

Monte Carlo Shortest Paths for Physical Agents

Juan-Antonio Fernández-Madrigal
Tech. Report, March 2008

System Engineering and Automation Dpt.
University of Málaga

1. Introduction

In this report we review the problem of searching shortest paths in directed graphs where the weights of the edges are positive probabilistic measures, with a perspective on application to physical agents (for a more in-depth and theoretical analysis see [1] and [2]). In particular, we use weights that are samples drawn from some stochastic process, for example produced by a mobile robot navigating between two distinctive places of a topological map able to measure time, energy, distance, etc. after navigation. In that kind of applications it is difficult to assume any specific model in the probability distributions of weights (although normality may be reasonable for complex processes as long as negative values are permitted). We just assume that the weight of an edge is independent on the weights of any other; they may be even drawn from probability distributions with different shapes and parameters. Under those assumptions, we review the most suitable approaches for stochastic shortest paths, and show how to adapt them to Monte Carlo sampled distributions.

2. The Stochastic Graph Model

There is a wide literature in the subject of stochastic shortest paths in directed graphs. The general model is as follows.

Def. 1. A directed, Monte Carlo stochastically-weighted graph (MC-graph from now on) is a tuple $G=(V,E,\rho,\lambda,\hat{\rho})$, where V is a finite (possibly empty) set of vertices (nodes), E is a finite (possibly empty) set of non-reflexive, non-parallel edges (arcs), ρ is the unknown ground truth probability distribution from which the weights of edges are drawn, ω is the history of weights actually sampled from that distribution, and $\hat{\rho}$ is a probability estimation of ρ based on that history. More specifically¹,

$$V=\{v_1, v_2, \dots, v_{|W|}\} \quad , \quad |W| \in \mathbb{N}^+ \quad (2.1)$$

$$E=\{e_1, e_2, \dots, e_{|E|}\} \quad , \quad |E| \in \mathbb{N} \quad , \quad \forall i, j \in [1, |E|]: e_i=(v_r, v_s) \in V \times V, \quad r \neq s \quad , \quad e_i=e_j \Rightarrow i=j$$

$$\rho: E \rightarrow \mathbb{N}$$

$$\rho(e_i) \equiv \rho_{e_i}(x), x \in \mathbb{R}^+ \cup \{0\}$$

$$\omega: E \rightarrow 2^{\mathbb{R}^+ \cup \{0\}}$$

$$\forall e_i \in E, \omega(e_i) = \{w_{i1}, w_{i2}, \dots, w_{in}\}, \quad n \in \mathbb{N}, \quad \forall j \in [1, n]: w_{ij} \sim \rho_{e_i}(x)$$

¹ $\mathbb{N}$ is the set of all the probability density functions that have support in the positive real line ($\mathbb{R}^+ \cup \{0\}$). Expression 2^A denotes the powerset of the set A .

$$\hat{p}: E \times \mathbb{R} \rightarrow \mathbb{R}$$

$$\hat{p}(e_i, x) \equiv \hat{p}_{e_i}(x) = \frac{1}{|\omega(e_i)|} \sum_{w_y \in \omega(e_i)} \delta_{w_y}(x)$$

Thus the weight of an edge can be measured through an unknown stochastic process that produces positive real numbers according to some unknown probability density function ρ (which can include not only stochastic effects internal to the process but also partial observability). The graph also holds the history of weights produced by that process up to the moment, represented for each edge e_i by a (possibly empty) set $\omega(e_i)$. A Monte Carlo estimation of the stochastic weight of an edge can be formed from the sampled weights in the history of that edge by using the estimator $\hat{p}$ (notice that in a Monte Carlo representation of a pdf, all the samples are considered equally probable; the sum of their importances is $\sum_{w_y \in \omega(e_i)} \frac{1}{|\omega(e_i)|} = 1$). The notation $\delta_a(x)$ is a Dirac's delta defined on all the possible values of x with the peak at $x=a$. By the law of large numbers, the Monte Carlo estimation approaches better the ground truth value of ρ as the number of samples increases.

