A Continuous-State Version of Discrete Randomized Shortest-Paths, with Application to Path Planning

Silvia García-Díez, Eric Vandenbussche and Marco Saerens

***Abstract*— This work investigates the continuous-state counterpart of the discrete randomized shortest-path framework (RSP, [23]) on a graph. Given a weighted directed graph G, the RSP considers the policy that minimizes the expected cost (exploitation) to reach a destination node from a source node, while maintaining a constant relative entropy spread in the graph (exploration). This results in a Boltzmann probability distribution on the (usually infinite) set of paths connecting the source node and the destination node, depending on an inverse temperature parameter θ. This framework defines a biased random walk on the graph that gradually favors low-cost paths as θ increases. It is shown that the continuous-state counterpart requires the solution of two partial differential equations – providing forward and backward variables – from which all the quantities of interest can be computed. For instance, the best local move is obtained by taking the gradient of the logarithm of one of these solutions, namely the backward variable. These partial differential equations are the so-called steady-state Bloch equations to which the Feynman-Kac formula provides a path integral solution. The RSP framework is therefore a discrete-state equivalent of the continuous Feynman-Kac diffusion process involving the Wiener measure. Finally, it is shown that the continuous-time continuous-state optimal randomized policy is obtained by solving a diffusion equation with an external drift provided by the gradient of the logarithm of the backward variable, playing the role of a potential.**

I. Introduction

Minimum-cost problems on a graph are of capital importance in a variety of problems, from robot path planning, to maze solving. Path planning [16] is a well-known problem in the robotics community, described by [26] as "checking the consequences of an action in an internal model before performing such actions". Originally, single-source single-destination problems were tackled, but when encountering multiple sources or destinations, more sophisticated approaches are needed (see, e.g., [18]).

The **Randomized Shortest Paths** (RSP) approach is a discrete method that tackles the problem of finding the minimum-cost path on a graph while keeping a constant level of spread entropy [23]. The introduced path randomization allows balancing the load (number of packages) per path in the case of multiple goals, while exploiting those goals in parallel. The RSP framework was inspired by the work of Akamatsu [1], who proposed a randomized policy for routing traffic in transportation networks, penalizing long paths. Similarly, the RSP assigns a probability distribution to the set of admissible paths. This distribution, though peaked around optimal paths, let the random walker take a random transition according to the (Shannon) entropy spread in the graph. In this way, we consider the entropy as a parameter controlling the trade-off between the exploration and the exploitation of the graph (when the entropy is zero, there is no uncertainty, and therefore no exploration).

In this paper, the continuous-state counterpart of the discrete RSP is investigated and applied to path planning. By defining a grid where each node has four neighbors (north, south, east, west) situated at a distance ϵ, and taking the limit $\epsilon \to 0$, a system of two independent partial differential equations computing forward and backward variables is obtained (Laplacian-based diffusion equations where the initial nodes are considered as sources and the destination nodes as sinks, and vice-versa). Once these variables are known, all the quantities of interest – such as the optimal randomized policy – can be easily computed.

The use of physical analogies in path planning methods is not new, though. Wave propagation methods [6], [21] represent the first of the three main kinds of physical analogy in optimal path planning. In [21], the authors introduce an analogue method for labyrinth solving based on parallel wave propagation through all possible paths in a reaction-diffusion media. Potential field methods [5], [11], [15] are the second popular technique borrowed from physics. The method in [15] proposes a real-time path planner based on an artificial potential field where the goal is represented as an attractive pole and the obstacles as repulsive faces. Similarly, the work from [5] proposes a smoothed version with two major advantages, i.e., it is based on a Laplace equation (which avoids local minima), and it benefits from the use of massively parallel architectures to solve this equation (efficient computation).

Eventually, diffusion strategies [6], [17], [18], [25], [24], [26], [27] appeared as the third type of widely studied physical analogy in many path planning algorithms. A reaction-diffusion mechanism is presented in [26] in order to complete the behavior of a multi-agent model based on analogical representations. The propagation of an agent's information through his network of neighbors leads to the computation of a gradient field that will guide a robot on an obstacle grid. The DIP (Diffusion in Potential Fields) method [6] computes a gradient field on a grid where each cell has an activation function which is computed by diffusion in a similar fashion to an automata's activation function. Laplace's equation is also used in diffusion strategies, e.g., the work in [25], [24] introduces the theoretical basis for a dynamic path planning approach using an unsteady diffusion equation with Dirichlet boundary conditions. This method enjoys the nice properties of Laplacian methods (high-speed, high efficiency), but also adapts to changing environments. A similar approach based on a fluid model represented by a Poisson equation with Newmann boundary conditions is presented by [17]. Analogue systems have also adopted this strategy, as for instance the method from [27], which represents obstacles as non-conducting solids in a conducting medium. More sophisticated methods, such as [18], cope with uneven natural terrain path planning. In this case, a viscous fluid formalism where external forces and friction are taken into account is used for multiple-source multiple-destination problems.

The continuous-state RSP version presented here belongs to the diffusion methods involving Laplacian differential operators with Dirichlet boundary conditions for multiple-source multiple-destination problems. Its most interesting properties are the fact that (1) it depends on a diffusion parameter controlling the trade-off between exploration and exploitation, (2) the resulting policy is optimal since it ensures minimal expected cost for a constant

S. García-Díez and M. Saerens are with ICTEAM and LSM, Université catholique de Louvain, 1348 Louvain-la-Neuve, Belgium {silvia.garciadiez,marco.saerens}@ucLouvain.be.

E. Vandenbussche is with IRIDIA, Université libre de Bruxelles, 1050 Bruxelles, Belgium evdbussc@gmail.com.

exploration, (3) it provides the minimum-cost policy when the diffusion parameter is low, and (4) it shows some interesting links between biased random walks on a graph (discrete RSP) and continuous-state Feynman-Kac diffusion processes. Its main drawback is that paths are considered as independent.

Section II introduces the RSP framework and the computation of its main quantities of interest. The continuous-state RSP extension is developed in Section III. Further details as well as as a physical interpretation and boundary conditions are here specified. The dynamic continuous-time continuous-state optimal policy is developed in Section IV. Two practical, simulation example, cases involving path planning are presented in Section V. Section VI discusses the obtained conclusions and possible extensions.

II. The discrete RSP framework

This section provides a short account of the discrete randomized shortest path (RSP) framework [23], [29], in the context of a single-source single-destination problem, which was initially inspired by a stochastic traffic assignment model [1]. It will be shown that this RSP framework allows solving minimal-cost problems in a graph by means of simple linear algebra operations.

