajectories. Given a trajectory $\tilde{x}$, we introduce the total length cost functional

$$L(\tilde{x}) = \int_0^{t_{\text{f}}} \|\dot{\tilde{x}}(t)\|\,\mathrm{d}t\,, \quad (6)$$

if $t_{\text{f}} = \inf\{t > 0 \mid \tilde{x}(t) \in X_{\text{goal}}\} < \infty$ and the trajectory is in X_{free}, and $L(\tilde{x}) = \infty$, otherwise.

Since the value of L is a random variable if $\tilde{x}$ is a random trajectory, the shortest path problem is formulated with respect to average cost optimality criterion.

Problem 1 (The Shortest Path Problem). *Given $x_{\text{init}} \in X_{\text{free}}$, find optimal feedback control π^* such that for all π*

$$\mathrm{E}\left[L(\tilde{x}(x_{\text{init}}, \pi^*))\right] \leq \mathrm{E}\left[L(\tilde{x}(x_{\text{init}}, \pi))\right]\,, \quad (7)$$

in which $\mathrm{E}[\cdot]$ *is the expectation of a random variable.*

C. Optimality Principle and Dynamic Programming

The solution for the optimal feedback control is computed using Bellman's Dynamic Programming algorithm [4]. To this end, we introduce the optimal cost-to-go function for a system starting at x and executing optimal plan

$$V(x) = \min_\pi \ \mathrm{E}\left[L(\tilde{x}(x, \pi))\right]\,. \quad (8)$$

The function $V : X_{\text{free}} \to [0, \infty]$ denotes the minimal expected trajectory length to reach the goal from point x.

Using (6) and (8) and the expectation smoothing property, we derive the HJB equation for the cost-to-go function

$$V(x) = \min_{u \in U} \mathrm{E}\left[\int_0^{\Delta t} \|g(\tilde{x}(t), u, \omega)\|\,\mathrm{d}t + \right.$$
$$\left. V(x + \Delta t f_{\Delta t}(x, u, \omega))\right]. \quad (9)$$

Solving (9) with boundary conditions $V(x) = 0$ if $x \in X_{\text{goal}}$, and $V(x) = \infty$ if $x \in X_{\text{obs}}$, we find the optimal cost-to-go function for all $x \in X_{\text{free}}$. Once V is computed, the feedback control is given as the minimizing argument in (9):

$$\pi(x) = \arg\min_{u \in U} \mathop{\mathrm{E}}_{\omega_0}\left[\int_0^{\Delta t} \|g(\tilde{x}(t), u, \omega)\|\,\mathrm{d}t + \right.$$
$$\left. V(x + \Delta t f_{\Delta t}(x, u, \omega))\right]. \quad (10)$$

III. RELATION TO DETERMINISTIC CONTROL PROBLEM

It is common to approximate an optimal control problem with a discrete state CMC [29], for which value iteration and policy iteration algorithms [8] are known. On the other hand, our approach is reciprocal: we approximate the continuous state CMC with an optimal control problem, for which the fast marching method [15], ordered upwind method [26], or other numerical techniques [29], [32] can be applied.

Before we present any further derivations, we assume that ∇V is Lipschitz continuous with constant M on X_{free}. In general, the gradient of V may not be Lipschitz continuous in a small neighborhood of a critical point, in which the optimal trajectory changes its direction, for example, a corner of an obstacle. Nevertheless, the set of all critical points has measure zero.

Using the mean value theorem, we derive from (9)

$$\left|\min_{u \in U} \mathop{\mathrm{E}}_\omega \left[c_{\Delta t}(x, u, \omega) + \nabla V(x) \cdot f_{\Delta t}(x, u, \omega)\right]\right| \leq \epsilon, \quad (11)$$

in which $c_{\Delta t}(x, u, \omega) = \Delta t^{-1} \int_0^{\Delta t} \|g(\tilde{x}(t), u, \omega)\|\,\mathrm{d}t$ and $\epsilon = \Delta t M (\max_{x, u, \omega} \|f_{\Delta t}(x, u, \omega)\|)^2 < \infty$ (g is bounded).