3. Probabilistic Shortest Path

In this section we focus on different ways of obtaining optimal paths in MC-graphs, first by maintaining all the information of the stochastic weights, then (more efficiently but with higher risk) by reducing information and representing only the first moments of the weights.

3.1. Using the Complete Stochastic Weights

In this subsection we explore the direct use of the stochastic weights without simplifying them to single values. Without reducing weights to single values, there are a number of possible optimality indexes that can be defined on weights (see [2]). We will employ the one presented in [3] that consists of considering optimal the path that is most probably the shortest one².

Sum of Weights

The first basic requirement for a shortest path algorithm is to be able to sum the weights along a given path (sequence of edges). In our case, that should be done by summing up the corresponding probability distributions given by ρ , but since that is not known, we estimate it by using the samples gathered by $\hat{p}$ and making up a new distribution for the sum of weights. The formal way to obtain that estimation is as follows. Let say that the random variable x_r represents the weights sampled from $\rho_{e_r}(x)$. Then we can define a new random variable which is the sum of two of these r.v.s., let say $u = x_r + x_s$. If we also define another auxiliary r.v. $v = x_s$ then the pair (u, v) is biunivocally linked to the pair (x_r, x_s) and thus we can set up the inverse relations $x_r = u - v$ and $x_s = v$ as functions. We are interested in the probability distribution of u , which can be obtained by marginalization over v of the joint probability distribution of (u, v) . In turn, this latter distribution can be deduced from the joint probability distribution of (x_r, x_s) by using the defined inverse relations [5]:

2 A different perspective would be to use a Markov Decision Process model to solve the problem; however it only allows for obtaining an optimal path with respect to the expected weights of edges (rewards), and thus most of the stochastic information would be lost [4]. Still another perspective is to use a linear programming approach to find the path that is shortest with a probability greater than a given threshold, but that is demonstrated to be sensitive to the length of the paths [2].

$$p_{uv}(u, v) = p_{x_r, x_s}(u - v, v) |J| \quad (3.1)$$

where J is the Jacobian matrix of the vector (x_r, x_s) with respect to the vector (u, v) , and $|J| = \begin{vmatrix} \frac{\partial x_r}{\partial u} & \frac{\partial x_r}{\partial v} \\ \frac{\partial x_s}{\partial u} & \frac{\partial x_s}{\partial v} \end{vmatrix}$ is its determinant. With our definition of (u, v) , that determinant is 1, and since (x_r, x_s) are independent r.v.s.,

$$p_{uv}(u, v) = p_{x_r}(u - v) p_{x_s}(v) \simeq \left[\frac{1}{|\omega(e_r)|} \sum_{w_r \in \omega(e_r)} \delta_{w_r}(u - v) \right] \left[\frac{1}{|\omega(e_s)|} \sum_{w_s \in \omega(e_s)} \delta_{w_s}(v) \right] = \frac{1}{|\omega(e_r)| |\omega(e_s)|} \left[\sum_{w_r \in \omega(e_r)} \delta_{w_r}(u - v) \right] \left[\sum_{w_s \in \omega(e_s)} \delta_{w_s}(v) \right] \quad (3.2)$$

Now the probability of the sum we looked for is:

$$p_{x_r + x_s}(x_r + x_s) = p_u(u) \simeq \frac{1}{|\omega(e_r)| |\omega(e_s)|} \int_{u=0}^{u=+\infty} \left[\sum_{w_r \in \omega(e_r)} \delta_{w_r}(u - v) \right] \left[\sum_{w_s \in \omega(e_s)} \delta_{w_s}(v) \right] dv \quad (3.3)$$