A. RSP: background and notations

Let G be a weighted directed graph containing n nodes, together with a source, or initial, node (node 1) and a destination, or goal, node (node n), with $n \neq 1$ (we assume that source and destination nodes are different). Multiple-source (-destination) problems are tackled by adding a new, dummy, source (destination) node connected through a directed link to all the original source (destination) nodes with a zero associated cost. Let us also assume that node n can be reached from the source node, and a positive cost $c_{kk'}$ is associated to each arc $k \to k'$. We consider graphs with no self-loops, which implies that the cost of remaining in the same state is infinite, $c_{kk} = \infty$ for all k. Similarly, an infinite cost is assumed when there is no arc between node k and k'. The cost matrix containing $c_{kk'}$ will be referred to as $\mathbf{C}$. We now adopt a **sum-over-paths formalism** [19]: let us further denote by $\mathcal{P}$ the set of all paths $\{\wp\}$ (including cycles) joining the source node 1 and the destination node n. Let the initial, reference, policy be uniform (pure exploration – the costs are not taken into account), $p^{(0)}_{kk'} = 1/n_k$, where n_k is the outdegree of node k. Let us now seek the optimal probability distribution $\mathrm{P}^*(\wp)$ on the set of paths $\mathcal{P}$ – assumed to be independent – minimizing the expected cost for reaching the destination node from the source node (exploitation) while maintaining a constant relative entropy J_0 with respect to this reference policy (exploration). The problem to solve is

$$\left| \begin{array}{ll} \underset{\mathrm{P}(\wp)}{\text{minimize}} & \sum_{\wp\in\mathcal{P}} \mathrm{P}(\wp) C(\wp) \\ \text{subject to} & \sum_{\wp\in\mathcal{P}} \mathrm{P}(\wp) \ln(\mathrm{P}(\wp)/\mathrm{P}^{(0)}(\wp)) = J_0 \end{array} \right. \quad (1)$$

where $C(\wp)$ is the sum of the local costs $c_{kk'}$ along path $\wp$ and $\mathrm{P}(\wp)$ is the probability of following $\wp$. Moreover, $\mathrm{P}^{(0)}(\wp)$ is the probability of following path $\wp$ when using the reference policy, i.e., transition probabilities $p^{(0)}_{kk'}$. The sum in (1) is defined on all paths $\wp \in \mathcal{P}$. It can be easily shown [19] that the optimal path probability distribution is a **Boltzmann distribution** on the set of paths $\mathcal{P}$,

$$\boxed{\mathrm{P}^*(\wp) = \frac{\mathrm{P}^{(0)}(\wp)\exp\left[-\theta C(\wp)\right]}{\sum_{\wp'\in\mathcal{P}} \mathrm{P}^{(0)}(\wp')\exp\left[-\theta C(\wp')\right]}} \quad (2)$$

where $\theta > 0$ is the inverse temperature and is directly related to the relative entropy J_0. This equation provides the **optimal randomized policy** expressed in terms of path probabilities. As expected, shorter paths $\wp$, having small $C(\wp)$, will be favored. Indeed, when θ is large, the probability distribution defined by Equation (2) is biased towards short (low cost) paths, i.e., when $\theta \to \infty$, the probability distribution peaks around the shortest paths. Let us further define the **partition function** as

$$\mathcal{Z} = \sum_{\wp\in\mathcal{P}} \mathrm{P}^{(0)}(\wp)\exp\left[-\theta C(\wp)\right] \quad (3)$$

Clearly, the **expected number of passages** $\overline{\eta}(k,k')$ through arc $k \to k'$, and the expected number of passages per node $\overline{n}(k')$ are given by ([13], but see [8] for an alternative derivation in the sum-over-paths context)

$$\overline{\eta}(k,k') = -\frac{1}{\theta}\frac{\partial(\ln\mathcal{Z})}{\partial c_{kk'}} \quad (4) \qquad \overline{n}(k') = \sum_{k\in P(k')} \overline{\eta}(k,k') \quad (5)$$

where $P(k')$ is the set of predecessors of node k' and $k' \neq 1$.

For the single-source single-destination problem with independent paths, the optimal policy expressed in terms of path probabilities in Equation (2), providing the minimum-cost policy for a constant J_0, can be re-expressed in terms of **local**, **optimal**, **transition probabilities**, i.e., a local first-order Markov policy (see [23])

$$p^*_{kk'} = \frac{\overline{\eta}(k,k')}{\sum_{l\in S(k)} \overline{\eta}(k,l)}, \text{ for } k \neq n \quad (6)$$

which provides the transition probabilities from every node of interest of the graph. Here, $S(k)$ is the set of successors of node k. Other useful measures, such as the expected cost, can also be calculated through the partition function [13]

$$\overline{C} = -\frac{1}{\theta}\frac{\partial(\ln\mathcal{Z})}{\partial\theta} \quad (7)$$

Let us now show how this partition function, $\mathcal{Z}$, can be computed from the cost matrix and the reference transition probabilities.

B. Computation of the partition function

It is shown in [23], [19] that the partition function $\mathcal{Z}$ can be computed from the immediate cost matrix, $\mathbf{C}$, and the reference transition matrix, $\mathbf{P}^{(0)}$, containing the $p^{(0)}_{kk'}$. We first define $\mathbf{W}$ as

$$\mathbf{W} = \mathbf{P}^{(0)} \circ \exp[-\theta\mathbf{C}] = \exp[-\theta\mathbf{C} + \ln\mathbf{P}^{(0)}] \quad (8)$$

where the logarithm/exponential functions are taken element-wise ($\circ$ is the Hadamard matrix product). Further developments [23] show that the partition function (Equation (3)) can be computed as

$$\mathcal{Z} = [(\mathbf{I}-\mathbf{W})^{-1} - \mathbf{I}]_{1n} = [\mathbf{Z}-\mathbf{I}]_{1n} = [\mathbf{Z}]_{1n} = z_{1n} \quad (9)$$

and by analogy with Markov chains, $\mathbf{Z} = (\mathbf{I}-\mathbf{W})^{-1} = \mathbf{I} + \mathbf{W} + \mathbf{W}^2 + \ldots$, will be called the **fundamental matrix**. Its elements (k,l) are denoted as z_{kl}.

C. Computation of the main quantities

Our next step aims at computing the expected number of passages through arc $k \to k'$. The partial derivative from Equation (4) can be readily computed [23], [19] as

$$\boxed{\overline{\eta}(k,k') = \frac{z_{1k} w_{kk'} z_{k'n}}{z_{1n}}} \quad (10)$$

where $w_{kk'}$ is element (k,k') of matrix $\mathbf{W}$. From Equation (5), the expected number of visits is given by

$$\boxed{\overline{n}(k') = \frac{z_{1k'} z_{k'n}}{z_{1n}}, \text{ for } k' \neq 1} \tag{11}$$

where we used the identity $z_{1k'} = \delta_{1k'} + \sum_{k \in P(k')} z_{1k} w_{kk'}$ that can easily be deduced from $\mathbf{Z} = \mathbf{I} + \mathbf{WZ}$. Moreover, from Equations (6) and (10), the **optimal transition probabilities** are

$$\boxed{p^*_{kk'} = \frac{w_{kk'} z_{k'n}}{\sum_{l \in S(k)} w_{kl} z_{ln}} = \frac{w_{kk'} z_{k'n}}{z_{kn}}, \text{ for } k \neq n} \tag{12}$$

where we used $z_{kn} = \delta_{kn} + \sum_{l \in S(k)} w_{kl} z_{ln}$. When $\theta \to \infty$, the $p^*_{kk'}$ encode the minimum-cost policy, while for intermediate values of θ, following Equation (1), they define a Markov chain minimizing the expected cost to the destination for a given relative entropy spread in the graph. It can be observed that these optimal transition probabilities (the optimal policy) do not depend on the initial, source, node (node 1) – they only depend on the destination node n. Equation (12) therefore defines the **optimal randomized policy** from any source node – it is actually the *local* policy counterpart of Equation (2) when paths are considered as independent [23].