We define expectations $c_{\Delta t}(x, u) \triangleq \mathrm{E}\left[c_{\Delta t}(x, u, \omega)\right]$ and $F_{\Delta t}(x, u) \triangleq \mathrm{E}\left[f_{\Delta t}(x, u, \omega)\right]$, and a deterministic system

$$\dot{x} = F_{\Delta t}(x, u)\,, \quad (12)$$

with cost functional

$$L'(\tilde{x}) = \int_0^{t_{\text{f}}} c_{\Delta t}(\tilde{x}(t), \tilde{u}(t))\,\mathrm{d}t\,, \quad (13)$$

in which $t_{\mathrm{f}} = \inf\{t > 0 \mid \tilde{x}(t) \in X_{\mathrm{goal}}\}$. The optimality equation for this deterministic system is

$$\min_{u \in U}\{c_{\Delta t}(x,u) + \nabla V(x) \cdot F_{\Delta t}(x,u)\} = 0, \quad (14)$$

which is an approximation of (11)

The derivations above show the relation between a continuous state CMC and a deterministic optimal control problem, which we call a *nearby* deterministic problem. In the next theorem, we prove the bound on the error between a solution of CMC and that of the nearby problem.

Theorem 2 (Modeling error bound). *Let V^* and V^{**} be solutions of (9) and (14) respectively. Also, let ∇V^* be Lipschitz with constant M on X_{free}, and g satisfy the conditions above. For sufficiently small Δt there exists C_1 independent of Δt, such that*

$$|V^*(x) - V^{**}(x)| \leq C_1 \Delta t \text{ for all } x \in X_{\mathrm{free}}. \quad (15)$$

Proof: Assume $\tilde{x}^* : [0, t_{\mathrm{f}}^*] \to X_{\mathrm{free}}$ and $\tilde{u}^* : [0, t_{\mathrm{f}}^*] \to U$ are optimal trajectory and control, respectively, of the stochastic system on the time interval $[0, t_{\mathrm{f}}^*]$, such that $\tilde{x}^*(0) = x$ for a given $x \in X_{\mathrm{free}}$. Consider $V^*(\tilde{x}^*)$ as a function of t. On one hand, we establish

$$-\int_0^{t_{\mathrm{f}}^*} \frac{\mathrm{d}}{\mathrm{d}t} V^*(\tilde{x}^*(t)) \, \mathrm{d}t = V^*(x). \quad (16)$$

On the other hand, from (11) follows

$$\int_0^{t_{\mathrm{f}}^*} \left(-\nabla V^*(\tilde{x}^*(t)) \cdot F_{\Delta t}(\tilde{x}^*(t), \tilde{u}^*(t), \omega)\right) \mathrm{d}t \geq$$
$$\int_0^{t_{\mathrm{f}}^*} c_{\Delta t}(\tilde{x}^*(t), \tilde{u}^*(t)) \, \mathrm{d}t - t_{\mathrm{f}}^* \epsilon. \quad (17)$$

Thus,

$$V^*(x) \geq \int_0^{t_{\mathrm{f}}^*} c_{\Delta t}(\tilde{x}^*(t), \tilde{u}^*(t)) \, \mathrm{d}t - t_{\mathrm{f}}^* \epsilon. \quad (18)$$

Similarly, considering $V^{**}(\tilde{x}^*)$ as a function of t, we conclude from (14)

$$V^{**}(x) \leq \int_0^{t_{\mathrm{f}}^*} c_{\Delta t}(\tilde{x}^*(t), \tilde{u}^*(t)) \, \mathrm{d}t. \quad (19)$$