By using the fact that $\delta_b(x - a) = \delta_{b+a}(x)$ and the same Dirac's behavior described in footnote 7,

$$\begin{aligned} p_{x_r + x_s}(x_r + x_s) &\simeq \frac{1}{|\omega(e_r)| |\omega(e_s)|} \int_{v=0}^{v=+\infty} \sum_{w_r \in \omega(e_r)} \left[\delta_{w_r+v}(u) \sum_{w_s \in \omega(e_s)} \delta_{w_s}(v) \right] dv = \\ &= \frac{1}{|\omega(e_r)| |\omega(e_s)|} \sum_{w_r \in \omega(e_r)} \sum_{w_s \in \omega(e_s)} \int_{v=0}^{v=+\infty} \delta_{w_r+v}(u) \delta_{w_s}(v) dv = \\ &= \frac{1}{|\omega(e_r)| |\omega(e_s)|} \sum_{w_r \in \omega(e_r)} \sum_{w_s \in \omega(e_s)} \delta_{w_r+w_s}(x_r + x_s) = \\ &= \frac{1}{|\omega(e_r)| |\omega(e_s)|} \sum_{(w_r, w_s) \in \omega(e_r) \times \omega(e_s)} \delta_{w_r+w_s}(x_r + x_s) \end{aligned} \quad (3.4)$$

which is a new Monte Carlo distribution with new samples (more than the original pdfs, concretely $|\omega(e_r)| |\omega(e_s)|^3$).

Equation (3.4) defines an addition operator for sampled pdfs which is both commutative and associative. In general, the sum of the weights associated to a sequence of edges $\langle e_1, e_2, \dots, e_k \rangle$, that is, the stochastic weight of a path, as long as its edges are not repeated⁴, is:

$$p_{\sum_{i=1}^k x_i}(\sum_{i=1}^k x_i) \simeq \frac{1}{\prod_{i=1}^k |\omega(e_i)|} \sum_{(w_{1i_1}, w_{2i_2}, \dots, w_{ki_k}) \in \omega(e_1) \times \omega(e_2) \times \dots \times \omega(e_k)} \delta_{\sum_{i=1}^k w_{i_i}}(\sum_{i=1}^k x_i) \quad (3.5)$$

Notice that this operator involves an exponential growth of the number of samples for representing the result. We do not deal with techniques for reducing that (while losing precision) since, as we will see further on, there is no tractable algorithm for finding shortest paths using these distributions.

Ordering of Weights, Optimality of Paths, and the Dynamic Programming Assumption

The second basic requirement for a shortest path algorithm is that there exists an order in the weights

³ Notice that $\sum_{(w_r, w_s) \in \omega(e_r) \times \omega(e_s)} \frac{1}{|\omega(e_r)| |\omega(e_s)|} = 1$ as long as $\sum_{w_r \in \omega(e_r)} \frac{1}{|\omega(e_r)|} = \sum_{w_s \in \omega(e_s)} \frac{1}{|\omega(e_s)|} = 1$.

⁴ Otherwise the independence assumption between weights of edges is violated.

of the edges, and therefore in their sum, for deciding the optimality of a path with respect to others. In addition, if efficient globally optimal path searching is intended through dynamic programming⁵ then a path being optimal must imply that its subpaths are also optimal, therefore the ordering of weights being a total order is a sufficient condition. Unfortunately, that is not true in a MC-graph since weights are not real numbers but sampled probability distributions. We will see that when using the most probably shortest path as optimality index, the dynamic programming assumption does not hold and therefore an exhaustive (possibly computationally intractable) search must be carried out.

Our weight ordering is based on the probability of the weight of an edge to be smaller than or equal to the weight of another, which, assuming independence between samples, is⁶:

$$\forall r, s \in [1, |E|], \forall x \sim \rho_{e_r}, y \sim \rho_{e_s}: p_{xy}(x \leq y) = \int_{x=0}^{x=+\infty} \int_{y=x}^{y=+\infty} p_{xy}(x, y) dx dy = \int_{x=0}^{x=+\infty} p_x(x) \left[\int_{y=x}^{y=+\infty} p_y(y) dy \right] dx \quad (3.6)$$