Therefore, the one-step ahead probability distribution of finding the random walker in state k' at time step $t+1$ when following the optimal policy (12), $\rho^*_{t+1}(k')$, given that its distribution was $\rho^*_t(k)$ at time t is

$$\rho^*_{t+1}(k') = \sum_{k=1}^{n} p^*_{kk'} \rho^*_t(k) = \sum_{k=1}^{n} \frac{w_{kk'} z_{k'n}}{(z_{kn} - \delta_{kn})} \rho^*_t(k) \tag{13}$$

Let us now come back to the computation of the elements of the fundamental matrix, z_{kl}. Since the Equations (10-12) only involve the first row and the last column of matrix $\mathbf{Z}$, they can be easily computed by solving two systems of linear equations. For the last column, we solve $(\mathbf{I} - \mathbf{W})\mathbf{z}^b = \mathbf{e}_n$, where $\mathbf{z}^b$ is the column vector of so-called *backward variables* ($[\mathbf{z}^b]_k = z_{kn}$). Symmetrically, the column vector of *forward variables*, $\mathbf{z}^f$ (with $[\mathbf{z}^f]_k = z_{1k}$), containing the first row of matrix $\mathbf{Z}$, is provided by $(\mathbf{I} - \mathbf{W})^{\mathrm{T}}\mathbf{z}^f = \mathbf{e}_1$. Thus, $z^f_k = z_{1k}$ and $z^b_k = z_{kn}$. Written element-wise, this reads

$$\begin{cases} z^f_1 = 1 + \sum_{k \in P(1)} p^{(0)}_{k1} \exp[-\theta c_{k1}]\, z^f_k \\ z^f_{k'} = \sum_{k \in P(k')} p^{(0)}_{kk'} \exp[-\theta c_{kk'}]\, z^f_k, \text{ for } k' \neq 1 \end{cases} \tag{14}$$

and for the backward variables,

$$\begin{cases} z^b_n = 1 + \sum_{k' \in S(n)} p^{(0)}_{nk'} \exp[-\theta c_{nk'}]\, z^b_{k'} \\ z^b_k = \sum_{k' \in S(k)} p^{(0)}_{kk'} \exp[-\theta c_{kk'}]\, z^b_{k'}, \text{ for } k \neq n \end{cases} \tag{15}$$

Now, these backward variables $z^b_k = z_{kn}$ given by Equation (15) have an interesting, intuitive, interpretation [19]. Consider a new random walk defined by the transition probabilities (with a tilde)

$$\widetilde{p}_{kk'} = p^{(0)}_{kk'} \exp\left[-\theta c_{kk'}\right] \tag{16}$$

without any normalization. Since $\theta > 0$, the transition matrix $\widetilde{\mathbf{P}}$, containing the $\widetilde{p}_{kk'}$, is sub-stochastic. This means that, at each time step of the random walk, the random walker has a non-zero probability of abandoning the walk equal to $\widetilde{p}_{\text{eva},k} = (1 - \sum_{k' \in S(k)} \widetilde{p}_{kk'})$. We will say that Equation (16) defines an **evaporating random walk** (ERW) since the probability of seeing the random walker pursuing its quest decreases at each time step. In that case, the backward variable z^b_k from Equation (15) can be interpreted as the *average number of visits* to the goal state n before evaporation during the ERW, when starting from an intermediate state k at $t = 0$ [14], [28]. The quantity $\ln z^b_k$ will act as a potential (see Equation (53)).

In the same manner, one can show that the expected cost from Equation (7), when starting from the source node and following the optimal policy, is provided by

$$\overline{C} = \sum_{k=1}^{n} \sum_{k' \in S(k)} c_{kk'} n_{kk'} = \frac{1}{z_{1n}} \sum_{k=1}^{n} \sum_{k' \in S(k)} z_{1k} c_{kk'} w_{kk'} z_{k'n} \tag{17}$$

where $\infty . \exp[-\infty] = 0$ by convention. Let us now take the continuous state-space limit to the RSP.

III. The continuous state-space equivalent to RSP

The main objective of this paper is to investigate how the discrete RSP can be adapted to a continuous-state domain. In order to answer this question, let us consider a two-dimensional undirected lattice, with $c_{kk'} = c_{k'k}$, on which we apply the RSP framework. Each node has four neighbors as displayed in Figure (1). The idea will be to let the grid become dense by taking the limit $\epsilon \to 0$. The first step is to study the behavior of the forward/backward variables when taking this limit.

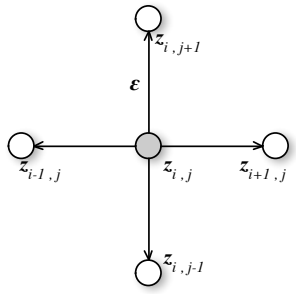

Fig. 1. Forward/backward variable in a grid configuration with 4 neighbors.

Let us recall that forward/backward variables are provided by Equations (14–15). Choosing uniform reference probabilities, $p^{(0)}_{kk'} = 1/4$ for all k, this reads

$$\begin{cases} z^f_1 = 1 + \sum_{k \in N(1)} \frac{1}{4} \exp[-\theta c_{k1}]\, z^f_k \\ z^f_{k'} = \sum_{k \in N(k')} \frac{1}{4} \exp[-\theta c_{kk'}]\, z^f_k, \text{ for } k' \neq 1 \end{cases} \tag{18}$$

where $N(k)$ is the set of neighbors of node k. Equations for the backward variables are obtained in the same way.

A. Computation of the forward/backward variables

We first consider the *forward equation*, Equation (18), and assume that each node on the grid is separated from its neighbors by a distance $\epsilon > 0$ (see Figure (1)). The forward variable z^f_k will be indexed by its position (x_k, y_k) and written as $z^f(x_k, y_k)$. In that case, the total cost along the path $\mathbf{r}(s) = (x(s), y(s))$ connecting node k to node k' is

$$c_{kk'} = \int_{(x_k, y_k)}^{(x_{k'}, y_{k'})} V(x(s), y(s)) ds \tag{19}$$

where $V(x,y) \geq 0$ is the *cost density* at (x,y) and s is the total displacement *along the trajectory* (its length). Note that, for the

sake of readability, we will also denote $V(x,y)$ – as well as the other variables – as $V_{x,y}$ or $V(\mathbf{r})$. In other words, it is assumed that the cost is only related to the position of the walker and not his direction – we therefore consider, for simplicity, that the cost $c_{kk'}$ is no more associated to the transition $k \to k'$, but only to the state k, $c_{kk'} = c_k$. Taking directions into account would require the use of tensors, which is not investigated in the present work.

As for any continuous-state stochastic process [3], [4], [9], [10], [12], [22], let us now assume that $\epsilon \to 0$ while maintaining the ratio $\epsilon^2/\delta s = c$ constant and finite, which means that in order to achieve a *net displacement* of ϵ, the random walker needs to make a total travel length of the order $\delta s \propto \epsilon^2$. This implies that the total length of the path followed by the random walker is of considerably larger magnitude than the final net displacement, ϵ [3]. When $\epsilon \to 0$, $c_{kk'} \simeq V(x_k, y_k)\delta s$ and Equation (18) can be rewritten for the grid of Figure (1) as

$$\begin{aligned} z^f_{x,y} &= \frac{\exp[-\theta V_{x,y}\delta s]}{4} z^f_{x+\epsilon,y} + \frac{\exp[-\theta V_{x,y}\delta s]}{4} z^f_{x-\epsilon,y} \\ &+ \frac{\exp[-\theta V_{x,y}\delta s]}{4} z^f_{x,y+\epsilon} + \frac{\exp[-\theta V_{x,y}\delta s]}{4} z^f_{x,y-\epsilon} \\ &= \frac{\exp[-\theta V_{x,y}\delta s]}{4} [z^f_{x+\epsilon,y} + z^f_{x-\epsilon,y} + z^f_{x,y+\epsilon} + z^f_{x,y-\epsilon}] \end{aligned} \quad (20)$$