Hence, from (18) and (19), it follows

$$V^{**}(x) - V^*(x) \leq t_{\mathrm{f}}^* \epsilon. \quad (20)$$

Further, assume $\tilde{x}^{**} : [0, t_{\mathrm{f}}^{**}] \to X_{\mathrm{free}}$ and $\tilde{u}^{**} : [0, t_{\mathrm{f}}^{**}] \to U$ are optimal trajectory and control, respectively, for the deterministic system on the time interval $[0, t_{\mathrm{f}}^{**}]$, such that $\tilde{x}^{**}(0) = x$ for the same $x \in X_{\mathrm{free}}$. Considering $V^*(\tilde{x}^{**})$ and $V^{**}(\tilde{x}^{**})$ as functions of t, we derive from (11) and (14)

$$V^*(x) - V^{**}(x) \leq t_{\mathrm{f}}^{**} \epsilon. \quad (21)$$

Thus, it follows from inequalities (20) and (21) that

$$|V^*(x) - V^{**}(x)| \leq \max\{t_{\mathrm{f}}^*, t_{\mathrm{f}}^{**}\} \epsilon. \quad (22)$$

The upper bound on t_{f}^* and t_{f}^{**} is given as follows: $\max\{t_{\mathrm{f}}^*, t_{\mathrm{f}}^{**}\} \leq \max\{V^*(x), V^{**}(x)\}/c_{\mathrm{min}} <$

∞, in which $c_{\mathrm{min}} = \min_{x,u} c_{\Delta t}(x,u) > 0$. Hence, we conclude the result of the theorem for $C_1 = \max\{V^*(x), V^{**}(x)\} 2 M f_{\mathrm{max}}^2 / c_{\mathrm{min}}$. ∎

IV. NUMERICAL DISCRETIZATION

A closed-form solution is rarely available for (14); thus, we must resort to numerical methods to find an approximate solution. To build a numerical discretization of (14), we follow closely the level-set method technique [15] and other related numerical schemes [23], [29], [32].

A. Piecewise Linear Approximation on Simplicial Complexes

First, define a simplicial complex $(X_{\mathrm{d}}, \mathcal{T})$, in which $X_{\mathrm{d}} = \{x_i \in X_{\mathrm{free}} \mid 1 \leq i \leq N\}$ is a set of vertices, and $\mathcal{T}$ is an abstract simplicial complex over a set $\{1, \ldots, N\}$. For a simplex $T \in \mathcal{T}$ the *geometrical representation* of T is set $X_T = \{\sum_{i \in T} \alpha_i x_i \mid \alpha_i \geq 0, \sum_{i \in T} \alpha_i = 1\}$. We call $(X_{\mathrm{d}}, \mathcal{T})$ a *simplicial discretization* of X_{free}, or a *mesh* on X_{free}, if $\bigcup_{T \in \mathcal{T}} X_T = X_{\mathrm{free}}$ and for all $T, T' \in \mathcal{T}$ their geometrical representations satisfy $X_T \cap X_{T'} = X_{T \cap T'}$.

Second, define *barycentric coordinates* $\{\alpha_i\}_{i \in T}$ of point $x \in X_T$, such that $x = \sum_{i \in T} \alpha_i x_i$, $\sum_{i \in T} \alpha_i = 1$, and $\alpha_i \geq 0$ for all $i \in T$. Use a mesh on X_{free} to approximate V with a piecewise linear function $\hat{V}$ defined as V_i at each node $x_i \in X_{\mathrm{d}}$ as follows: for some $T \in \mathcal{T}$ and all $x = \sum_{i \in T} \alpha_i x_i$

$$\hat{V}(x) \triangleq \sum_{i \in T} \alpha_i V_i. \quad (23)$$

Note that, since $\hat{V}(x)$ is a linear function on X_T, the gradient of $\hat{V}$, denoted as $\nabla_T \hat{V}$, is a constant vector on X_T.

Third, we approximate $c_{\Delta t}$ and $F_{\Delta t}$ as functions of x with constant functions on each X_T, denoted $\hat{c}_T$ and $\hat{F}_T$, respectively.