In the case of a Monte Carlo representation of the probability distributions, that exact value can be approximated as⁷:

$$\begin{aligned} \forall r, s \in [1, |E|], \forall x \sim \rho_{e_r}, y \sim \rho_{e_s}: p_{xy}(x \leq y) &\simeq \int_{x=0}^{x=+\infty} \frac{1}{|\omega(e_r)|} \sum_{w_{ri} \in \omega(e_r)} \delta_{w_{ri}}(x) \left[\int_{y=x}^{y=+\infty} \frac{1}{|\omega(e_s)|} \sum_{w_{sj} \in \omega(e_s)} \delta_{w_{sj}}(y) dy \right] dx \\ &= \int_{x=0}^{x=+\infty} \frac{1}{|\omega(e_r)|} \sum_{w_{ri} \in \omega(e_r)} \delta_{w_{ri}}(x) \left[\frac{1}{|\omega(e_s)|} \sum_{w_{sj} \in \omega(e_s)} \int_{y=x}^{y=+\infty} \delta_{w_{sj}}(y) dy \right] dx \\ &= \int_{x=0}^{x=+\infty} \frac{1}{|\omega(e_r)|} \sum_{w_{ri} \in \omega(e_r)} \delta_{w_{ri}}(x) \left[\frac{1}{|\omega(e_s)|} \sum_{w_{sj} \in \omega(e_s): w_{sj} \geq x} 1 \right] dx \\ &= \frac{1}{|\omega(e_r)|} \sum_{w_{ri} \in \omega(e_r)} \int_{x=0}^{x=+\infty} \delta_{w_{ri}}(x) \left[\frac{1}{|\omega(e_s)|} |\{w_{sj} \in \omega(e_s): w_{sj} \geq x\}| \right] dx \\ &= \frac{1}{|\omega(e_r)|} \sum_{w_{ri} \in \omega(e_r)} \left[\frac{1}{|\omega(e_s)|} |\{w_{sj} \in \omega(e_s): w_{sj} \geq w_{ri}\}| \right] \\ &= \frac{1}{|\omega(e_r)| |\omega(e_s)|} \sum_{w_{ri} \in \omega(e_r)} |\{w_{sj} \in \omega(e_s): w_{sj} \geq w_{ri}\}| \end{aligned} \quad (3.7)$$

or in words: the probability that any measured weight of an edge is smaller than or equal to any measured weight of another edge is proportional to the number of sampled weights from the second edge that are greater than or equal to each sampled weight of the first edge. If instead of “smaller than or equal to” we want an stricter “smaller than”, the estimation would be:

$$p_{xy}(x < y) \simeq \frac{1}{|\omega(e_r)| |\omega(e_s)|} \sum_{w_{ri} \in \omega(e_r)} |\{w_{sj} \in \omega(e_s): w_{sj} > w_{ri}\}| \quad (3.8)$$

Equation (3.8) induces a partial order in the space of our stochastic weights: a sampled pdf for the weights of an edge e_r will be considered strictly smaller than the one of another edge e_s if and only if $\forall x \sim \rho_{e_r}, y \sim \rho_{e_s}: p_{xy}(x < y) = 1$, or conversely, $\forall x \sim \rho_{e_r}, y \sim \rho_{e_s}: p_{xy}(y < x) = 0$. Otherwise both sampled pdfs will be considered incomparable⁸. If we wish to relax the mentioned requirement, we could set the weight of

⁵ The dynamic programming assumption is efficient (tractable) since it leads to the possibility of doing a local search for finding the global optimal path.

⁶ We do not explicitly write down the graph to which we refer as long as it does not lead to ambiguity.

⁷ This deduction only uses the fact that the integral of the Dirac's delta over the portion where its peak lies is 1 and the integral of the Dirac's delta times some function over the domain of that function (as long as the peak of the delta also lies within that domain) is the value of the function at the peak of the delta. Expression $|A|$ denotes the size (number of elements) of set A .

⁸ They will be equal only if their respective histories of sampled weights contain the same elements.

e_r smaller than the one of e_s iff $\forall x \sim \rho_{e_r}, y \sim \rho_{e_s}: p_{xy}(x < y) \geq \pi$ or conversely $\forall x \sim \rho_{e_r}, y \sim \rho_{e_s}: p_{xy}(y < x) \leq 1 - \pi$, where $\pi \in [0, 1]$ is a threshold (1 for no threshold, 0 for setting all the pdfs as incomparable). In this way, we sacrifice decision power in the shortest path search for the sake of diminishing, for example, the influence of outliers or sporadic measure errors.