Expanding each term up to the second order of ϵ, e.g.,

$$z^f_{x-\epsilon,y} = z^f_{x,y} - \frac{\partial z^f_{x,y}}{\partial y}\epsilon + \frac{1}{2}\frac{\partial^2 z^f_{x,y}}{\partial y^2}\epsilon^2 + o(\epsilon^3) \quad (21)$$

provides

$$z^f_{x,y} = \frac{e^{-[\theta V_{x,y}\delta s]}}{4} \left(4\, z^f_{x,y} + \frac{\partial^2 z^f_{x,y}}{\partial x^2}\epsilon^2 + \frac{\partial^2 z^f_{x,y}}{\partial y^2}\epsilon^2 + o(\epsilon^3)\right) \quad (22)$$

Keeping in mind that $\delta s = \epsilon^2/c$ and further expanding $e^{-[\theta V_{x,y}\delta s]} = (1 - \frac{\theta}{c}V_{x,y}\epsilon^2) + o(\epsilon^3)$, we obtain

$$\begin{aligned} z^f_{x,y} &= \frac{1}{4}(1 - \frac{\theta}{c}V_{x,y}\epsilon^2 + o(\epsilon^3)) \\ &\times \left(4\, z^f_{x,y} + \frac{\partial^2 z^f_{x,y}}{\partial x^2}\epsilon^2 + \frac{\partial^2 z^f_{x,y}}{\partial y^2}\epsilon^2 + o(\epsilon^3)\right) \end{aligned} \quad (23)$$

Without loss of generality, the constant c can be absorbed by θ: we now choose the units of θ in such a way that $c = 1$. Then, by defining the *diffusion constant* as $D = 1/(4\theta)$ and keeping only the terms in ϵ^2,

$$\frac{\partial^2 z^f_{x,y}}{\partial x^2} + \frac{\partial^2 z^f_{x,y}}{\partial y^2} = \frac{1}{D}V_{x,y} z^f_{x,y}\,, \text{ or } D\Delta z^f_{x,y} = V_{x,y}\, z^f_{x,y} \quad (24)$$

This is exactly the stationary solution of a Schrodinger-like diffusion equation without the imaginary term:

$$\mu\frac{\partial z^f(\mathbf{r},t)}{\partial t} = D\Delta z^f(\mathbf{r},t) - V(\mathbf{r})z^f(\mathbf{r},t) \quad (25)$$

where $V(\mathbf{r})$ plays the role of a potential and $\mathbf{r}(\tau) = (x(\tau), y(\tau))$. Equation (25) is also sometimes called the Bloch equation [4] in physics. It corresponds to a simple diffusion process for which the particle can disappear with a probability density $V(\mathbf{r})$ per unit of time at position $\mathbf{r}$, up to a normalization factor.

1) A diffusion process interpretation: There exists, indeed, an intuition related to general diffusion processes behind this equation [3], [4], [10]. The well-known first Fick's law states that particle flow, $\mathbf{j}$, of a diffusing material in any part of the system is proportional to the local density of particle gradient (e.g., [3], [4], [10]). In other words,

$$\mathbf{j}(\mathbf{r},t) = -D\boldsymbol{\nabla}\rho_t(\mathbf{r}) \quad (26)$$

where $\rho_t(\mathbf{r})$ is the particle density at time t and position $\mathbf{r} = (x,y)$, D is the diffusion constant, and $\mathbf{j}$ denotes the particle flow, i.e., $\mathbf{j}\cdot\mathbf{n}$, with $||\mathbf{n}|| = 1$, is the net number of diffusing particles per unit of time passing through position $\mathbf{r}$ in the direction of $\mathbf{n}$. Furthermore, if particles are neither created nor destroyed, the basic continuity relations [4], [10], [22] in two dimensions are verified (see [2] for standard notations)

$$\frac{\partial}{\partial t}\iint_\Omega \rho_t(\mathbf{r})\, dxdy = -\oint_{\partial\Omega} \mathbf{j}(\mathbf{r},t)\cdot d\boldsymbol{\sigma} \quad (27)$$

where $\partial\Omega$ is the region boundary and $d\boldsymbol{\sigma}$ is the infinitesimal contour vector directed to the outside of $\partial\Omega$. Or, equivalently, from the divergence theorem,

$$\frac{\partial\rho_t(\mathbf{r})}{\partial t} = -\text{div}\ \mathbf{j}(\mathbf{r},t) \quad (28)$$

Combining Fick's law with the continuity Equation (28) yields

$$\frac{\partial\rho_t(\mathbf{r})}{\partial t} = D\Delta\rho_t(\mathbf{r}) \quad (29)$$

Assume now that, instead of Equation (28), the density of particles is governed by

$$\frac{\partial\rho_t(\mathbf{r})}{\partial t} = -\text{div}\ \mathbf{j}(\mathbf{r},t) - V(\mathbf{r})\rho_t(\mathbf{r}) \quad (30)$$

which considers that particles are disappearing with a density $V(\mathbf{r})$ per unit of time [4]. This mimics the "evaporating" random walk behavior of the discrete RSP described in Section II-C. The resulting equation is

$$\frac{\partial\rho_t(\mathbf{r})}{\partial t} = D\Delta\rho_t(\mathbf{r}) - V(\mathbf{r})\rho_t(\mathbf{r}) \quad (31)$$

which is exactly Equation (25). In addition, when an external force $\mathbf{f}$ is present – implying a drift (velocity) in the direction of $\mathbf{f}$ – this results in an additional flow of the form

$$\mathbf{j}_f(\mathbf{r},t) = -\gamma\rho_t(\mathbf{r})\,\mathbf{f} \quad (32)$$

with γ being a mobility coefficient, the inverse of the friction coefficient [3], [4]. The flow $\mathbf{j}$ of Equation (26) therefore becomes $\mathbf{j}(\mathbf{r},t) = -D\boldsymbol{\nabla}\rho_t(\mathbf{r}) - \gamma\rho_t(\mathbf{r})\,\mathbf{f}$, yielding

$$\frac{\partial\rho_t(\mathbf{r})}{\partial t} = D\Delta\rho_t(\mathbf{r}) - \gamma\,\text{div}(\rho_t(\mathbf{r})\,\mathbf{f}) - V(\mathbf{r})\rho_t(\mathbf{r}) \quad (33)$$

instead of (31). This equation will be encountered later, when the optimal policy is derived (see Equation (53)). Interestingly, it can be shown that the solution to Equation (31) is provided by [4], [7], [20]

$$\mathbb{E}_W\{\exp[-\int_0^t V(x(\tau),y(\tau))d\tau]\} \quad (34)$$

where $\mathbb{E}_W$ represents the expectation according to the Wiener measure. This corresponds to the celebrated Feynman-Kac formula which states that the solution to (31) can be interpreted as the expectation on all possible paths, each path being weighted by the exponential of minus the total cost cumulated along the path. Therefore, low-cost paths are favored with respect to high-cost paths. The stochasticity of the process (the exploration) is regulated by the diffusion constant D. The discrete RSP can therefore be considered as a discrete-state discrete-time counterpart of the Feynman-Kac diffusion process as well as the Bloch equation.

2) Initial and boundary conditions: We still have to precise the initial conditions and the boundary conditions. By looking at Equation (18), it can be seen that there is a unit source at node 1. Denoting the position of this source node 1 as (x_f, y_f) (the subscript f is for forward), the source becomes a delta of Dirac centered at this location. The coefficient multiplying this delta of Dirac is computed in the Appendix, and is equal to $-4D$. The forward stationary equation (24) becomes

$$\boxed{\Delta z^f(x,y) = \frac{1}{D} V(x,y) z^f(x,y) - 4\delta(x-x_f)\delta(y-y_f)} \quad (35)$$

with $D = 1/(4\theta)$. It can be observed that D plays the same role as a temperature (inverse of θ).