Finally, for all $T \in \mathcal{T}$ we would like $\nabla_T \hat{V}$ to satisfy (14):

$$0 = \min_{u \in U}\left\{\hat{c}_T(u) + \nabla_T \hat{V} \cdot \hat{F}_T(u)\right\}. \quad (24)$$

In addition to (24), we impose boundary conditions

$$V_i = 0, \text{ for all } i \text{ such that } x_i \in X_{\mathrm{goal}}. \quad (25)$$

Thus, equations (24) and (25) define a system of algebraic equations with respect to unknowns $\{V_i\}_{i=1}^N$. It follows from (24), the approximate feedback control $\hat{\pi}$ is a constant function on X_T, which we denote as

$$\hat{\pi}_T(x) = \arg\min_{u \in U}\left\{\hat{c}_T(u) + \nabla_T \hat{V} \cdot \hat{F}_T(u)\right\}. \quad (26)$$

B. Error Bounds

The interpolation error is defined as the difference between V, a solution of (14), and $\hat{V}$, a solution of (24). Before we prove the bound on the interpolation error, we need two supplemental lemmas[1].

---

[1] The proof of these lemmas is technical and tedious. Moreover, it does not provide any additional insight on the problem; thus, we avoid it.

Lemma 3 (Linear Interpolation on d-dimensional Simplex). *Let X_T be a d-dimensional simplex with vertices $\{x_i\}_{i=0}^d$. Assume V is continuously differentiable on X_T, and ∇V is Lipschitz on X_T with constant M. Consider linear function $\hat{V}$ such that $\hat{V}(x_i) = V(x_i)$ for all $i = 1, \ldots, d$.*

Under this assumptions

$$|V(x) - \hat{V}(x)| \leq M d h_T^2/4 , \text{ for all } x \in X_T , \quad (27)$$

in which $h_T = \sup_{x,x' \in X_T} \|x - x'\|$.

Lemma 4 (Local Interpolation Error). *Let the optimal trajectories of (14) and (24) pass through x_i and follow through the interior of X_T ($i \in T$)[2]. Let $T_i = T \setminus \{i\}$. These assumptions imply*

$$|V(x) - \hat{V}(x)| \leq M d h_T^2/4 + \max_{j \in T_i} |V(x_j) - \hat{V}(x_j)| . \quad (28)$$

Using the result of Lemma 4, we prove the next theorem that determines the bound on the approximate solution error for a piecewise linear discretization.

Theorem 5 (Global Interpolation Error). *Assume the conditions of Lemma 4, then the global error bound is*

$$|V(x) - \hat{V}(x)| \leq C_2 \overline{h} , \quad (29)$$

in which $\overline{h} = \max_{T \in \mathcal{T}} h_T$ and C_2 is constant independent of the mesh.

Proof: From Lemma 4 it follows that the local error is bounded by $M(d-1)h_T^2/4$. Assuming that error accumulates at each iteration of the algorithm, conclude that global error is bounded by $IM(d-1)h_T^2/4$ (here I is the maximal iteration number). Finally, the iteration number is proportional to the diameter of X_{free}, with respect to geodesic distance, divided by $\underline{h} = \min_{T \in \mathcal{T}} h_T$. Thus,

$$|V(x) - \hat{V}(x)| \leq \frac{\text{diam}(X_{\text{free}})M(d-1)}{4} \frac{\overline{h}}{\underline{h}} \overline{h} . \quad (30)$$

By letting $C_2 = \text{diam}(X_{\text{free}})M(d-1)\overline{h}/4\underline{h}$ and assuming that smallest to largest cell size ratio is kept constant, we prove the theorem. ∎

Finally, we formulate a corollary from Theorems 2 and 5. The next theorem proves the total error bound using both: the approximation of CMC with the deterministic control problem, and using a piecewise linear interpolation to solve the deterministic control problem.