Now we turn into demonstrating if the dynamic programming assumption holds with the defined optimality measure. Let say that there are two different paths $\lambda_1 = \langle e_{\lambda_1, 1}, e_{\lambda_1, 2}, \dots, e_{\lambda_1, s_1} \rangle$ and $\lambda_2 = \langle e_{\lambda_2, 1}, e_{\lambda_2, 2}, \dots, e_{\lambda_2, s_2} \rangle$ that connect vertex v_a to vertex v_b . Since the total weights of both paths are sampled pdfs (recall equation (3.5)), we do not lose generality if consider those paths as single edges, let say e_1 and e_2 respectively, that is, the paths total weights are sampled pdfs that can be written as $\hat{\rho}_{\lambda_1}(x) = \frac{1}{|\omega(e_1)|} \sum_{s=1}^{|\omega(e_1)|} \delta_{w_{1s}}(x)$ and $\hat{\rho}_{\lambda_2}(x) = \frac{1}{|\omega(e_2)|} \sum_{t=1}^{|\omega(e_2)|} \delta_{w_{2t}}(x)$, respectively, where $|\omega(e_q)|$ is the number of samples from the pdf of the complete path λ_q (which in turn is the product of the number of samples from all the edges of the path, according to (3.5)), and w_{qh} are those samples, each one being the sum of one sample taken from each edge of the path.

Now assume that there is another edge $e' = (v_b, v_c)$ that may extend both paths. For the dynamic programming assumption to hold, the extension must preserve the optimality in the set $\{\lambda_1, \lambda_2\}$, that is,

$$\lambda_1 < \lambda_2 \Rightarrow \lambda_1 \circ e' < \lambda_2 \circ e' \quad (3.9)$$

where the “less than” comparison between paths is an abbreviation for $p(\hat{\rho}_{\lambda_1} < \hat{\rho}_{\lambda_2}) = 1$ (and $p(\hat{\rho}_{\lambda_1 \circ e'} < \hat{\rho}_{\lambda_2 \circ e'}) = 1$, respectively) and the $\circ$ operation is the concatenation of a new edge to the end of an existing path to form a new path.

By using (3.4), the sampled pdfs for the extended paths are

$$\begin{aligned} \hat{\rho}_{\lambda_1 \circ e'}(x) &= \frac{1}{|\omega(e_1)| |\omega(e')|} \sum_{(w_r, w'_s) \in \omega(e_1) \times \omega(e')} \delta_{w_r + w'_s}(x) \\ \hat{\rho}_{\lambda_2 \circ e'}(x) &= \frac{1}{|\omega(e_2)| |\omega(e')|} \sum_{(w_r, w'_s) \in \omega(e_2) \times \omega(e')} \delta_{w_r + w'_s}(x) \end{aligned} \quad (3.10)$$

respectively. The probability that the former is smaller than the latter is, using (3.8),