Exactly the same reasoning applies to the *backward variable*, and the partial differential equation easily follows

$$\boxed{\Delta z^b(x,y) = \frac{1}{D} V(x,y) z^b(x,y) - 4\delta(x-x_b)\delta(y-y_b)} \quad (36)$$

Concerning the boundary conditions, a barrier is produced by defining an infinite cost on the boundaries, preventing the random walkers from reaching them. This allows to mimic the discrete situation of the RSP on a graph [23]. Dirichlet boundary conditions stating that both z^f and z^b are equal to zero on the boundary are therefore used. Thus, both in the continuous and the discrete case, an internal boundary layer $\partial\Omega$ is added with $V(x,y) = \infty$ for $(x,y) \in \partial\Omega$ in the continuous case, and $c_{kk'} = \infty$ for $k' \in \partial\Omega$ for the discrete one.

Notice that if we want to solve Equations (35-36) numerically by using a simple finite difference approximation with a central difference method, we exactly obtain Equations (14–15) with $\theta = f(D)$, some function of the diffusion constant. Let us for instance examine Equation (36). At the interior of the domain, $\Omega \backslash \partial\Omega$, the difference equation corresponding to (36) is

$$D(z^b_{x+\epsilon,y} + z^b_{x-\epsilon,y} + z^b_{x,y+\epsilon} + z^b_{x,y-\epsilon} - 4z^b_{x,y}) = \epsilon^2 V_{x,y} z^b_{x,y} \quad (37)$$

Isolating $z^b_{x,y}$ in this last equation provides

$$\left(\frac{\epsilon^2 V_{x,y}}{D} + 4\right) z^b_{x,y} = z^b_{x+\epsilon,y} + z^b_{x-\epsilon,y} + z^b_{x,y+\epsilon} + z^b_{x,y-\epsilon} \quad (38)$$

Assuming a small ϵ^2 in comparison with the value of D, this last equation is similar (up to the order $o(\epsilon^4)$) to

$$4\exp\left[\frac{\epsilon^2 V_{x,y}}{4D}\right] z^b_{x,y} = z^b_{x+\epsilon,y} + z^b_{x-\epsilon,y} + z^b_{x,y+\epsilon} + z^b_{x,y-\epsilon} \quad (39)$$

which, in turn, is equivalent to the discrete counterpart (15) as shown now. Indeed, considering a sufficient dense grid with a small ϵ, $c_{kk'} \simeq V(x_k, y_k)\delta s$, let us rewrite Equation (15) as

$$4\exp[\theta V(x_k, y_k)\,\delta s]\, z^b_k = \sum_{k' \in N(k)} z^b_{k'} \quad (40)$$

Evaluating the z^b_k on the grid yields

$$4\exp[\theta V_{x,y}\,\delta s]\, z^b_{x,y} = z^b_{x+\epsilon,y} + z^b_{x-\epsilon,y} + z^b_{x,y+\epsilon} + z^b_{x,y-\epsilon} \quad (41)$$

which, by using $\delta s = \epsilon^2$ and $D = 1/(4\theta)$, corresponds exactly to Equation (39).

Therefore, the RSP framework can be considered as the discrete-state counterpart of the continuous-state Feynman-Kac process. The expected number of visits at any position of the grid, as well as the expected cost, are derived in the next subsection.

B. Computation of some of the basic quantities

We are ready now to compute the quantities of interest for the continuous-state framework. The continuous-state equivalent of the discrete *partition function* $\mathcal{Z} = z_{1n}$ is $z^f(x_b, y_b) = z^b(x_f, y_f)$. From Equation (11), the *expected number of visits* to position (x,y) when following the optimal policy is

$$\boxed{\overline{n}(x,y) = \frac{z^f(x,y)\, z^b(x,y)}{z^f(x_b, y_b)}, \text{ for } (x,y) \neq (x_f, y_f)} \quad (42)$$

The *expected cost*, initially computed with Equation (17), for reaching (x_b, y_b) from (x_f, y_f) is provided by

$$\boxed{\overline{C} = \frac{\iint_\Omega z^f(x,y) z^b(x,y)\, V(x,y)\, dx\, dy}{z^f(x_b, y_b)}} \quad (43)$$

Let us now turn to the investigation of the optimal policy.

IV. The dynamic, global, optimal, policy

Let us consider the continuous-time, continuous-state, dynamics of a random walker following the optimal policy provided by Equation (12), or Equation (13) for the one-step ahead policy, in the discrete case. It will be assumed that $z^b(x,y)$ is known (computed through Equation (36)) and that the time is provided by the total displacement along the trajectory, i.e. $\delta t = \delta s$ (a unit velocity). The objective is to derive the probability density, $\rho^*_t(x,y)$, of finding the random walker in position (x,y) at time t when starting from some position $\rho^*_0(x,y) = \delta(x-x_0)\delta(y-y_0)$ and following the optimal randomized policy given by Equation (12), i.e., the continuous-state continuous-time counterpart of Equation (13).

It takes to the random walker a time $\delta t = \delta s = \epsilon^2$ to make a net displacement of ϵ so that taking a time step of δt and evaluating Equation (13) at $t + \delta t$ on position (x,y) of the two-dimensional grid (see Figure (1)), as well as assuming that (x,y) is in the interior of Ω so that the term δ_{kn} in the denominator of Equation (13) cancels, yields

$$\begin{aligned}\rho^*_{t+\delta t}(x,y) = &\frac{\exp[-\theta V_{x+\epsilon,y}\delta s]}{4} \frac{z^b_{x,y}}{z^b_{x+\epsilon,y}}\, \rho^*_t(x+\epsilon, y) \\ &+ \frac{\exp[-\theta V_{x-\epsilon,y}\delta s]}{4} \frac{z^b_{x,y}}{z^b_{x-\epsilon,y}}\, \rho^*_t(x-\epsilon, y) \\ &+ \frac{\exp[-\theta V_{x,y+\epsilon}\delta s]}{4} \frac{z^b_{x,y}}{z^b_{x,y+\epsilon}}\, \rho^*_t(x, y+\epsilon) \\ &+ \frac{\exp[-\theta V_{x,y-\epsilon}\delta s]}{4} \frac{z^b_{x,y}}{z^b_{x,y-\epsilon}}\, \rho^*_t(x, y-\epsilon) \end{aligned} \quad (44)$$

Remembering that $\delta s = \delta t = \epsilon^2$, and expanding $\rho^*_{t+\delta t}(x,y)$ as well as the exponentials up to ϵ^2 gives

$$\begin{aligned}\rho^*_t(x,y) + \frac{\partial \rho^*_t(x,y)}{\partial t}\epsilon^2 + o(\epsilon^3) = &\frac{z^b_{x,y}\left(1 - \theta V_{x,y}\epsilon^2 + o(\epsilon^3)\right)}{4} \\ \times \underbrace{\left[\underbrace{\frac{\rho^*_t(x+\epsilon,y)}{z^b_{x+\epsilon,y}}}_{(i)} + \underbrace{\frac{\rho^*_t(x-\epsilon,y)}{z^b_{x-\epsilon,y}}}_{(ii)} + \underbrace{\frac{\rho^*_t(x,y+\epsilon)}{z^b_{x,y+\epsilon}}}_{(iii)} + \underbrace{\frac{\rho^*_t(x,y-\epsilon)}{z^b_{x,y-\epsilon}}}_{(iv)}\right]}_{[(i)+(ii)+(iii)+(iv)]}\end{aligned} \quad (45)$$