Theorem 6 (Total Error). *Assume system (4) is given along with a cost functional (6). Let V be the cost-to-go function, defined in (8). Also, let the discretization procedure, outlined in Section IV-A, be used to find the approximate cost-to-go function $\hat{V}$. Under these conditions, the total error is bounded by*

$$|V(x) - \hat{V}(x)| \leq C_1 \Delta t + C_2 \overline{h} , \quad (31)$$

---

[2]In the case the optimal trajectories do not follow in the same simplex, a more general result can be established by considering a union of simplices.

in which C_1 and C_2 are given in Theorems 2 and 5, respectively, and independent of Δt and $\overline{h}$.

V. SIMPLICIAL LABEL CORRECTING ALGORITHM

If considered for all $T \in \mathcal{T}$, (24) defines a system of algebraic equations. Generic numerical algorithms exist to solve this type of systems, for example, fixed-point iteration, Newton's method, and the secant updating method. In fact, two of these methods are successfully applied in the case of finite state CMCs: The value iteration algorithm is an instance of the fixed-point iteration and the policy iteration is an implementation of Newton's method.

On the other hand, generic methods usually do not efficiently exploit the structure of the problem. Particularly, for finite state CMCs, if the policy is *consistently improving* [24], [29], then the nodes can be computed in increasing order of cost-to-go function values. Dijkstra's algorithm is an efficient implementation of this idea, which computes the cost-to-go function in $O(n \log(n))$ time, where n is the vertex number.

Moreover, a generalization of Dijkstra's algorithm, the family of Label Correcting Algorithms, was first introduced in [12] for the SPP on graphs. In the generic LCA, the order in which nodes are computed is not specified, and various heuristics can be used to further reduce the total running time [6]. Moreover, the family of LCA generalizes many other algorithms, such as the A* algorithm, Bellman-Ford algorithm, and so on.

Another generalization of Dijkstra's algorithm was proposed in [29] to find the continuous shortest path among obstacles and in [15] to find geodesics on triangulated surfaces. In [32] the authors formulated the Simplicial Dijkstra's and A* algorithms, which compute the shortest path on simplicial complexes of arbitrary dimension. Thus, by discretizing X_{free} with a simplicial complex, an approximation of the shortest path in X_{free} (of any dimension) can be found.

In this paper, we propose a generic Simplicial Label Correcting Algorithm (SLCA) as a generalization of the previously introduced the Simplicial Dijkstra and the Simplicial A* algorithms, as well as LCA on graphs:

Input: Simplicial complex $(X_d, \mathcal{T})$, goal set X_{goal}
Output: Approximations $\hat{V}_i$ of $V(x_i)$ and $\hat{\pi}_T$ of $\pi(x)$
1: Initialize set Q of "open" nodes i for which $x_i \in X_{\text{goal}}$
2: Initialize set of labels $\hat{V}$ for all vertices, such that $\hat{V}_i \leftarrow 0$ for $i \in Q$, and $\hat{V}_i \leftarrow \infty$ otherwise
3: **while** Q is not empty **do**
4:     Pop j from Q
5:     **for all** T such that $j \in T$ **do**
6:         $i \leftarrow \arg\max_{k \in T}\{V_k\}$
7:         **if** $i \neq j$ **then**
8:             $(V^*, \pi^*) \leftarrow$ **update**(i, T)
9:             **if** $\hat{V}^* < \hat{V}_i$ **then**
10:                 $\hat{V}_i \leftarrow V^*$; $\pi_T \leftarrow \pi^*$
11:                 Push i into Q if $i \notin Q$

We apply this algorithm to solve the approximate stochastic shortest path problem among obstacles using simplicial decomposition of X_{free}.

Note that the vertex selection strategy (see line 4 above) is not specified in the generic SLCA, and it is left up to the specific implementation. For example, the Simplicial Bellman-Ford Algorithm uses a FIFO queue; the Simplicial Dijkstra algorithm is implemented using a priority queue over the values of the cost-to-go function; adding the heuristic of the cost-to-come function generates the Simplicial A* algorithm. More advanced vertex selection strategies, which significantly reduce the running time for LCA, are analyzed in [6]. Incorporation of these strategies into SLCA is the topic of the ongoing investigation.