$$\begin{aligned} p(\hat{\rho}_{\lambda_1 \circ e'} < \hat{\rho}_{\lambda_2 \circ e'}) &= \frac{1}{|\omega(e_1)| |\omega(e_2)| |\omega(e')|^2} \sum_{(w_r, w'_s) \in \omega(e_1) \times \omega(e')} | \{ (w_r, w'_s) \in \omega(e_2) \times \omega(e') : w_r + w'_s > w_i + w'_j \} | = \\ &= \frac{1}{|\omega(e_1)| |\omega(e_2)| |\omega(e')|^2} \sum_{w_i \in \omega(e_1)} \sum_{w'_j \in \omega(e')} | \{ (w_r, w'_s) \in \omega(e_2) \times \omega(e') : w_r + w'_s > w_i + w'_j \} | = \\ &= \frac{1}{|\omega(e_1)| |\omega(e_2)| |\omega(e')|^2} \sum_{w_i \in \omega(e_1)} \sum_{w'_j \in \omega(e')} \left[\frac{| \{ (w_r, w'_s) \in \omega(e_2) \times \omega(e') : w_r > w_i \} | - | \{ (w_r, w'_s) \in \omega(e_2) \times \omega(e') : w_r > w_i \wedge w_r + w'_s \leq w_i + w'_j \} |}{| \{ (w_r, w'_s) \in \omega(e_2) \times \omega(e') : w_r \leq w_i \wedge w_r + w'_s > w_i + w'_j \} |} \right] = \\ &= \frac{1}{|\omega(e_1)| |\omega(e_2)| |\omega(e')|^2} \sum_{w_i \in \omega(e_1)} \left[\frac{| \omega(e') | | \{ (w_r, w'_s) \in \omega(e_2) \times \omega(e') : w_r > w_i \} | - \sum_{w'_j \in \omega(e')} | \{ (w_r, w'_s) \in \omega(e_2) \times \omega(e') : w_r > w_i \wedge w_r + w'_s \leq w_i + w'_j \} |}{\sum_{w'_j \in \omega(e')} | \{ (w_r, w'_s) \in \omega(e_2) \times \omega(e') : w_r \leq w_i \wedge w_r + w'_s > w_i + w'_j \} |} \right] = \\ &= \frac{1}{|\omega(e_1)| |\omega(e_2)| |\omega(e')|^2} \sum_{w_i \in \omega(e_1)} \left[\frac{| \omega(e') |^2 | \{ w_r \in \omega(e_2) : w_r > w_i \} | - \sum_{w'_j \in \omega(e')} | \{ (w_r, w'_s) \in \omega(e_2) \times \omega(e') : w_r > w_i \wedge w_r + w'_s \leq w_i + w'_j \} |}{\sum_{w'_j \in \omega(e')} | \{ (w_r, w'_s) \in \omega(e_2) \times \omega(e') : w_r \leq w_i \wedge w_r + w'_s > w_i + w'_j \} |} \right] = \end{aligned} \quad (3.11)$$

On the other hand, the probability for $\lambda_1 < \lambda_2$ is

$$p(\hat{\rho}_{\lambda_1} < \hat{\rho}_{\lambda_2}) = \frac{1}{|\omega(e_1)||\omega(e_2)|} \sum_{w_i \in \omega(e_1)} |\{w_r \in \omega(e_2) : w_r > w_i\}| \quad (3.12)$$

By hypothesis, we consider (3.12) to equal one, which only occurs if all the $w_r \in \omega(e_2)$ are greater than any $w_i \in \omega(e_1)$. Inserting that hypothesis into equation (3.11) yields:

$$\begin{aligned} p(\hat{\rho}_{\lambda_1 \circ e'} < \hat{\rho}_{\lambda_2 \circ e'}) &= \frac{1}{|\omega(e_1)||\omega(e_2)||\omega(e')|^2} \sum_{w_i \in \omega(e_1)} \left[|\omega(e')|^2 |\omega(e_2)| - \sum_{w_j \in \omega(e')} |\{(w_r, w'_s) \in \omega(e_2) \times \omega(e') : w_r + w'_s \leq w_i + w'_j\}| \right] = \\ &= 1 - \frac{1}{|\omega(e_1)||\omega(e_2)||\omega(e')|^2} \sum_{w_i \in \omega(e_1)} \sum_{w'_j \in \omega(e')} |\{(w_r, w'_s) \in \omega(e_2) \times \omega(e') : w_r + w'_s \leq w_i + w'_j\}| \end{aligned} \quad (3.13)$$

The second term of this last equation counts the number of pairs taken from the extended path $\lambda_2 \circ e'$ in which the sum of the weight samples of the pair is smaller than any pair of samples from the extended path $\lambda_1 \circ e'$. That is, the term reflects a lost of optimality of λ_1 in favour of λ_2 resulting from extending both paths with e' . If and only if that term is zero, $p(\hat{\rho}_{\lambda_1 \circ e'} < \hat{\rho}_{\lambda_2 \circ e'})$ is 1 and the dynamic programming assumption is satisfied.