We now develop the terms (i)-(iv) appearing in the second line of the previous equation (45). For instance, the first term (i) is

$$\frac{\rho_t^*(x+\epsilon,y)}{z^b_{x+\epsilon,y}} = \frac{\rho_t^*(x,y)}{z^b_{x,y}} + \frac{\partial}{\partial x}\left(\frac{\rho_t^*(x,y)}{z^b_{x,y}}\right)\epsilon + \frac{1}{2}\frac{\partial^2}{\partial x^2}\left(\frac{\rho_t^*(x,y)}{z^b_{x,y}}\right)\epsilon^2 + o(\epsilon^3) \quad (46)$$

We immediately observe that terms (i)-(iv) of order ϵ in Equation (45) cancel out because they are evaluated both at $+\epsilon$ and $-\epsilon$. Therefore, dropping the dependency on (x,y) and referring $\partial z(x,y)/\partial x$ as $\partial_x z$ for convenience,

$$[(i)+...+(iv)] = 4\frac{\rho_t^*}{z^b} + \left[\partial_x^2\left(\frac{\rho_t^*}{z^b}\right) + \partial_y^2\left(\frac{\rho_t^*}{z^b}\right)\right]\epsilon^2 + o(\epsilon^3) \quad (47)$$

For the second derivative term in the previous equation (47), we obtain

$$\partial_x^2\left(\frac{\rho_t^*}{z^b}\right) = \frac{(\partial_x^2\rho_t^*)(z^b)^2 + 2\rho_t^*(\partial_x z^b)^2}{(z^b)^3} - \frac{2(\partial_x\rho_t^*)(\partial_x z^b) + \rho_t^*(\partial_x^2 z^b)}{(z^b)^2} \quad (48)$$

and the corresponding formula for $\partial_y^2(\rho_t^*/z^b)$ is obtained by substituting y for x in Equation (48). Replacing the values of these second derivatives in Equation (47) yields

$$[(i)+(ii)+(iii)+(iv)] = o(\epsilon^3) + 4\frac{\rho_t^*}{z^b} + \left[\frac{\Delta\rho_t^*}{z^b} + 2\rho_t^*\frac{||\boldsymbol{\nabla} z^b||^2}{(z^b)^3} - 2\frac{\boldsymbol{\nabla}\rho_t^*.\boldsymbol{\nabla} z^b}{(z^b)^2} - \rho_t^*\frac{\Delta z^b}{(z^b)^2}\right]\epsilon^2 \quad (49)$$

so that Equation (45) becomes

$$\rho_t^*(x,y) + \frac{\partial\rho_t^*(x,y)}{\partial t}\epsilon^2 + o(\epsilon^3) = \left(1-\theta V_{x,y}\epsilon^2 + o(\epsilon^3)\right)\times \left[\rho_t^* + \left(\frac{\Delta\rho_t^*}{4} + \rho_t^*\frac{||\boldsymbol{\nabla} z^b||^2}{2(z^b)^2} - \frac{\boldsymbol{\nabla}\rho_t^*.\boldsymbol{\nabla} z^b}{2z^b} - \rho_t^*\frac{\Delta z^b}{4z^b}\right)\epsilon^2 + o(\epsilon^3)\right]$$

Keeping the terms up to the second order in ϵ provides

$$\frac{\partial\rho_t^*}{\partial t} = -\theta V\rho_t^* + \frac{\Delta\rho_t^*}{4} + \rho_t^*\frac{||\boldsymbol{\nabla} z^b||^2}{2(z^b)^2} - \frac{\boldsymbol{\nabla}\rho_t^*.\boldsymbol{\nabla} z^b}{2z^b} - \underbrace{\rho_t^*\frac{\Delta z^b}{4z^b}}_{\rho_t^*\frac{\Delta z^b}{4z^b} - \rho_t^*\frac{\Delta z^b}{2z^b}} = \frac{\Delta\rho_t^*}{4} + \rho_t^*\left(\frac{\Delta z^b}{4z^b} - \theta V\right) - \frac{1}{2}\left(-\rho_t^*\frac{||\boldsymbol{\nabla} z^b||^2}{(z^b)^2} + \frac{\boldsymbol{\nabla}\rho_t^*.\boldsymbol{\nabla} z^b}{z^b} + \rho_t^*\frac{\Delta z^b}{z^b}\right) \quad (50)$$

Noticing that $D = 1/(4\theta)$ and rewriting Equation (36) as

$$\frac{\Delta z^b}{4z^b} - \theta V = -\frac{\delta(x-x_b)\delta(y-y_b)}{z^b} \quad (51)$$

then combining this last result with

$$-\rho_t^*\frac{||\boldsymbol{\nabla} z^b||^2}{(z^b)^2} + \frac{\boldsymbol{\nabla}\rho_t^*.\boldsymbol{\nabla} z^b}{z^b} + \rho_t^*\frac{\Delta z^b}{z^b} = \mathrm{div}(\rho_t^*\boldsymbol{\nabla}\ln z^b), \quad (52)$$

we finally obtain for the optimal policy (50):

$$\boxed{\frac{\partial\rho_t^*(x,y)}{\partial t} = \frac{\Delta\rho_t^*(x,y)}{4} - \frac{1}{2}\mathrm{div}(\rho_t^*(x,y)\boldsymbol{\nabla}\ln z^b(x,y)) - \frac{\rho_t^*(x_b,y_b)\delta(x-x_b)\delta(y-y_b)}{z^b(x_b,y_b)}} \quad (53)$$

which corresponds to a Bloch diffusion equation with a drift (diffusion process subject to an external force) [4] as in Equation (33). The initial condition at $t=0$ is centered on the point of interest: $\rho_0^*(x,y) = \delta(x-x_0)\delta(y-y_0)$. The first term on the right-hand side of Equation (53) is a diffusion term with $\theta = 1$ or $D = 1/4$, the second one corresponds to a drift driven by the force $\mathbf{f} = \boldsymbol{\nabla}\ln z^b(x,y)$ (see Equation (33)) with $\gamma = 1/2$ and where $\ln z^b(x,y)$ plays the role of a **potential** known in advance (it is provided by the solution of Equation (36)); eventually, the last term corresponds to an absorption (sink) of the probability density in the goal state.

Therefore, the algorithm for computing the optimal randomized policy at a point of interest (x_0,y_0) and a time t is the following:

1) Find the backward variable $z^b(x,y)$ by solving Equation (36) with respect to $z^b(x,y)$. $\ln z^b(x,y)$ is the associated potential.
2) Compute the optimal randomized policy $\rho_t^*(x,y)$, providing the probability of jumping to position (x,y) from position (x_0,y_0) during time step t, by solving Equation (53) with respect to $\rho_t^*(x,y)$ at the point of interest (x_0,y_0) and for the predefined length/time t.

On the other hand, the best direction to follow corresponds to the orientation of the steepest ascent of $\ln z^b$, for which the gradient is maximum. Thus, the *deterministic* optimal policy tells us that we have to move in the direction of $\boldsymbol{\nabla}\ln z^b(x,y)$ at any point (x,y). This very simple rule will provide the minimal cost direction when θ is sufficiently large, and thus D is low – in which case the paths probability distribution is peaked on the shortest paths. Interestingly, this policy is similar to an existing technique for mobile robot path planning [25], [24]. The authors use a diffusion equation akin to Equation (36) (they, however, assume a constant $V(x,y)$) and propose to follow the steepest ascent of the solution. They claim that this technique provides the shortest path to the goal state, but no proof is provided. These techniques are known as potential function techniques in the field of robotics.