Function **update** in line 8 is defined to solve (24) locally. It takes simplex T and vertex $i \in T$ as arguments, and returns V_i such that (24) holds. Note that an alternative numerical discretization requires a different implementation of **update**. However, the SLCA framework remains unchanged for various numerical procedures.

VI. Results and Discussion

Two stochastic systems are considered in the numerical experiments. In both cases $d = 2$, $\Omega = [-\alpha, \alpha]$ is equipped with a uniform probability measure, and $U = \{u \in \mathbb{R}^2 \mid \|u\| = 1\}$. System 1 and System 2 are defined using (4) and the corresponding functions $g_1(x, u, \omega) = u + \omega R(\pi/2)u$, and $g_2(x, u, \omega) = R(\omega)u$, in which R is a rotation matrix

$$R(\phi) = \begin{bmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{bmatrix}. \tag{32}$$

First, we investigate the convergence of the proposed numerical method in a simple domain, for which the optimal feedback control law is known. In this setting, we approximate the true average cost with $V_{\mathrm{MC}}(x)$, derived from a Monte Carlo simulation of a hundred sampled trajectories. We compare computed cost-to-go function, $\hat{V}$, with V_{MC} using L_∞ and L_2 error estimates, given as $\mathcal{E}_{L_\infty}(\hat{V}, V_{\mathrm{MC}}) = \sup_{x \in X_{\mathrm{free}}} \left| \hat{V}(x) - V_{\mathrm{MC}}(x) \right|$ and $\mathcal{E}_{L_2}(\hat{V}, V_{\mathrm{MC}}) = \left(\int_{X_{\mathrm{free}}} \left| \hat{V}(x) - V_{\mathrm{MC}}(x) \right|^2 \mathrm{d}x \right)^{1/2}$, respectively. Figure 1 depicts the log-log plot of error with respect to $\overline{h}$ for the considered systems. Lines with slope 1 are drawn to facilitate linear relation between the error and the mesh size and to illustrate the agreement with Theorem 6.

Second, the application of SLCA to a real world problem is demonstrated on a general domain with obstacles. The heat map of the cost-to-go function and a set of trajectories computed using Monte Carlo simulations are illustrated in Figure 2. From the simulation, it is evident that open-loop controls are incapable of safely steering the system towards the goal. The optimal feedback control law, however, succeeds in this task and guarantees the average path length optimality.

VII. Conclusions

In conclusion, we considered a discrete-time stochastic system with parametric disturbances over a continuous state space with obstacles. The average shortest path problem is
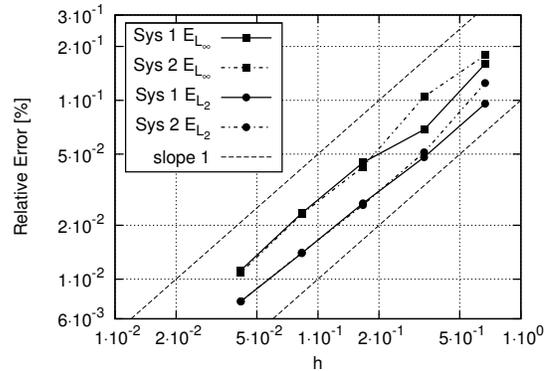
Fig. 1. The error vs. mesh quality parameter $\overline{h}$.
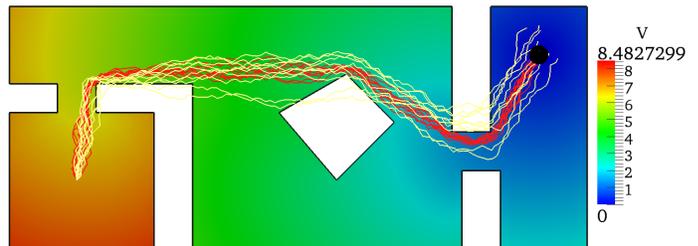
Fig. 2. Experimental domain with X_{goal} represented by the black circle in the upper-right corner, and a Monte Carlo simulation of the optimal open-loop control (yellow) and the optimal feedback control (red).

formulated for this system. The finite difference HJB equation was derived for this stochastic SPP. The closed form solution is not available for this type of equations, in general. Thus, we must resort to numerical computations.