For clarifying this result, consider that the samples of the weights of e_1 , e_2 , and e' are not empty, and thus they have maxima and minima that define three intervals: $[m_1, M_1] = [\min(\omega(e_1)), \max(\omega(e_1))]$, $[m_2, M_2] = [\min(\omega(e_2)), \max(\omega(e_2))]$, and $[m', M'] = [\min(\omega(e')), \max(\omega(e'))]$ respectively, with $m_2 > M_1 \equiv m_2 - M_1 > 0$ if the hypothesis $p(\hat{\rho}_{\lambda_1} < \hat{\rho}_{\lambda_2}) = 1$ holds. The extended paths have the corresponding intervals $[m'_1, M'_1] = [m_1 + m', M_1 + M']$ and $[m'_2, M'_2] = [m_2 + m', M_2 + M']$. Under this interval based formulation, the dynamic programming assumption holds if and only if $m_2 + m' > M_1 + M' \equiv m_2 - M_1 > M' - m'$, which cannot be completely assured in any graph.

3.2. Summarizing Weights into Single Values

One possible solution to the intractability of shortest path search when considering the weights of edges as complete probability distributions is to summarize the weights into single real values, for example the mean of the distributions or a function of the first moments [7]. In the following we will use as the new weights of the edges a cost or utility function:

$$c(e_r) = E[\rho_{e_r}(x)] + k \sigma^2[\rho_{e_r}(x)] \quad (3.14)$$

Notice that with this definition for the cost function, due to the linearity of expectation and variance (the latter because the involved variables are independent and unscaled), the sum of costs satisfies:

$$\begin{aligned} c(e_r \circ e_s) &= E[\rho_{e_r + e_s}(x)] + k \sigma^2[\rho_{e_r + e_s}(x)] = E[\rho_{e_r}(x) + \rho_{e_s}(x)] + k (\sigma^2[\rho_{e_r}(x)] + \sigma^2[\rho_{e_s}(x)]) = \\ &= E[\rho_{e_r}(x)] + E[\rho_{e_s}(x)] + k \sigma^2[\rho_{e_r}(x)] + k \sigma^2[\rho_{e_s}(x)] = c(e_r) + c(e_s) \end{aligned} \quad (3.15)$$

Thus we have a well defined deterministic cost with a total order, and thus the dynamic programming assumption holds and for example the Dijkstra's algorithm [6], which has a computational cost of $O(|V|^2)$ (see Fig. 1), can be applied. However, we cannot use any algorithm yet from our definition of costs, since

the pdfs $\rho(x)$ are unknown. We have to obtain an statistical approximation from our sampled distributions $\hat{\rho}(x)$. An unbiased estimator of the real mean and variance are the statistical sample mean and variance:

$$\begin{aligned}
 \hat{c}(e_r) &= \frac{1}{|\omega(e_r)|} \sum_{i=1}^{|\omega(e_r)|} w_{r_i} + k \frac{1}{|\omega(e_r)|-1} \sum_{i=1}^{|\omega(e_r)|} \left[w_{r_i} - \left(\frac{1}{|\omega(e_r)|} \sum_{i=1}^{|\omega(e_r)|} w_{r_i} \right) \right]^2 = \\
 &= \frac{1}{|\omega(e_r)|} \sum_{i=1}^{|\omega(e_r)|} w_{r_i} + k \frac{1}{|\omega(e_r)|-1} \sum_{i=1}^{|\omega(e_r)|} w_{r_i}^2 + k \frac{1}{|\omega(e_r)|-1} \sum_{i=1}^{|\omega(e_r)|} \left(\frac{1}{|\omega(e_r)|} \sum_{i=1}^{|\omega(e_r)|} w_{r_i} \right)^2 - 2k \frac{1}{|\omega(e_r)|-1} \sum_{i=1}^{|\omega(e_r)|} w_{r_i} \left(\frac{1}{|\omega(e_r)|} \sum_{i=1}^{|\omega(e_r)|} w_{r_i} \right) = \\
 &= \frac{1}{|\omega(e_r)|} \sum_{i=1}^{|\omega(e_r)|} w_{r_i} + k \frac{1}{|\omega(e_r)|-1} \sum_{i=1}^{|\omega(e_r)|} w_{r_i}^2 + k \frac{1}{(|\omega(e_r)|-1)|\omega(e_r)|} \left(\sum_{i=1}^{|\omega(e_r)|} w_{r_i} \right)^2 - 2k \frac{1}{(|\omega(e_r)|-1)|\omega(e_r)|} \left(\sum_{i=1}^{|\omega(e_r)|} w_{r_i} \right) \sum_{i=1}^{|\omega(e_r)|} w_{r_i} = \\
 &= \frac{1}{|\omega(e_r)|} \sum_{i=1}^{|\omega(e_r)|} w_{r_i} + k \left(\frac{1}{|\omega(e_r)|-1} \sum_{i=1}^{|\omega(e_r)|} w_{r_i}^2 - \frac{1}{(|\omega(e_r)|-1)|\omega(e_r)|} \left(\sum_{i=1}^{|\omega(e_r)|} w_{r_i} \right)^2 \right)
 \end{aligned} \tag{3.16}$$

This approach also has the advantage that the sum of weights does not increment the number of samples in the sampled pdfs, thus it is memory efficient.

```

1  Dijkstra(Graph, SourceVertex)

2      dist <- array of |V| // min. dist. to each vertex from SourceVertex
3      dist[u<>SourceVertex] <- infinity
4      dist[SourceVertex] <- 0
5      previous <- array of |V| // previous vertex of each vertex when
                                // following shortest paths backwards to
                                // SourceVertex

6      Q <- Graph
7      while Q is not empty:
8          u <- vertex w in Q with least dist[w]
9          delete u from Q // for preventing cycles
10         for each destination vertex v in Q following an edge from u
11             alt <- dist[u] + length(u, v)
12             if (alt < dist[v]) then
13                 dist[v] <- alt
14                 previous[v] <- u
15             end if
16         end for
17     end while
18     return previous

```

Fig. 1. Deterministic shortest path Dijkstra's algorithm in pseudocode.

4. Extensions

There are three evident extensions to the work presented here: i) to extend the formalization to negative weights and use the Bellman-Ford algorithm instead of the Dijkstra's; ii) to include more prior information about the probability distributions of weights in order to enhance the estimations (in this report we rely on the law of large numbers exclusively, which may have a poor convergence), for example bayesian or recursive bayesian approaches; iii) to evaluate more precisely the estimation error of using sampled pdfs.

Bibliography

[1] Evdokia Nikolova, **Path Planning Under Uncertainty: Complexity and Algorithms**, *International Conference on*

Automatic Planning and Scheduling - ICAPS'06 Doctoral Consortium - Thesis Abstract), 2006.

- [2] R.P. Loui, **Optimal Paths in Graphs with Stochastic or Multidimensional Weights**, *Communications of the ACM*, vol. 26, no. 9, pp. 670-676, 1983.
- [3] C.E. Sigal, A.A. B. Pritsker, J.J. Solberg, **The Stochastic Shortest Route Problem**, *Operations Research*, vol. 28, no. 5, pp. 1122-1129, 1980.
- [4] D.J. White, **Markov Decision Processes**, *John Wiley & Sons*, ISBN 978-0471936275, 1993.
- [5] R.H. Myers, S.L. Myers, and R.E. Walpole, **Probability and Statistics for Engineers and Scientists**, *Prentice-Hall*, 2002.
- [6] E.W. Dijkstra, **A Note on Two Problems in Connexion with Graphs**, *Numerische Mathematik* vol. 1, pp. 269- 271, 1959.
- [7] L. Deng, M.D.F. Wong, **An Exact Algorithm for the Statistical Shortest Path Problem**, *Proceedings of the 2006 conference on Asia South Pacific design automation*, 2006.