V. Simulations

A series of simulations have been performed in order to investigate the behavior of our method. The first series of simulations (see Figures (2–4)) illustrate the influence of the diffusion coefficient on a diffusion media with a simple Gaussian obstacle. For each value of $D = \{1, 0.1, 0.01\}$, four results (drawings) are reported in a 40×40 grid of nodes, red being the highest probability of passing through the node, blue being the lowest: (i) the Gaussian obstacle, (ii) the logarithm of the forward variable computed from Equation (35), (iii) the logarithm of the backward variable computed from Equation (36), and (iv) the average number of visits per node (Equation (42)). Both the forward and backward variables have been calculated by means of a finite difference approximation.

A second series of simulations on a simple 36×36 maze (inspired from the one used in [6]) with two obstacles of varying length have also been performed. In this case, a single-source multiple-destination maze problem with a constant diffusion parameter, $D = 0.1$, is solved. In this second case, the influence of the length of the obstacle determines the shortest path and, thus, the chosen goal node. Results are reported in Figures (5–7). It must be noted that when both goals are equidistant from the source node (see Figure (6)) both solution paths will be almost equally explored. On the remaining two cases (see Figures (5) and (7)), the shortest goal will be favored over the farthest one. It must be noted that, although this is a single-source multiple-destination problem, our method extends to the case of multiple-source multiple-destination problems.

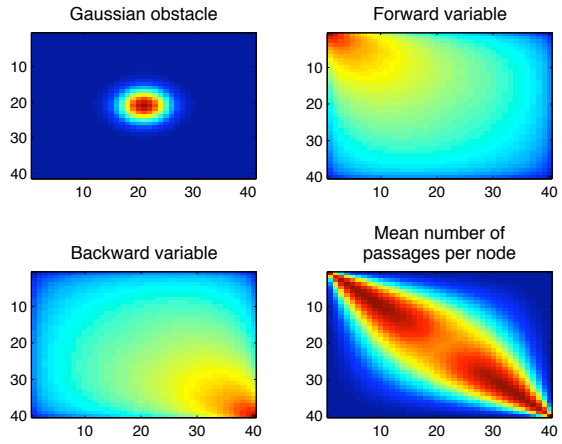

Fig. 2. Continuous RSP with Gaussian obstacle and $D = 1$.

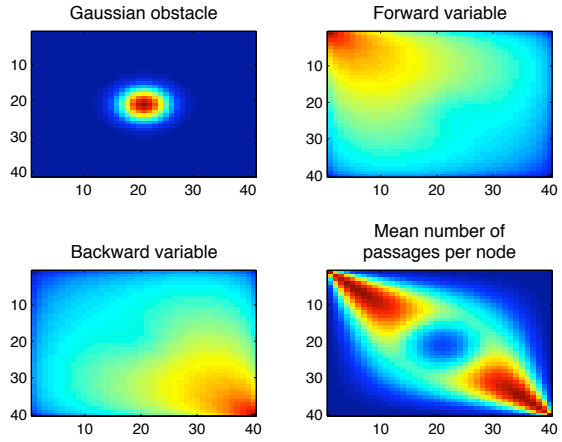

Fig. 3. Continuous RSP with Gaussian obstacle and $D = 0.1$.

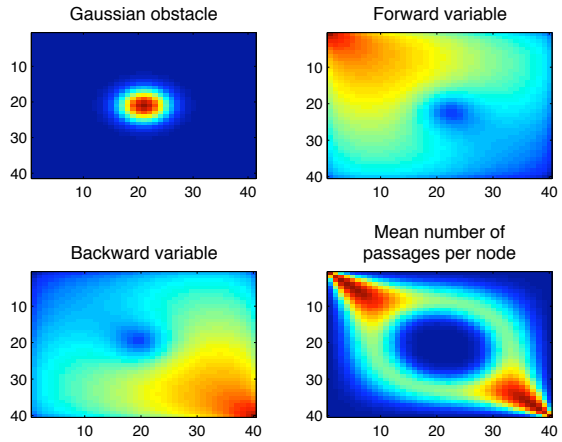

Fig. 4. Continuous RSP with Gaussian obstacle and $D = 0.01$.

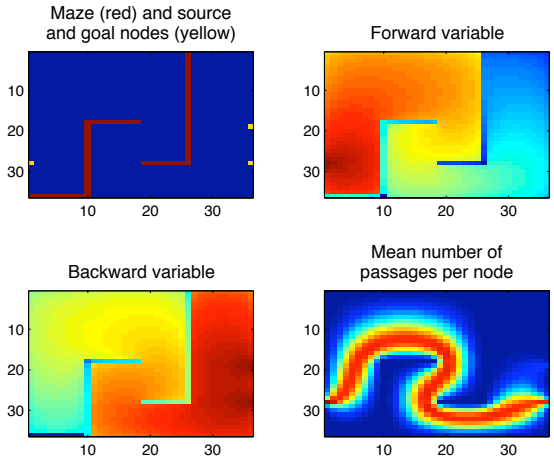

Fig. 5. First maze with one source (on the left of the drawing) and two goal nodes (on the right of the drawing), with $D = 0.1$.

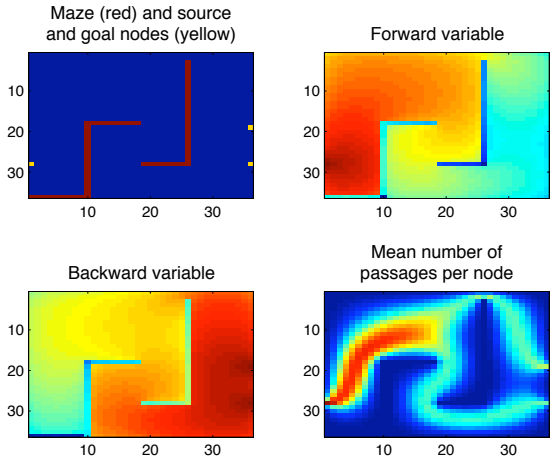

Fig. 6. Second maze with the same source and goal nodes, with $D = 0.1$.

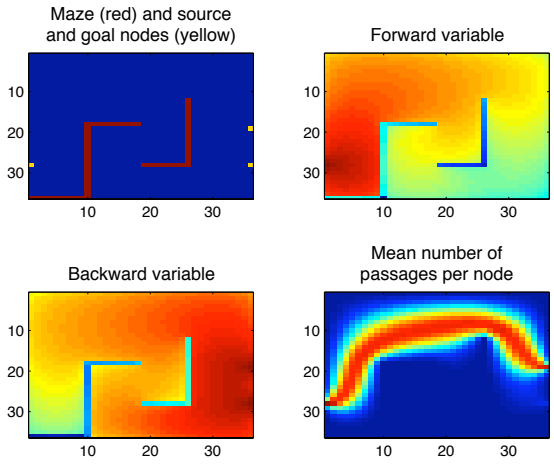

Fig. 7. Third maze with the same source and goal nodes, with $D = 0.1$.