We demonstrated the relation between the HJB equation for the system with disturbances and the HJB equation for the nearby deterministic system. In Theorem 2, we proved the bound on the error between cost-to-go functions for the stochastic and deterministic problems. This result allowed us to compute the optimal cost-to-go function of the stochastic problem using numerical methods originally developed for deterministic control problems.

Additionally, we proposed the generalization of the family of LCA over simplicial complexes. The family of SLCA is capable of solving continuous optimal control problems directly, while avoiding artificial discretizations by reachability graphs or trees. The result of the proposed algorithm is the optimal cost-to-go function and the feedback control defined on the simplicial complex. Also, SLCA includes the Simplicial Dijkstra and the Simplicial A* Algorithms, previously introduced by the authors. Moreover, various node selection strategies can be implemented to improve the running time.

Finally, we analyzed the error between the numerical solution, obtained by the SLCA, and that computed using a Monte Carlo simulation. We demonstrated that the error decreases linearly with respect to the mesh size, h, as the computational mesh is refined, which confirms the prediction of Theorem 6.

REFERENCES

[1] Ron Alterovitz, Thierry Simeon, and Ken Goldberg. The Stochastic Motion Roadmap: A Sampling Framework for Planning with Markov Motion Uncertainty. In *Robotics Science and Systems*, June 2007.

[2] M. Athans. The role and use of the stochastic linear-quadratic-Gaussian problem in control system design. *Automatic Control, IEEE Transactions on*, 16(6):529–552, December 1971.

[3] T. J. Barth and J. A. Sethian. Numerical Schemes for the Hamilton-Jacobi and Level Set Equations on Triangulated Domains. *Journal of Computational Physics*, 145(1):1–40, September 1998.

[4] Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, March 1957.

[5] Dimitri P. Bertsekas. Convergence of discretization procedures in dynamic programming. *Automatic Control, IEEE Transactions on*, 20(3):415–419, June 1975.

[6] Dimitri P. Bertsekas. A simple and fast label correcting algorithm for shortest paths. *Networks*, 23(8):703–709, 1993.

[7] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control, Vol. II*. Athena Scientific, 3rd edition, January 2007.

[8] Dimitri P. Bertsekas and John N. Tsitsiklis. An Analysis of Stochastic Shortest Path Problems. *Mathematics of Operations Research*, 16(3), 1991.

[9] Rene Boel and Pravin Varaiya. Optimal control of Markovian jump processes. In *Decision and Control including the 14th Symposium on Adaptive Processes, 1975 IEEE Conference on*, volume 14, pages 153–156. IEEE, December 1975.

[10] A. Bry and N. Roy. Rapidly-exploring Random Belief Trees for motion planning under uncertainty. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 723–730. IEEE, May 2011.

[11] B. Fox. Finite-state approximations to denumerable-state dynamic programs. *Journal of Mathematical Analysis and Applications*, 34(3):665–670, June 1971.

[12] Giorgio Gallo and Stefano Pallottino. Shortest path methods: A unifying approach. In Giorgio Gallo and Claudio Sandi, editors, *Mathematical Programming Studies*, volume 26 of *Mathematical Programming Studies*, chapter 3, pages 38–64. Springer Berlin Heidelberg, Berlin, Heidelberg, 1986.

[13] Giorgio Gallo and Stefano Pallottino. Shortest path algorithms. *Annals of Operations Research*, 13(1):1–79, December 1988.

[14] Geoffrey J. Gordon. Stable Function Approximation in Dynamic Programming. Technical report, Carnegie-Mellon Univ., School of Computer Science, Pittsburgh, PA, January 1995.

[15] R. Kimmel and J. A. Sethian. Computing geodesic paths on manifolds. *Proceedings of the National Academy of Sciences of the United States of America*, 95(15):8431–8435, July 1998.

[16] Harold J. Kushner. Optimality Conditions for the Average Cost per Unit Time Problem with a Diffusion Model. *SIAM Journal on Control and Optimization*, 16(2):330–346, 1978.

[17] Harold J. Kushner. Numerical Methods for Stochastic Control Problems in Continuous Time. *SIAM Journal on Control and Optimization*, 28(5):999–1048, 1990.

[18] Steven M. Lavalle and Prashanth Konkimalla. Algorithms for Computing Numerical Optimal Feedback Motion Strategies. *The International Journal of Robotics Research*, 20(9):729–752, September 2001.

[19] William S. Lovejoy. Computationally Feasible Bounds for Partially Observed Markov Decision Processes. *Operations Research*, 39(1), 1991.

[20] Sean P. Meyn. *Control Techniques for Complex Networks*. Cambridge University Press, 1 edition, December 2007.

[21] Sean P. Meyn and R. L. Tweedie. Stability of Markovian Processes III: Foster-Lyapunov Criteria for Continuous-Time Processes. *Advances in Applied Probability*, 25(3), 1993.

[22] Thomas E. Morton. On the Asymptotic Convergence Rate of Cost Differences for Markovian Decision Processes. *Operations Research*, 19(1), 1971.

[23] Rémi Munos and Andrew Moore. Variable Resolution Discretization in Optimal Control. *Machine Learning*, 49(2):291–323, November 2002.

[24] Sang Nguyen and Stefano Pallottino. Hyperpaths and shortest hyperpaths Combinatorial Optimization. In Bruno Simeone, editor, *Combinatorial Optimization*, volume 1403 of *Lecture Notes in Mathematics*, chapter 10, pages 258–271. Springer Berlin / Heidelberg, 1989.

[25] J. W. Roberts, I. R. Manchester, and R. Tedrake. Feedback controller parameterizations for Reinforcement Learning. In *Adaptive Dynamic Programming And Reinforcement Learning (ADPRL), 2011 IEEE Symposium on*, pages 310–317. IEEE, April 2011.

[26] J. A. Sethian and A. Vladimirsky. Ordered Upwind Methods for Static Hamilton–Jacobi Equations: Theory and Algorithms. *SIAM Journal on Numerical Analysis*, 41(1):325–363, 2003.

[27] Richard D. Smallwood and Edward J. Sondik. The Optimal Control of Partially Observable Markov Processes Over a Finite Horizon. *Operations Research*, 21(5):1071–1088, 1973.

[28] Edward J. Sondik. The Optimal Control of Partially Observable Markov Processes Over the Infinite Horizon: Discounted Costs. *Operations Research*, 26(2):282–304, 1978.

[29] John N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *Automatic Control, IEEE Transactions on*, 40(9):1528–1538, August 1995.

[30] Jur van den Berg, Pieter Abbeel, and Ken Goldberg. LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information. *The International Journal of Robotics Research*, 30(7):895–913, June 2011.

[31] D. J. White. Dynamic programming, Markov chains, and the method of successive approximations. *Journal of Mathematical Analysis and Applications*, 6:373–376, 1963.

[32] Dmitry S. Yershov and Steven M. LaValle. Simplicial Dijkstra and A* algorithms for optimal feedback planning. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3862–3867. IEEE, 2011.

[33] Yu Zhou and G. S. Chirikjian. Probabilistic models of dead-reckoning error in nonholonomic mobile robots. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 2, pages 1594–1599 vol.2. IEEE, 2003.