VI. Conclusions

This work investigated the continuous-state counterpart of the discrete randomized shortest-path on a graph. It allowed to set some bridges between biased random walks on a discrete graph and the continuous-state Feynman-Kac diffusion process. From the application side, it provides an optimal randomized policy for solving continuous-state path planning problems with multiple sources and multiple destinations. The main drawback of this model is that it assumes that paths are independent, which is hardly the case in practice. Further work will study the possibility of introducing a mass parameter (inertia) for smoothing the trajectories, therefore avoiding abrupt changes in direction. Indeed, the Wiener measure naturally accounts for a kinetic energy term cumulated along the trajectory [4], [20]. This would also enhance the physical interpretation of the model.

References

[1] T. Akamatsu. Cyclic flows, Markov process and stochastic traffic assignment. *Transportation Research B*, 30(5):369–386, 1996.

[2] G. Arfken and H. Weber. *Mathematical methods for physics*. Elsevier, 2005.

[3] H. Berg. *Random walks in biology*. Princeton University Press, 1993.

[4] M. Chaichian and A. Demichev. *Path integrals in physics, vol 1: stochastic processes and quantum mechanics*. Institute of Physics Publishing, 2001.

[5] C. Connolly, J. Burns, and R. Weiss. Path planning using Laplace's equation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 2102–2106, 1990.

[6] K. Dautenhahn and H. Cruse. Computer simulations of path generation and path form modification with local rules working on a parallel cell-based architecture. *Computers & Mathematics with Applications*, 28(5):75 – 88, 1994.

[7] P. Del Moral. *Feynman-Kac formulae: genealogical and interacting particle systems with applications*. Springer, 2004.

[8] S. García-Díez, F. Fouss, M. Shimbo, and M. Saerens. A sum-over-paths extension of edit distances accounting for all sequence alignments. *Pattern Recognition*, 44(6):1172 – 1182, 2011.

[9] C. W. Gardiner. *Handbook of stochastic methods*. Springer, 2002.

[10] M. Holmes. *Introduction to the foundations of applied mathematics*. Springer, 2002.

[11] Y. Hwang and N. Ahuja. A potential field approach to path planning. *IEEE Transactions on Robotics and Automation*, 8(1):23–32, 1992.

[12] K. Jacobs. *Stochastic processes for physicists: understanding noisy systems*. Cambridge University Press, 2010.

[13] E. T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4):620–630, 1957.

[14] J. G. Kemeny and J. L. Snell. *Finite Markov chains, 2nd printing*. Springer-Verlag, 1976.

[15] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–98, 1986.

[16] S. M. LaValle. *Planning algorithms*. Cambridge University Press, 2006.

[17] Z. X. Li and T. D. Bui. Robot path planning using fluid model. *Journal of Intelligent and Robotics Systems*, 21(1):29–50, 1998.

[18] C. Louste and A. Liegeois. Near optimal robust path planning for mobile robots: the viscous fluid method with friction. *Journal of Intelligent and Robotics Systems*, 27(1-2):99–112, 2000.

[19] A. Mantrach, L. Yen, J. Callut, K. Francoisse, M. Shimbo, and M. Saerens. The sum-over-paths covariance kernel: A novel covariance measure between nodes of a directed graph. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1112–1126, 2010.

[20] R. M. Mazo. *Brownian motion: fluctuations, dynamics, and applications*. Oxford University Press, 2002.

[21] N. G. Rambidi and D. Yakovenchuck. Finding paths in a labyrinth based on reaction-diffusion media. *Biosystems*, 51(2):67 – 72, 1999.

[22] J. Rudnick and G. Gaspari. *Elements of the random walk*. Cambridge University Press, 2004.

[23] M. Saerens, Y. Achbany, F. Fouss, and L. Yen. Randomized shortest-path problems: Two related models. *Neural Computation*, 21(8):2363–2404, 2009.

[24] G. Schmidt and K. Azarm. Mobile robot navigation in a dynamic world using an unsteady diffusion equation strategy. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 642–647, 1992.

[25] G. Schmidt and W. Neubauer. High-speed robot path planning in time-varying environment employing a diffusion equation strategy. In S. G. Tzafestas, editor, *Robotic systems: advanced techniques and applications*, pages 207–215. Kluwer Academic Publishers, 1992.

[26] L. Steels. Exploiting analogical representations. *Robotics and Autonomous Systems*, 6(1-2):71–88, June 1990.

[27] L. Tarassenko and A. Blake. Analogue computation of collision-free paths. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 540–545, 1991.

[28] H. M. Taylor and S. Karlin. *An introduction to stochastic modeling, 3th Ed.* Academic Press, 1998.

[29] L. Yen, A. Mantrach, M. Shimbo, and M. Saerens. A family of dissimilarity measures between nodes generalizing both the shortest-path and the commute-time distances. In *Proceedings of the 14th SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2008)*, pages 785–793, 2008.

Appendix

Let us derive the source term coefficient of Equations (35-36). Rewriting the discrete equation for the forward variable on the grid (Equation (20)) by including the source term gives

$$z^f_{x,y} = \frac{\exp[-\theta V_{x,y}\delta s]}{4}[z^f_{x+\epsilon,y} + z^f_{x-\epsilon,y} + z^f_{x,y+\epsilon} + z^f_{x,y-\epsilon}] + \delta(x,x_f)\delta(y,y_f) \quad (54)$$

where $\delta(x,x_f), \delta(y,y_f)$ are Kronecker deltas. Expanding the different terms up to the second order as in Section III-A, Equation (23), yields

$$z^f_{x,y} = \frac{\left(1-\theta V_{x,y}\epsilon^2 + o(\epsilon^3)\right)}{4}\left[4\, z^f_{x,y} + \Delta z^f_{x,y}\epsilon^2 + o(\epsilon^3)\right] + \delta(x,x_f)\delta(y,y_f) \quad (55)$$

Thus, by rearranging this last equation, we obtain

$$\delta(x,x_f)\delta(y,y_f) = (\theta V_{x,y} z^f_{x,y} - \frac{1}{4}\Delta z^f_{x,y})\epsilon^2 + o(\epsilon^3) \quad (56)$$

Summing this last equation over the entire grid and taking the limit $\epsilon \to 0$ with $dx = dy = \epsilon$ provides

$$1 = \sum_{x,y}\delta(x,x_f)\delta(y,y_f) \quad (57)$$

$$= \sum_{x,y}[(\theta V_{x,y} z^f_{x,y} - \frac{1}{4}\Delta z^f_{x,y})\epsilon^2 + o(\epsilon^3)] \quad (58)$$

$$\simeq \frac{1}{\epsilon^2}\int_{x,y} dxdy\,[(\theta V_{x,y} z^f_{x,y} - \frac{1}{4}\Delta z^f_{x,y})\epsilon^2 + o(\epsilon^3)] \quad (59)$$

$$= \int_{x,y} dxdy\,(\theta V_{x,y} z^f_{x,y} - \frac{1}{4}\Delta z^f_{x,y}) + o(\epsilon) \quad (60)$$

Therefore, $\theta V_{x,y} z^f_{x,y} - \frac{1}{4}\Delta z^f_{x,y} = \delta(x-x_f)\delta(y-y_f)$, with $\delta(x-x_f), \delta(y-y_f)$ being Dirac deltas. Since $D = 1/(4\theta)$, we finally obtain Equation (35)

$$D\Delta z^f_{x,y} = V_{x,y} z^f_{x,y} - 4D\delta(x-x_f)\delta(y-y_f) \quad (61)$$

Acknowledgments

This work is partially supported by the *Fonds pour la formation à la Recherche dans l'Industrie et dans l'Agriculture* (F.R.I.A.) under grant reference F3/5/5-MCF/ROI/BC-21716. Part of this work has also been funded by projects with the "Région Wallonne". We thank this institution for giving us the opportunity to conduct both fundamental and applied research.